UNIVERSITY OF TARTU
Institute of Computer Science
Computer Sciences Curriculum

Karl Suurkaev

Seeing the forest behind the trees: A novel method for generating data for overlapping object segmentation

Bachelor's Thesis (9 ECTS)

Supervisors:
Tõnis Laasfeld, MSc
Kaspar Hollo, MSc
Dmytro Fishman, PhD

Tartu 2023

Seeing the forest behind the trees: A novel method for generating data for overlapping object segmentation

Abstract:

Computer vision is a rapidly developing academic field which is gaining traction throughout different fields of expertise. Deep learning and artificial neural networks are at the forefront of recent developments but many problems remain unsolved. One of the prominent problems is the detection and segmentation of overlapping objects from images. This thesis proposes a novel layered data acquisition approach for images with overlapping objects which aims to improve the ground truth data quality. The data was generated using a custom built robotic system. The resulting dataset was tested against the U-Net and YOLOv5 artificial neural networks. Additionally, the same network models were trained on a directly annotated dataset for better results comparison. The thesis also investigated if this new data acquisition approach could be automated using artificial neural networks. The results showed that the novel approach is on par with the direct approach but allows automation of ground truth data generation. This potentially allows easy generation of large datasets which improves model quality through substantially larger quantities of training data.

Keywords:

Deep learning, computer vision, overlapping objects, robotics

CERCS:

P176 - Artificial intelligence; P170 - Computer science, numerical analysis, systems, control

Kuidas puude tagant metsa näha: Uudne andmete loomise lähenemine kaetud objektide liigendamiseks

Lühikokkuvõte:

Masinnägemine on kiiresti arenev akadeemiline valdkond, mida rakendatakse erinevates valdkondades. Sügavõpe ja tehisnärvivõrgud on ühed edukamatest masinnägemise meetoditest, kuid mitmed ülesanded on senini lõplikult lahendamata. Üks olulisim lahendamata probleem on kattuvate objektide tuvastamine ja liigendamine. Käesolev lõputöö pakub välja uue kihilise andmete kogumise meetodi kattuvate objektidega piltide jaoks, mille eesmärk on andmete kvaliteedi parendamine. Andmete kogumiseks kasutati erilahendusena disainitud robootilist süsteemi. Tulemusena saadud andmestikku kasutati U-Net ja YOLOv5 tehisnärvivõrkude treenimiseks. Lisaks treeniti võrdlusena tehisnärvivõrke klassikaliselt annoteeritud andmestiku peal. Lõpuks uuriti ka võimalusi annotatsioonide automaatseks genereerimiseks. Tulemused näitasid, et uus lähenemine saavutab võrreldava kvaliteedi otsese meetodiga, kuid võimaldab potentsiaalselt suuremahulist automaatset andmete märgendamist. See lubaks omakorda saavutada kõrge kvaliteediga tulemusi tänu suuremale treeningandmete mahule.

Võtmesõnad:

Sügavõpe, masinnägemine, kattuvad objektid, robootika

CERCS:

P176 - Tehisintellekt; P170 - Arvutiteadus, arvutusmeetodid, süsteemid, juhtimine (automaatjuhtimisteooria)

Table of Contents

Introduction

With advancements in artificial intelligence, efforts have been made to adapt machine learning methods to work in a manner that is inspired by the way neurons operate in the human brain. One problem domain has been the visual apprehension of our surroundings which in the field of artificial intelligence has been aptly titled computer vision. Deep learning models have been at the forefront of computer vision developments but there are still many difficulties to overcome.

One of these difficulties is the detection and segmentation of objects that overlap with each other. Deep learning models have had problems with fully detecting objects that overlap with each other as data identifying an object can become obfuscated. This obfuscation forces deep learning models to make predictions in an area it can not directly observe. The same issue also arises when annotating the data as information needed to fully annotate an object of interest can become hidden from the human eye. Knowing the shape of the objects and context, humans are able to predict the extent of the hidden part of the object, but these predictions might not always be correct. A general solution to the overlapping object segmentation problem has not been found.

