Hand Segmentation with Mask-RCNN Using Mainly Synthetic Images as Training Sets and Repetitive Training Strategy

Amin Dadgar and Guido Brunnett
Computer Science, Chemnitz University of Technology, Straße der Nationen 62, 09111, Chemnitz, Germany

Keywords: Machine Learning, Neural Networks, Deep Learning, Segmentation, Synthetic Training Set, Transfer Learning, Learning Saturation, Premature Learning Saturation, and Repetitive Training.

Abstract: We propose an approach to segment hands in real scenes. To that, we employ 1) a relatively large amount of sorely simplistic synthetic images, 2) a small number of real images, and propose 3) a training scheme of *repetitive training* to resolve the phenomenon we call *premature* learning saturation (for using relatively large training set). The results suggest the feasibility of hand segmentation subject to attending to the parameters and specifications of each category with meticulous care. We conduct a short study to quantitatively demonstrate the benefits of our repetitive training on a more general ground with the Mask-RCNN framework.

1 INTRODUCTION

For the training of deep neural networks, the creation and annotation of large amounts of data are major issues. The feasibility of employing synthetic training data appears to be attractive for the immense costs reduction foreseeable in those phases.

A recent advancement on this direction is an approach referred to as ***SaneNet*** (Dadgar and Brunnett, 2020). Albeit the employment of simplistic synthetic data as the training-set, their approach demonstrated promising results on detecting hands in various scenarios. Inspired from the *invariancy concept* existed in conventional deep nets, the SaneNet exploited this notion further to accomplish the goal. Building on these experiences, we extend this approach towards the more challenging task of hand segmentation.

There exist discrepancies (e.g., in texture, scene, background, and object details) between synthetic and real images. A workaround for attenuating the impacts of those discrepancies is to employ a larger amount of synthetic data with higher diversity. Supposedly, increasing the amount of training data should not pose an obstacle, for the data can be generated and annotated automatically. However, in training our networks on relatively larger amounts of synthetic data, we discovered a problem we refer to as *premature* learning saturation (Fig.1).

Learning *saturation* is a state of training in which the training loss does not decrease meaningfully as the hidden units output values close to the asymptotic ends of the activation function (Rakitianskaia and Engelbrecht, 2015). This state is usually displayed in the loss, after an initial drop and a somewhat consistent decrease, by undirected fluctuations. Such behavior can have many reasons, such as, high initial weights, a small-sized net (e.g., underfitting), and an improper learning rate. However, if one selects these parameters carefully, networks would encounter this state during the final stages of training.

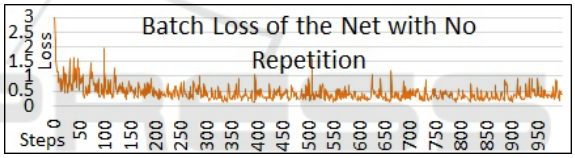


Figure 1: Learning saturation is a well-known state of training phases in which the training loss does not decrease meaningfully and can have many reasons. This state occurs in all training phases of our experiments. However, when we have to employ relatively larger datasets, this learning saturation would be premature. That means the network enters that state in the early stages/steps. That leads the networks to merely learn the main features from the initial portion of the training set and almost ignore the rest of the data. The Figure shows one epoch, with 1000 steps, of a training process using 45K synthetic images. After approximately step 300^{th}, the loss seemingly worsens, and the training enters the saturation phase. Thus the network has seemingly processed merely 15000 images effectively (e.g., one-third of the entire training session). In other words, about 30K of them could not contribute meaningfully to the learning. We call the phenomenon *premature* learning saturation.

Dadgar, A. and Brunnett, G.
Hand Segmentation with Mask-RCNN Using Mainly Synthetic Images as Training Sets and Repetitive Training Strategy.
DOI: 10.5220/0011658900003417
In *Proceedings of the 18th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2023) - Volume 4: VISAPP*, pages 220-228
ISBN: 978-989-758-634-7; ISSN: 2184-4321