This thesis presents a new approach to data generation and annotation for images with overlapping objects. The new layered approach aims to improve the quality of the data in comparison to direct manual annotation approaches. The thesis will investigate to what extent data quality can improve with this new approach, how the changes in data quality affect the training and predictions of deep learning models, and if the data annotation for this approach could be automated using deep learning models.

1 Background

Machine learning (ML) is a widely applied and rapidly developing academic field. Common topics include the development of new machine learning model architectures including deep neural networks, aspects of dataset generation and preprocessing among others.

1.1 Computer Vision

Computer vision (CV) is an academic field dedicated to automatically obtaining meaningful information from visual inputs such as digital images and videos through the use of dedicated software and hardware [1]. The discoveries within this field of study have found many use cases such as the development of self-driving cars [2], the automatic sorting and grading of food items [3], the segmentation of cell nuclei [4] and a variety of tasks in other medical fields such as cardiology and pathology [5] to name just a few.

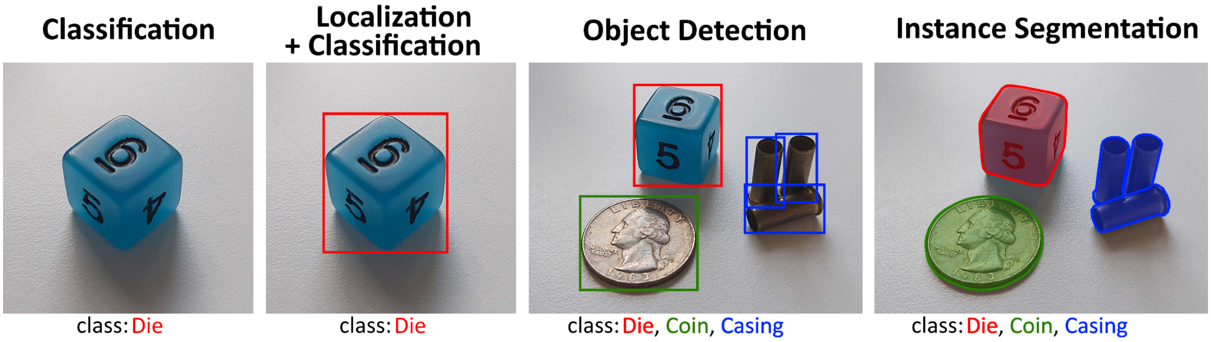


Figure 1. A visual showcase of the classification, localization, object detection and instance segmentation tasks.

Many of the problems that have been or are being solved with CV fall under one or several common categories including classification, localization, detection, and segmentation (Figure 1). The goal of object classification is to identify the class of the image, while localization is about determining the object's location on the image [6]. Object detection is about determining the locations and the classes of multiple objects on an image [6]. Instance segmentation goes a step further and attempts to determine the classes of the objects on the image and to determine exactly which pixels belong to the found object instances [7]. The tasks in this order become sequentially more complex because instance segmentation can always be converted into object detection bounding boxes and localization combined with classification is a special case of object detection.

1.2 Deep Learning

Deep learning (DL) is a ML approach where artificial neural networks (ANNs) are used to learn useful patterns from the underlying data [8]. In the context of images, these patterns are called features [9]. Features can range from low-level features (e.g., corners, edges) to high-level features (e.g., whole objects) (Figure 2).

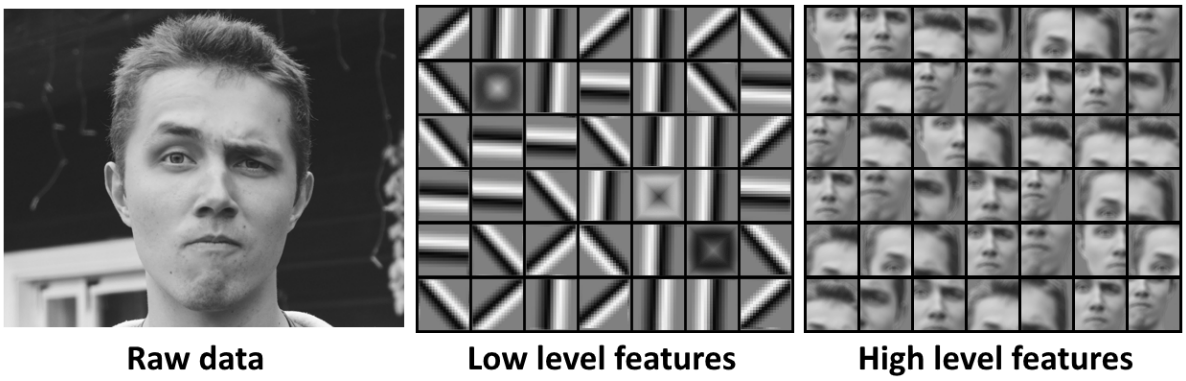


Figure 2. Visualisation of raw image data, low level features and high level features.

ANNs are ML models constructed from layers of nodes known as neurons, hence the name of the approach [10]. The connections between neurons are characterised by a weight and an activation function. During the training process the weights are adjusted so that the entire network minimises the difference between the neural network output and the expected output called ground truth. If each sample in the dataset has a corresponding ground truth label, the approach is called supervised learning [12]. The label types can vary depending on the task at hand such as classification or instance segmentation. Deep ANNs are differentiated from shallow neural networks due to employing a large number of layers in the neural network compared to shallow ANNs which employ a single hidden layer [10]. Deep ANNs generally allow learning higher level features and thus achieve higher quality with fewer weights.

1.2.1 Convolutional Neural Networks

Convolutional neural networks (CNNs) are a subtype of ANNs which are mainly used in the field of CV [12]. CNNs mainly consist of three types of layers: convolutional layers, pooling layers and fully-connected layers. The convolutional filters within the convolutional layers are applied to the input image to produce a set of feature maps. Pooling layers are used to decrease the size of the layer input by aggregating multiple values into one, for example by applying a 2x2 filter on the input and taking the maximum of all the values within the filter. Pooling layers are essential because they allow the construction of higher level features by allowing the next convolutional layer to receive input from increasingly distant parts of the image. The fully-connected layers perform in the same ways as they would in a standard ANN where each neuron applies the weights on the input.

1.3 Object Overlap Problem

Occluded objects are objects that are partially covered by other objects that are not of interest themselves, while overlapping objects are covered by other objects of interest (Figure 3). It is necessary to differentiate between occlusion and overlap as this can change how the data is interpreted and annotated. Occlusions commonly arise in various situations such as when detecting traffic signs from images [13], in which case a traffic sign could be partially occluded by a tree or a vehicle, while an overlap happens when a traffic sign is partially hidden behind another traffic sign.

It is also important to define if the entire object (entire overlap) or only the clearly visible part (partial overlap) of the object is to be detected (Figure 3). This thesis focuses on the detection of the entire object, including the segments which might be occluded or overlapping.