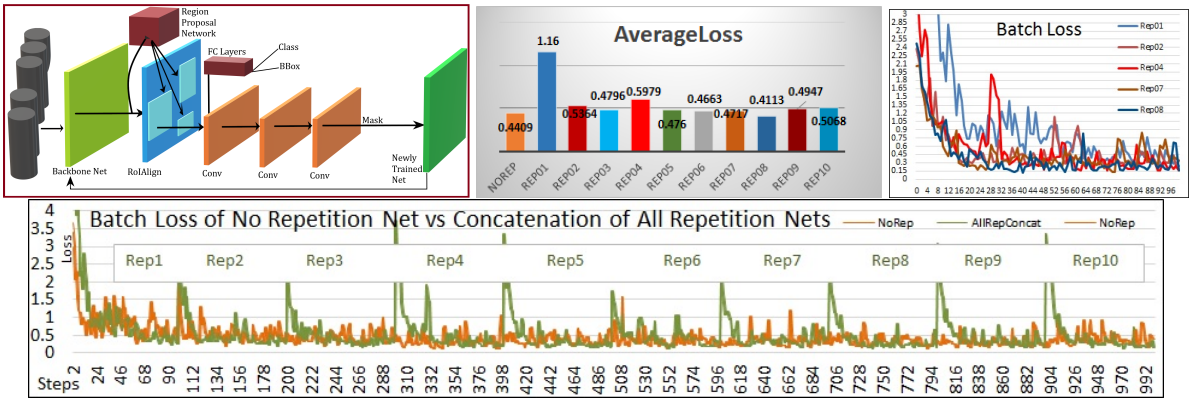


Figure 2: Repetitive training (upper left) is a specific scheme to approach training. There, we divide our database into smaller subsets. Then, we train our several networks over each of these subsets. While training, we employ the resulting network from the previous subset as the backbone of the following subset's training. With this approach, we aim to address the issue of premature learning saturation. In upper middle figure, besides clear improvements in the average loss after each repetition (except for repetition 4), the overall loss (0.4113) after 8^{th} repetition is better than the loss of the training with no repetition (0.4409) on the far left orange bar. In this short comparison, repetitions 9 and 10 did not introduce any improvements in the average loss. Therefore, we would consider repetition eight's network as the final net. In upper right figure, each training session has one epoch, with 100 steps (and 4.5K images). By arguing the occurrence of saturation on approximately $> 80^{th}$ step, the saturation takes place in relatively late stages and thus is not premature, demonstrating a potential explanation for the advantage of our repetitive training. The bottom figure shows the comparison between the batch losses of the repetitions (in a concatenated format) and of the conventional training without repetition. The green curve demonstrates the advantage of repetitive training (specially at the end of the epoch).

The problem arises when networks enter that state ere long the final (compared to the total number of) training stages/steps. In cases where synthetic data constitutes a much larger shred of the training-set, and thus the size of the training sets ought to be higher than usual, the learning saturation seems to occur sooner (thus premature). Therefore, facing a trade-off becomes inevitable. That is, on one side, we resort to increasing the size of the synthetic set to compensate for the existing discrepancies between synthetic and real data. On the other side, networks seldom learn (if at all) useful features, after some *early* stages, due to that premature learning saturation consequence.

In our experiments, we frequently encountered the phenomenon directly after the networks merely employed a small portion of the training data. Consequently, we suspected this early (premature) saturation leads the networks to learn simply from the initial training data and seemingly, ignores almost the rest.

To mitigate that problem, we propose a particular type of learning scheme called *repetitive training* (Fig.2-left). In our approach, we first randomly divide our relatively large set of training data into smaller subsets. Then, starting from the first subset, we train the default backbone CNN (e.g., *Resnet*101). Next, we employ the resultant network from the previous training phase as the backbone net, and with the second subset, we train a new network. We continue this procedure until all subsets are used. Therefore, we *repetitively* transfer the knowledge learned from the net (and subset) of the last iteration to the next.

The main focus of this paper is to report on our results for hand segmentation using networks mainly trained with synthetic data (plus a few real images ≈ 250) and with repetitive training. We consume no effort to introduce a new model's architecture. Therefore, we employ a well-known neural network architecture which is called the Mask-RCNN (He et al., 2020). To segment a new object, a dataset with $\approx 4K$ images has an adequate size for training a network of such characteristics. However, the size of our synthetic data amounts to $\approx 45K$ images, thus higher than the typical size by $\times 10$. Therefore, one can examine the effect of repetitive training on this network.

There is a set of parameters that plays a crucial role in determining the performance of the networks (Section 4). Among them, the background's similarities (besides the repetitive training) seems to play the most significant role. Therefore, here our focus is to segment hands on the inputs when the background has similarities with the training set, using repetitive training on Mask-RCNN. We leave the more general investigations of our repetitive training's benefits on different scenarios and networks for the future works.

Thus, our work goes beyond the existing literature in two ways. First, we extend the Mask-RCNN with mainly synthetic data to segment hands. Secondly, with our repetitive training, we exploit the transfer

learning in a novel way suitable for this challenging task (on specific examples), and Mask-RCNN, despite employing mainly sorely simplistic synthetic images (alongside a few real ones which have background similarities with the test set) as the training set.

Since our results point toward hand segmentation on a limited number of test sets, we end the introduction by elaborating more on the general advantages of repetitive training in resolving premature saturation, based on a short comparative study (Fig.2). To conduct a systematic comparison, we perform both of these training schemes with the same training set (e.g., synthetic data without any real data), the same number of epochs (e.g., one), and equal values for the learning rates (e.g., 0.001). The number of *overall* steps also is equal (e.g. 1000). In the case of conventional training (e.g., the orange bar at the far left side of the Fig.2-upper middle), we carry out these 1000 steps in one session. However, in the case of repetitive training, each repetition has 100 steps (e.g., $10 \times 100 = 1000$). The major difference the two, here, is the number of layers under training. For conventional training, we let *all* layers, whereas, for the repetitive training, we consider alternative layers of *all,all,all,3+,3+,all,all,4+,5+,all* learn from our training set. Finding the optimal frozen layers for each repetition can be an iterative process.