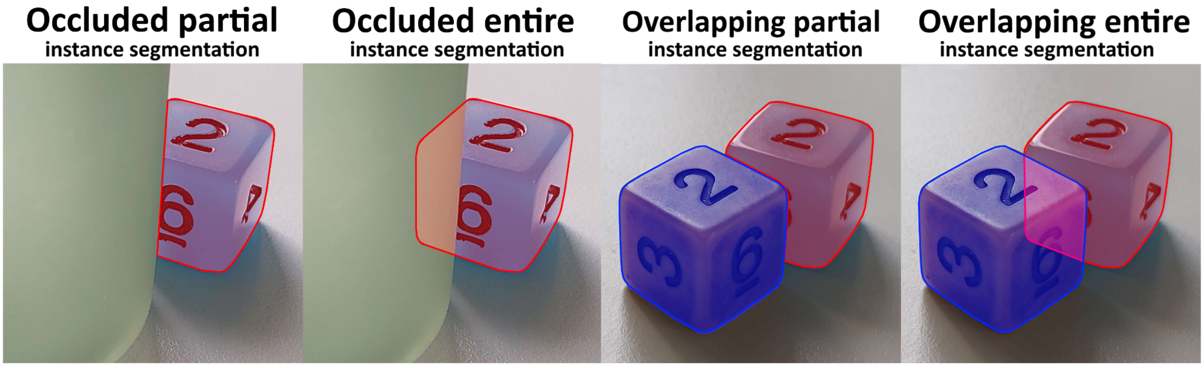


Figure 3. Examples of the differences between partial and entire instance segmentations and between occluded and overlapping objects.

The difficulty of entirely detecting an occluded or overlapping object arises from the fact that the ML model has to extrapolate the segmentation of hidden pixels by mostly relying on high level features. This means that during the training phase the model needs to learn to segment based on both high level and low level features while also learning when to ignore low level features for the hidden parts of objects. This task may be partially simplified if the obscuring objects are not entirely opaque [14] and convoluted versions of low level features may be used for segmentation. This will be taken advantage of in this study.

Several previous works have attempted to address overlapping object segmentation. In the works of Toda et al. regarding synthetic data creation for ANNs [15], the problem was diminished by thresholding the overlap area between objects. This approach helped avoid the problem by minimising overlap between objects but the problem still persisted. Zafari et al. [16] applied an approach where they set the approximate shape of the objects of interest which improved the prediction results. Knowing the shape of the object helped with detection, but it is difficult to generalise this solution to arbitrary shapes.

1.3.1 Methods for Ground Truth Generation for Overlapping Objects

Methods for ground truth generation for overlapping objects are somewhat different from strategies used for cases without overlaps or occlusions since it is necessary to annotate an area which is not directly observable. The naive option is to annotate objects manually by estimating the hidden parts of the objects which may suffer from errors. Another possibility is to use synthetic dataset generation [17]. However, this may produce artefacts as the synthetic generators may be unable to account for shadow and reflection generation and removal. Additional problem arises if the objects are transparent or semi-transparent as estimating optical caustics is a non-trivial problem [18]. Yet another potential approach is to construct 2D projections of 3D scenes through predicting the object movement [19]. However, such 3D scenes may be impossible or restrictively expensive to obtain in case of microscopy or medical imaging due to technological limitations.

1.3.2 Existing Datasets

There are not many readily available datasets that fit the purposes of this thesis. Public datasets can be found for occluded objects, such as the Occluded COCO[1] dataset and the OccludedPASCAL3D+[2] dataset, but these datasets are underrepresenting transparent or semi-transparent objects of interest that are overlapping with each other. The comparative availability of open source datasets with overlapping objects is low according to Papers with Code[3]. The datasets also have few public entries for model benchmarks. A dataset with semi-transparent overlapping objects had to be created for this thesis since no relevant readily available dataset could be found.

[1] https://paperswithcode.com/dataset/occluded-coco

[2] https://paperswithcode.com/dataset/occludedpascal3d

[3] https://paperswithcode.com/

2 Methods

2.1 Generation of Images with Semi-transparent Overlapping Objects

In order to explore the problem of precisely detecting the overlapping objects, a corresponding custom dataset was created. It was created by placing small semi-transparent objects on two separate layers after which three images were taken: one in which the layers were placed over each other and one of each layer separately. Semi-transparent objects were chosen as they do not cause complete loss of information about the object in case of object overlap. Eppendorf tube caps were chosen as the transparent objects and two Petri dishes acted as the two layers.

A dedicated robot was designed and 3D printed to automate the process of generating and acquiring data. The robot was made up of two arms with attached Petri dishes, one central frame, two axles, two tensile strings, a belt, a stepper motor, a web camera, and a single-board microcontroller (Figure 4). The stepper motor, which was connected to the drive axle with a belt, rotated the axle in 90 degree intervals. The axle wound up and then released the arms. Both arms were attached to the frame using the second axle and were connected separately to a pillar with tensile string. After an arm was wound up it was suddenly released which caused the tensile string to rapidly contract and pull the arm back. This sudden motion was used to automatically randomise object locations on the Petri dish and create variation in the data.

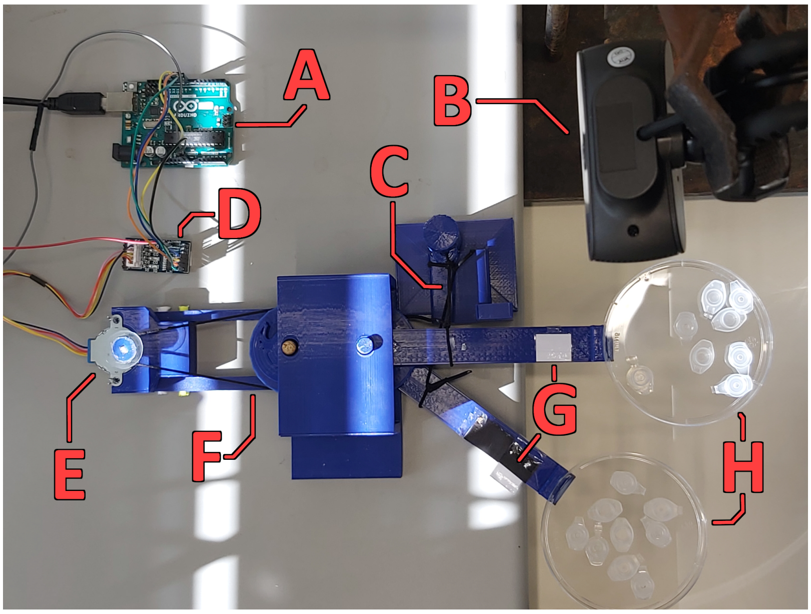


Figure 4. Robotic system for generation of images of overlapping objects. A) Arduino UNO R3 microcontroller B) Trust Trino web camera C) Tensile strings D) ULN2003 stepper motor driver E) 28BYJ-48 stepper motor F) Drive belt G) Alignment calibration markers H) Petri dishes with Eppendorf tube caps.

The winding and releasing of either arm of the robot was asynchronous (Figure 5). This allowed the capturing of the bottom object layer, then both layers overlapping, and then only the top object layer. Images were taken from a video feed of a web camera mounted on a top-down perspective. The video feed from the camera was also used to recalibrate the arms after every cycle in case there had been any misalignment from loss of traction between the belt and the axis. This cycle was then repeated by the microcontroller as many times as desired.

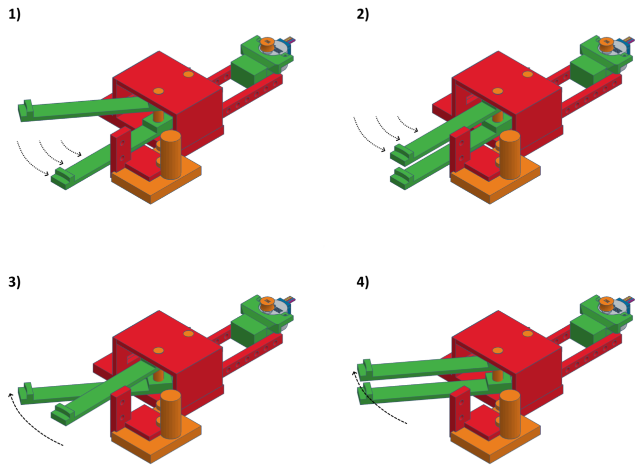


Figure 5. A diagram illustrating the motion cycle of the robot. 1) The lower arm is released. 2) The upper arm is released. 3) The lower arm is wound back. 4) The upper arm is wound back.