As seen in Fig.2-upper right, the repetitive training improves the average loss continuously (expect on the 4^{th}) with the best loss of 0.4113 on the 8^{th} repetition (which is better than the conventional loss of 0.4409). Repetitions 9 and 10 did not improve the average loss. Therefore, we would consider the 8^{th} net as the final network. Besides, the Fig.2-bottom shows the comparison between the batch losses of all repetitions (with concatenation) and that of the conventional training. The green curve demonstrates our repetitive training's advantages at the end of each epoch. We illustrate the loss of a selected repetitions in Fig.2-upper right. One can argue that saturation begins after step 80. Compared to 100 total steps, the saturation is not *premature* anymore, illustrating each net learns from much of the data, and providing a potential explanation for the advantages of our repetitive training.

Though the prime inspiration of the method is to repeatedly train the resulting networks on the next subset. In Section 4 we demonstrate utilizing an identical subset for retraining the next net, in some cases, would also lead to satisfactory results. Testing the broader applicability of the repetitive training, as a more general training strategy, requires an extensive study of different CNN models on various application domains and objects. Therefore, here, we provide evidence for the feasibility of the approach for a specific problem, particular object, and a certain network.

2 LITERATURE REVIEW

Currently, there exist many successful object segmentation frameworks, and they fall into two main categories: *semantic*, and *instance* segmentation. The earlier one is a class-level segmentation (e.g., DeepLab (Chen et al., 2018)), whereas, the latter one, on the other hand, identifies each object on instance-level for all trained classes.

Belonging to the family of region-based convolutional neural networks or RCNN (Girshick et al., 2012), the Mask-RCNN demonstrated one of the most successful performances on instance segmentation. The current default framework segments 99 objects spanning a wide range of categories (e.g., from living creatures to electronic devices). It supplies a simple framework that is straightforward to extend to segment a new set of objects. Besides, it possesses other significant properties that make it an appropriate choice of network for this study.

First, it provides us with a wide range of possibilities in setting parameters, from as simple as epoch numbers to as sophisticated as freezing some layers during the training. That permits a more thorough investigation of proposed approaches. Second, besides employing standard convolutional neural networks (e.g., *VGG*, *Resnet*50, or *Resnet*101 (He et al., 2016)), MRCNN allows us to employ a self-trained net as the *backbone* to transfer learning. That is essential to study the properties of repetitive training.

Transfer learning (Pan and Yang, 2009), enables machine learning frameworks to transfer knowledge by storing the information (e.g., weights in the realm of convolutional neural networks, CNN), gained from one field of a problem (e.g., person segmentation) and applying it to a related but *different* domain (e.g., hand segmentation). Our strategy of repetitive learning also follows the approach of transfer learning, but in a different fashion. More specifically, the knowledge is not transferred to a different domain, but to the same domain (with different training-set and different parameters) *over and over again*. Beside investigating this training strategy, an informative study in which how a segmentation framework, in general, and the Mask-RCNN, in specific, would perform when trained on synthetic data, is missing in the literature.

Figure 3: Forearm rotation with the bind-pose fingers (Top). Forearm rotation with changing fingers' states (Bottom).

3 METHODOLOGY

Synthetic Data. To attend to the goal of SaneNet extension to segmentation, we employ a synthetic database with similar constraints as in (Dadgar and Brunnett, 2018). That database is created by considering *in-plane-axis* rotation axes as following: If $\overrightarrow{V} = \{v_i | i = 1,2,...,28\}$ is the total pose vector of one (right) hand then,

$\{v_1, v_2, v_3\}$ is the global translation (3 DoF, L_1),

$\{v_4, v_5, v_6\}$ is the global rotation (3 DoF, L_2),

$\{v_7, v_8, \}$ is the wrist rotation (2 DoF, L_3),

$\{v_9, v_{10}, v_{11}\}$ is the little finger rotation, L_{5_1},

$\{v_{13}, v_{14}, v_{15}\}$ is the ring finger rotation, L_{5_2}

$\{v_{17}, v_{18}, v_{19}\}$ is the middle finger rotation, L_{5_3},

$\{v_{21}, v_{22}, v_{23}\}$ is the index finger rotation, L_{5_4}, and

$\{v_{25}, v_{26}, v_{27}, v_{28}\}$ is the thumb finger rotation, L_{5_5}.