To realise this design many different materials and resources were used. The structural parts of the robot (Appendix 1) were designed using Tinkercad[4] and then 3D printed using PLA as the material. An Arduino UNO R3[5] single-board microcontroller was used to control the mechanism. The script for controlling the robot was written in MATLAB (R2022b)[6]. A Trust Trino[7] web camera was used for video input with a video resolution of 1280x720 at a 52° viewing angle. The mechanism was driven by a 28BYJ-48[8] stepper motor that was interfaced with an ULN2003[9] stepper motor driver.

2.2 Data Annotation

The generated data underwent two different types of annotation processes resulting in two ground truth datasets. Similar data splitting and augmentation was then applied to both datasets.

2.2.1 Direct Manual Annotation Approach

The direct annotation approach used only the images with the overlapping layers. No image quality improvement techniques other than best judgement were used to correctly annotate the images. This was possible since the general shape and size of the objects was known beforehand while some variation in the object's morphology remained. To ensure consistency, a set of rules and guidelines were made for data annotation (Appendix 2).

Though the annotation guidelines (Appendix 2) were followed, there was still often a need to rely on one’s best judgement on the object's location, orientation, shape and size. (Figure 6).

[4] https://www.tinkercad.com/
[5] https://docs.arduino.cc/hardware/uno-rev3
[6] https://se.mathworks.com/help/matlab/release-notes.html
[7] https://www.trust.com/en/product/18679-trino-hd-video-webcam
[8] https://components101.com/motors/28byj-48-stepper-motor
[9] https://www.elecrow.com/wiki/index.php?title=ULN2003_Stepper_Motor_Driver

Relying on best judgement lowered the quality of the annotations, as the annotation confidence was lower in the overlapping areas.

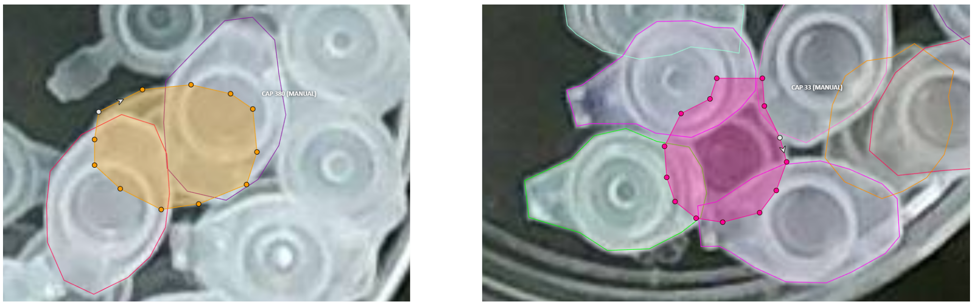


Figure 6. Annotations from the direct manual annotation approach. Examples of how the presence of an object can be seen but the exact size, shape and orientation is difficult to pinpoint due to overlap. The estimations of the occluded objects are highlighted by polygons.

2.2.2 Layered Annotation Approach

The second approach used the separate images of each layer (Figure 7). Each layer was annotated separately and then merged onto the corresponding images with the overlapping layers. This approach raised the quality of the annotations as object borders were clearly visible. The layered approach was taken after the direct manual approach had been completed over the entire dataset to avoid any bias from already having seen the overlapping objects separately.