By selecting the Forearm rotation with bind-pose (L_2) and with fingers motion (L_{11}) in *in-plane-axis* rotation (Fig.3), the data structures are:

$$\begin{aligned} \overrightarrow{V}_{L_{2_{InPlane}}} &= \{v_4, v_5\} \\ \overrightarrow{V}_{L_{11_{InPlane}}} &= \{v_4, v_5, v_9, v_{10}, v_{11}, v_{12}, v_{13}, v_{14}, \\ & v_{15}, v_{17}, v_{18}, v_{19}, v_{21}, v_{22}, v_{23}, v_{25}, v_{26}, v_{27}, v_{28}\} \end{aligned} \quad (1)$$

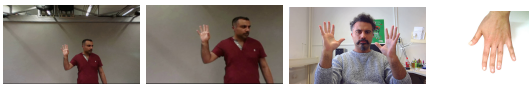
Considering 2°-step resolution over two *in-plane* axes for both L_2 and L_{11} data structures (1), there will be $[(360/2 \times 360/2) = 32400 \times 2] \approx 65K$ poses. By eliminating the non-plausible poses, the final number of synthetic poses will be $45K+$. We then construct the images with white background and simple shading and extract the mask of the hands and perform the automatic annotation.

Repetitive Training. For extending transfer learning to repetitive training, we divide the $45K+$ synthetic images into S smaller subsets by randomly selecting the images. We, in this paper, use $S = 4$ or $= 10$ with an equal number of synthetic images in each subset ($\approx 10K$ or $\approx 4.5K$). Then, we train a set of networks using each of those subsets within a queue of sessions. That is, beginning with the first subset, we train the first network. Then, we employ the resultant network as the backbone net for the next training session, and by employing the second subset, we train a second network. We continue the procedure until all subsets are used. To initialize (e.g., training the first network), we employ an off-the-shelf net (e.g., *Resnet*101) as the backbone. For each repetition session, the Mask-RCNN parameters' values, such as step, epoch, ROI numbers, and the frozen layers, can be different marking one advantage of this training scheme. For it assists us to concentrate on improving a particular aspect of the network during each session.

Real Data. In our approach, we consider a few real images from three different datasets for training purposes. We will analyze these three sets and their characteristics later in detail (Section 4). However, here we mark that there are 128 images from Egohands (Fig.5) (Bambach et al., 2015), 130 images from Kawulok (Fig.7) (Kawulok et al., 2014), and two more sets from alternative labs as 125 images from Altlab1 (Fig.6-left), and 125 images from Altlab2 (Fig.6-right). For some networks, we employ merely one set, and for some others, we might use two real-sets, in turn or combined. That would lead the total number of real images either to be ≈ 130 or ≈ 250. In either cases, the percentage of the real to the synthetic data, μ, remains insignificant (e.g., $\frac{130}{45K} = 0.29\%$, or $\frac{250}{45K} = 0.56\%$). However, their specifications (e.g., background, scale, size) should play a crucial role in the successful segmentation.

4 EXPERIMENT

We spotted relatively a large set of parameters (in three categories) require to be properly adjusted: **a)** Employing repetitive-training and its settings or not using it, **b)** Real training data specifications, and **c)** Entire training data specifications. We consider four parameters from the first two categories (**a)** and **b)**) to define the experiments' *type*s (Table 1): 1) **AHA**: All Real Hands in the training-set are Annotated, 2) **BGS** and 3) **SCS** which stand for Background and Scale Similarity between the real part of training-set and the test-set respectively, and 4) **RPT**: Repetitive Training. By setting them true or false, we will have $2^4 = 16$ variant net types which rises to 18 types (denoted as A-R) with two extra cases of training with no real date. However, we employ a selection of them



Vid_1 Vid_2 Vid_3 Img_1

Figure 4: Four test sets: Vid_1, which has a similar background and context with one of the training sets. Second, Vid_2, which is a cropped and scaled-up version of the Vid_1 with the identical size. Third, Vid_3, which its background, scale, and context are different from the real training sets. Fourth, Img_1, which consists of randomly selected 1500 images from 11K-Hand dataset (Afifi, 2017) 100 of which are annotated for IoU and $F1_{score}$ computation.

in our evaluations (Sections 4.1) to keep the paper within reasonable limits. Additionally, some of these types (e.g., $Ty_{(F)}$) do not require training the network anew but rather can be addressed by employing a different test sets. We train the networks with stochastic gradient descent (SGD) optimizer, momentum 0.9, learning rate of 0.001, and the initial backbone net as Resnet101 unless mentioned otherwise. During the training, we set the image size to $1024 \times 800px$.

The category **c)** refers to a set of parameters such as *i*) the subsets' number and the amount of images in each, and *ii*) the replications' number for real images. The *replication* parameter is necessary for, the amount of synthetic images (compared to the real ones) is excessive (e.g., $\geq 45K$). Thus the probability of real images' engagement within the training process decreases for each step and the chance of networks to learn more from them before entering the saturation phase decreases. Therefore, we replicate (e.g., simple copy-paste) them ($n = 5$ or 10) times to alleviate the issue. Additionally, from category **a)**, there is another parameter by which we determine the choice of frozen layers. If we systematically defined the experiments with these parameters (too), we would have exceeded the limits of this paper.

We consider four test sets for the evaluations (Fig.4): First, Vid_1 which has a similar background and context with one of the training sets Fig.6 and contains one subject. There, the hand undertakes various challenging postures and the frames' size is $1920 \times 1100px$. Second, Vid_2 is a cropped and scaled-up version of Vid_1 with the same frame size. Vid_2 enables us to examine and investigate the influence of scales on performance. Third, Vid_3 which has different background, scale, and context with *Altlab* training sets Fig.6. The video contains one subject but two hands (which sometimes occlude each other), and the frames' size is $1920 \times 1080px$. Fourth, Img_1 which consists of randomly chosen 1500 images from the 11K-hand dataset (Afifi, 2017). This test-set contains limited hand postures but is diverse over the subjects, skin colors, and genders with the size of

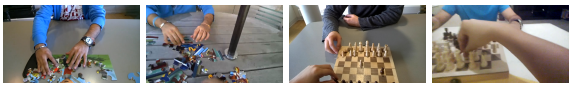


Figure 5: EgoHands Dataset (Bambach et al., 2015). They have the size of 1280×720. The scenarios here are *two-subject-games* (e.g., chess), and therefore, there exist variant amounts of hands with complex backgrounds.

$1600 \times 1200px$. For IoU and $F1_{score}$ calculations of Img_1, we randomly select 100 images from it.

To gain an insight into our four training image sets, we demonstrate their characteristics and their comparisons with the test sets briefly: First is the synthetic images (Fig.3) with the size of $1280 \times 720px$ and plain white backgrounds (comparable to the Img_1 test-set). The images contain no shadows and possess merely shading. The hand model has a fixed size, and the hand's region relatively covers a small portion of the entire image.

Second is the Egohands images (Fig.5) which have the size of $1280 \times 720px$. The scenarios are *two-person-games* (e.g., chess), and we select them in a fashion that they contain no heads. Here, there exist variant amounts of (maximum four) hands and complex backgrounds. Thus, there are seldom semblances of scale and context between this and the test sets. Additionally, for this set, we consider two annotation strategies: a) right-hands only, b) all-hands.

Third, the $Altlab_1$ images (Fig.6-left) have the size of $1920 \times 1100px$. Its scenario is free movements of hands with occasional *digit* gestures. The set contains heads, and the background color and the scene are identical to that of the Vid_1 test-set. That leads the Altlab1's set to be the most similar training-set to one of the test sets (Vid_1). However, there are valuable discrepancies on the subject, hands' side, and many gestures between them. Fourth, the $Altlab_2$ images (Fig.6-right) have the size of $1920 \times 1100px$. The scenario here is also free movements of hands with occasional digit gestures. Besides containing similar discrepancies of $Altlab_1$, this time, the background and the scene are different from that of the Vid_1 test-set.

Finally, the Kawulok images (Fig.7) have their sizes range from 151×208 to 640×480. The scenario in Kawulok set is digit-gestures. The images here contain merely one hand with no heads and have a simplistic (but non-white) backgrounds, and the hands cover most of the image patch. Thus, this set is the closest one, in characteristics, to the Img_1.

By considering the content of these real sets, we define two experiments as follows: When the training set contains synthetic images with 1) No real data, 2) Various combinations of real data. To evaluate the performance of the segmentation networks quantitatively, we consider four metrics: I) Mean IoU, II) Ac-

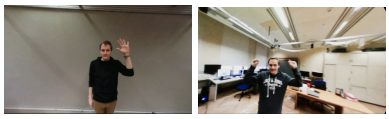


Figure 6: The $Altlab_1$ training-set (left) and $Altlab_2$ training-set (right). Both sets have images with the size of 1920×1100 pixels and contains head with scenario of free movement of hands with occasional digit gestures. The $Altlab_1$ has the background color and the scene identical to that of the Vid_1 test-set. However, the background color and the scene of the The $Altlab_1$ are different from Vid_1.

Figure 7: Kawulok1Hand Dataset (Kawulok et al., 2014). The selected images have their sizes range from 151×208 to 640×480. The scenario here is specific digit-gestures. The entire images of this set contain merely one hand, and they do not include heads. They have a simplistic (but non-white) background, and the hands cover most of the image.

curacy (Acc), and III) $F1_{Score}$. Besides these three, we define a fourth metric denoted as TPH. It reveals the number of true positives oriented toward the number of segmented hands having the *highest* score. There are frames in which the true positive has the highest score compared to the false positive scores. Thus we can filter out the false positives as if there is none. Therefore, the TPH can reveal valuable information about the performance of the networks.

4.1 Evaluation

We trained and tested an ample amount of networks (300+) with variant parameter settings to achieve our best results, pacing through a trial-and-error path. For the evaluation purposes, we select a limited number of (= 11) cases (Table 3, 4) within two experiments considering 6 types (Table 2).