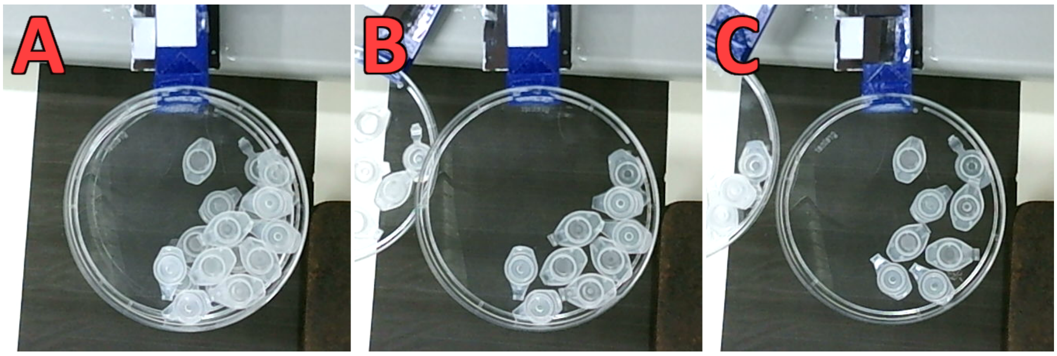


Figure 7. Visualisation of representative images from the datasets. A) Overlapped layers B) Top layer C) Bottom layer

2.2.3 Annotation Realignment

The robot's design used alignment flags to assure that the images in the single layers and overlapping layers would not shift between different acquisitions. However, in some cases this was not sufficient (Figure 8a) and a software alignment step was added. The resulting images were aligned with MATLAB (R2022b)[10] scripts using a control point registration approach (Figure 8b). The control points were selected manually.

[10] https://se.mathworks.com/help/matlab/release-notes.html

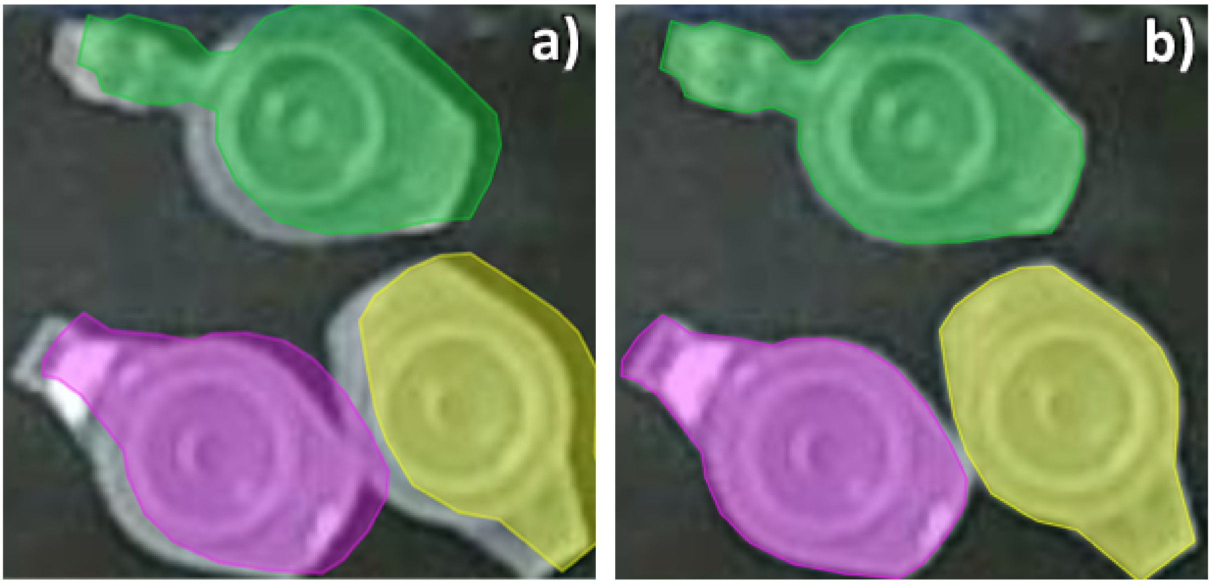


Figure 8. Examples of how objects and their respective annotations were not aligned on some images. The polygon instance annotations are highlighted with different colours.