$\boldsymbol{Exp}_1$: This experiment employs two net types ($Ty_{(A)}$ and $Ty_{(B)}$) seeking to investigate the potentials of the synthetic dataset and repetitive training, when no real data is employed. Since the background of our training set, here, is simplistic, the main focus here is to investigate the influence of the repetition for segmenting hands on Img_1 which has simple backgrounds ($C_{\#_1}$ and $C_{\#_2}$ in Table 3). However, we present two more cases that can reveal interesting information about our simplistic training set. The cases $C_{\#_1}$, $C_{\#_3}$, and $C_{\#_4}$ have the same network that we test on three different test sets Vid_1, Vid_2, Img_1, respectively ($epochs = 5$, $steps = 1000$, $ROI = 300$, and trained layers = 4+). The case $C_{\#_2}$ employs repetitive training (ten repetitions) with the same learning rate as other cases, various epoch numbers for each repetition (maximum 5), and the following trained layers: $all, all, 3+, 3+, 4+, 4+, 5+, 5+, heads, heads$. As seen in the table, $C_{\#_2}$ though have a slightly worse Acc and $F1_{score}$, due to merely one more false positive on $C_{\#_2}$. However, both of them have $TPH = 100$, so the false positives can be canceled out, emphasizing more weight on the IoU for their comparison. As for the IoU, we face an improvement (to almost 90%) on $C_{\#_2}$, pointing toward the benefits of our repetitive training scheme. In $C_{\#_3}$ and $C_{\#_4}$ (with more complex backgrounds), there are many false positives (Fig.8-a). However, the true positives are also high resulting in noticeable scores. In $C_{\#_4}$, the hands' scale becomes similar (to the synthetic images), and the background becomes simpler. Therefore, the performance is much better than in $C_{\#_3}$. One major issue with $C_{\#_3}$ and $C_{\#_4}$ is that the true positives do not exhibit the highest scores for us to filter false cases (low TPH), suggesting we should include real data and repetitive training if higher IoU and $F1_{score}$ are sought. To end, we also carried out repetitive training with *no* real data for Vid_1 and Vid_2. The repetition reduced false positives, only in some cases. However, the true positives also decreased, so we do not report those cases.

$\boldsymbol{Exp}_2$: This experiment seeks to investigate the potentials of the synthetic dataset and repetitive training, combined with real data. For Egohands dataset, we employ the annotation of all-hands and in some cases annotation of right hands-only. Furthermore, we divide the training set into 10 and 4 subsets. This experiment (considering all scales and backgrounds similarities) fall into types $Ty_{(D)}$, $Ty_{(L)}$, $Ty_{(P)}$, and $Ty_{(R)}$ with illustrating the eight best cases. When using Egohands (thoroughly distinctive background) in our training set, the performance does not suggest any advantages. However, there is a case that reveals a unique behavior ($C_{\#_5}$). That is, false positives show considerable reduction using our repetitive training (and 10 subsets with $4.5K$ images in each set). As shown in Fig.8-b, though there are no true positives, the mere reduction of false positives to zero is an optimistic outcome because we employ it as the backbone net of our future repetitions. Zero false positives are achievable in earlier repetitions of our training scheme. However, we continue the repetition until robust error-free results are witnessed (e.g., $C_{\#_5}$) on Vid_1 (and also on Vid_2). We train this network with epoch=4, spets=150, RoI=150, with $\times 5$ replications of Egohands, and only right hands are annotated. Then we employ the $Altlab_1$ dataset (that has similarities of backgrounds with Vid_1 and Vid_2) as a part of our training set, with dividing the synthetic dataset into 4 subsets (each containing $\approx 10K$ images)

Table 1: By setting RPT, AHA, BGS, and SCS true/false and adding two case (of no real date) we have 18 types (A-R).

Type	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
RPT	×	✓	×	✓	×	✓	×	✓	×	✓	×	✓	×	✓	×	✓	×	✓
AHA	No RIDB	No RIDB	×	×	×	×	×	×	×	×	✓	✓	✓	✓	✓	✓	✓	✓
BGS			×	×	×	×	✓	✓	✓	✓	×	×	×	×	✓	✓	✓	✓
SCS			×	×	✓	✓	×	×	✓	✓	×	×	✓	✓	×	×	✓	✓

Table 2: Network's types we selected in our experiments.

Network's types of Evaluation Section						
Exp	Exp_1		Exp_2			
Type	A	B	D	L	P	R

Network's types of Ablation Study Section								
Type	K	L	M	N	O	P	Q	R

Table 3: $\boldsymbol{Exp}_1$: No real data is included in the training sets.

	Input	TtlHnd	IoU	Acc	F1
$C_{\#_1}$	Img_1	100	0.837	0.862	0.926
$C_{\#_2}$	Img_1	100	0.89	0.855	0.922
$C_{\#_3}$	Vid_1	1425	0.483	0.259	0.412
$C_{\#_4}$	Vid_2	1425	0.580	0.515	0.680

and $\times 10$ replications. In repetitive training, instead of using *Resnet*101, we employ $C_{\#_5}$ as the backbone. Subsequently, that leads to indirectly combining the $Altlab_1$ and Egohands (and their advantages) during the training sessions. With such a combination, we decrease the false positives while increasing the true positives. In $C_{\#_6}$, we train from layer fourth onward and thus freeze the previous ones. Though the number of FP is high, *most* metrics experienced significant improvements compared to $C_{\#_3}$. We restart the training from layer three onward, and with this modification, signs of profound improvement begin to appear, ($C_{\#_9}$ in Table 4 and Fig.8-c and -d). Testing the identical network of $C_{\#_9}$ on the Vid_2 (in $C_{\#_{10}}$), decreases $F1_{Score}$. However, there is an increment in the number of those true positives (e.g., hands) that exhibit the highest value. That suggests the network's success on distinct hand scales. In the repetition for $C_{\#_9}$, merely changing the freezing layer (and not the subset) was adequate to significantly improve the results compared to $C_{\#_8}$. For training these networks, we set $epoch = 4$, $steps = 500$, and $ROI = 300$.

Table 4: $\boldsymbol{Exp}_2$: Real data is included in the train sets.

Repetition on *Resnet*101: with Egohands, 10 Subsets					
	Input	TtlHnd	IoU	Acc	F1
$C_{\#_5}$	Vid_1	1425	NAN	NAN	NAN
Repetition on $C_{\#_5}$: with $Altlab_1$ 4 Subsets, Starting From Lyr 4					
$C_{\#_6}$	Vid_1	1425	0.592	0.361	0.530
Repetition on $C_{\#_5}$: with $Altlab_1$ 4 Subsets, Starting From Lyr 3					
$C_{\#_7}$	Vid_1	1425	0.826	0.030	0.059
$C_{\#_8}$	Vid_1	1425	0.747	0.543	0.704
$C_{\#_9}$	Vid_1	1425	0.700	0.811	0.900
$C_{\#_{10}}$	Vid_2	1425	0.703	0.607	0.755
$C_{\#_{11}}$	Vid_3	1500	0.452	0.560	0.718

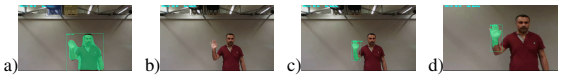


Figure 8: Result visualization: a) $C_{\#_3}$ resulted in many false positives, b) $C_{\#_5}$ created optimism because of zero false positives. The best results achieved using c) $C_{\#_9}$, and d) $C_{\#_{10}}$.

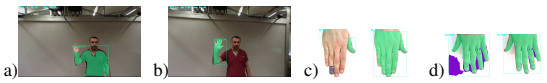


Figure 9: Result visualization: a) False Positives when *Resnet*50 is the Backbone Network, b) True Positives, c) Comparison of a network similar to $C_{\#_9}$ with a network trained on Kawulok Dataset Fig.7, d) Comparison of networks (trained with no real data) using conventional training $C_{\#_1}$ and using repetitive training $C_{\#_2}$ on Img_1.

4.2 Ablation Study

In exploring our way for a successful segmentation we considered training various types (See Table 2) and initially focused on using Vid_1. One of the troublesome false positives to mitigate during this period was the 'person' (or 'subject'). Because, during many testing phases, its region remained among the FPs irrespective of the selected parameter settings. Initially, we suspected this false positive comes with the off-the-shelve backbone default classes. However, by altering the backbone from *Resnet*101 to *Resnet*50 and employing repetitive training (Fig.9-a), the segmented area of the false positives is partial and resembles a scaled-up hand. That led us to this realization, the hand's scales and backgrounds in the real data play significant roles. Additionally, the initial training's repetitions resulted in no true positives. However, as we continued the repetition process, the TPs increased to a skimp but an informative amount (five segmentations) as in Fig.9-b. To further examine the significance of scale similarities, we trained one more set of networks with our selected Kawulok dataset (Fig.7), with repetition, and tested them on the

Img_1. As qualitatively visible in Fig.9-c, the network trained on the Kawulak dataset as a part of the training set achieves higher IoU (0.78 compared to 0.06 when trained on $Altlab$s). We anticipated that is for the higher similarities in scales the Kawulak dataset has with Img_1.

We then employed the $Altlab$ datasets (See Fig.6) as the real part of training-set ($Ty_{(M)}$, $Ty_{(N)}$, $Ty_{(O)}$, $Ty_{(P)}$, $Ty_{(Q)}$, and $Ty_{(R)}$). By employing $Altlab_1$, the false positive number enhanced. However, since the overall true positive faced a slight decay, the $F1_{score}$ slightly decreased. By employing $Altlab_2$ ($C_{\#_6}$), we achieved slightly better enhancement for the false positives. We speculated it is because of the variant backgrounds the $Altlab_2$ training set contains. However, true positives showed more decrease resulting in the same $F1_{score}$. When the images' number in each subset was $10K$, the performance did not always illustrate improvements. That points to the networks' sensitivity of the training on the number of images each subset ought to contain, and consequently the number of repetitions we should carry out during the training.

Above, we witnessed enhancement in true positives using $Altlab_1$. In $C_{\#_5}$, we faced enhancement of false positives using Egohands. Therefore, we considered training with the combination Egohands dataset with $Altlab$ (especially $Altlab_1$) to intertwine the advantages of both training sets: 1) reducing the false positives with the Egohands dataset and 2) increasing the true positives with $Altlab_1$. We considered six types of $Ty_{(K)}$, $Ty_{(L)}$, $Ty_{(M)}$, $Ty_{(N)}$, $Ty_{(Q)}$, $Ty_{(R)}$ with various settings, such as the number of real data replication (e.g., $\times 10$) and the number of subsets and repetitions. However, we did not achieve satisfactory performance. That is, despite witnessing some improvements in IoU, by employing repetitive training, a pivotal improvement (on both IoU and $F1_{Scores}$) was not measurable. All these guided us: First, to employ Egohands data, carry out the repetition ($C_{\#_5}$). Second, to use the $Altlab_1$ and perform another repetition ($C_{\#_9}$). That is, we combined the two datasets in two separate repetitive sessions toward $\boldsymbol{E}xp_2$.

5 DISCUSSION & CONCLUSION

We extended the Mask-RCNN framework to hand segmentation using mainly simplistic synthetic images and a few real images (SaneNet approach) as the training-set. To benefit from our sorely simplistic data, we introduced a specific training scheme with *repetition*. We had to accurately set a range of parameters (from the real to synthetic data, and the repetition to frozen layers) to achieve optimal results.

For training and testing ample number of networks, we considered five training and four test sets, with diverse characteristics (backgrounds, scales, and scenarios). The performance was significant even when our training set was mainly simplistic synthetic data. We also conducted a short study (Section 1), quantitatively demonstrating the benefits of our repetitive training on a more general ground (with Mask-RCCN). There, we showed that repetitive training can help improving the average and batch loss when we employ similar parameters for both schemes. Initially, we aimed for the right-hands only segmentation. But this goal is not feasible, even if we train the Deeplab, (Chen et al., 2018) with purely $4K$ real images.

As future works, we can pace into several fascinating paths. First, we can consider repetitive training using purely (but with slightly more sophisticated) synthetic data. That enables us to analyze the effectiveness of the method in eliminating the necessity of incorporating real images. As a piece of evidence in that direction, we trained networks solely on synthetic data once without repetitive training ($C_{\#_1}$) and once with repetition ($C_{\#_2}$) and achieved a higher IoU on a simple test-set (See Fig.9-d). Second, we can evaluate the efficacy of our training scheme on a wider range of tasks such as classification, detection, segmentation, and pose estimation networks using purely real data. Alongside diverse tasks and networks, the consideration of various objects (besides hands) can examine the validity of this training as a more general scheme in the realm of machine learning.

ACKNOWLEDGEMENTS

This project was funded by the Deutsche Forschungsgemeinschaft (DFG) CRC 1410 / project ID 416228727. I also thank Mr. Stefan Helmert, with whom I conducted fruitful scientific discussions.

REFERENCES

Afifi, M. (2017). 11K Hands: Gender recognition and biometric identification using a large dataset of hand images. *Springer*.

Bambach, S., Crandall, D. J., and Yu, C. (2015). Lending A Hand: Detecting Hands and Recognizing Activities in Complex Egocentric Interactions. In *IEEE*.

Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., and Yuille, A. L. (2018). DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. *IEEE Trans PAMI*, 40(4):834–848.

Dadgar, A. and Brunnett, G. (2018). Multi-Forest Classification and Layered Exhaustive Search Using a Fully Hierarchical Hand Posture / Gesture Database. In *VISAPP*, Funchal.

Dadgar, A. and Brunnett, G. (2020). SaneNet: Training a Fully Convolutional Neural Network Using Synthetic Data for Hand Detection. *IEEE SAMI*, pages 251–256.

Girshick, R., Donahue, J., Darrell, T., Malik, J., and Berkeley, U. C. (2012). Rich feature hierarchies for accurate object detection and semantic segmentation.

He, K., Gkioxari, G., Dollár, P., and Girshick, R. (2020). Mask R-CNN. *IEEE Trans on PAMI*, 42(2):386–397.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. *IEEE Proc CVPR*, Dec:770–778.

Kawulok, M., Kawulok, J., Nalepa, J., and Smolka, B. (2014). Self-adaptive algorithm for segmenting skin regions. *Eurasip Journl on Adv in SigProc*, (1):1–22.

Pan, S. J. and Yang, Q. (2009). A Survey on Transfer Learning. *IEEE Knowledge & Data Eng*, 194:781–789.

Rakitianskaia, A. and Engelbrecht, A. (2015). Measuring saturation in neural networks. *IEEE Symp Series on Computational Intelligence*, pages 1423–1430.