2.2.4 Data Split and Augmentations

Data splits and augmentation were performed before feeding the data to the CNNs to ensure that the input data was as similar as possible between the different models and experiments. Additionally, the data in the training split was augmented to increase the number of training samples.

The data was split and augmented using the Roboflow[11] software. The original 27 images that the datasets were based on were split between training, validation and test sets. The split resulted in 19 training, 4 validation and 4 test images. For both datasets the images were assigned into splits manually based on the same image identifiers. The splits were set to have a matching pixel class distribution in the splits (Table 1).

The training images were randomly augmented using horizontal and vertical flipping, 90° rotations and an additional rotation between -30° and 30°, resulting in a total of 57 training images. The test and validation splits were not augmented.

During dataset splitting it was discovered that there were only a small proportion of pixels that were covered by more than two objects: ~0.2% of all pixels or ~2% of object pixels. When metrics were being calculated, all pixels that were covered by two or more objects were grouped together.

[11] https://app.roboflow.com/

Table 1. Table showing the distribution of pixel based classes in the data splits of the two datasets.

Dataset	**Direct manual dataset**			**Layered dataset**		
Class \ Split	Train	Validation	Test	Train	Validation	Test
Background: **0**	86.9%	87.4%	87.1%	86.6%	87.3%	86.8%
Object per pixel: **1**	9.91%	8.64%	9.80%	9.45%	8.13%	9.37%
Objects per pixel: **2**	3.08%	3.75%	2.87%	3.79%	4.35%	3.48%
Objects per pixel: **3**	0.13%	0.18%	0.23%	0.19%	0.21%	0.37%
Objects per pixel: **4**	0.001%	0.002%	0.013%	0.010%	0.006%	0.016%

2.3 Deep Learning Models

Two different DL model architectures, U-Net and YOLOv5, were used in the experiments to see how the different architectures and approaches were affected by the data acquisition methods. The architectures were chosen due to their wide usage in computer vision tasks.

2.3.1 U-Net

U-Net is a semantic segmentation CNN model [20]. The architecture consists of two main branches (an encoding branch and a decoding branch), a bottleneck layer and skip connections that connect the two branches.

The encoder extracts useful features and the context from the input image through the use of convolutional and max pooling layers. The decoder uses the extracted features and the captured context with upsampling and convolutional layers to produce the resulting segmentation of the object of interest. The skip connections connect layers from the two branches that are situated at the same levels. The skip connection arrangement allows the transfer of features directly from the encoder to the decoder and thus improving the quality of the final segmentation.

The U-Net model architecture used in this thesis is largely based on the works of Fishman et al. [21]. The trained models had input dimensions of 448 x 448 x 3, output dimensions of 448 x 448 x 5 and a depth layer of five with a total of 1 million trainable parameters.

2.3.2 YOLOv5

YOLOv5 is an object detection and image segmentation CNN model [22]. The architecture consists of three main components: the backbone, the neck and the head [23].

The backbone is a CNN that is used to extract useful features from the input image. Afterwards, the extracted image features are aggregated by the neck and sent to the head. Finally, the head carries out the object detection and segmentation by extracting regions of interest from the aggregated image features.

The YOLOv5 segmentation models created by Jocher et al. [22] were utilised in this thesis. The trained models had input dimensions of 448 x 448 x 3, output dimensions of 448 x 448 x

N, where *N* is the number of segmented objects. The number of trainable parameters depended on the specific YOLOv5 model used, with model N having 2 million parameters, model S having 7.6 million parameters, model M having 22 million parameters, and model L having 47.9 million parameters.

2.4 Metrics

Multiple different metrics were used to observe and measure the results of the experiments. Three pixelwise classes were considered one-by-one as the positive class for calculating metrics: one object on a pixel, two or more objects on a pixel, and one or more objects on a pixel (Figure 9). The metrics were calculated for all three classes.

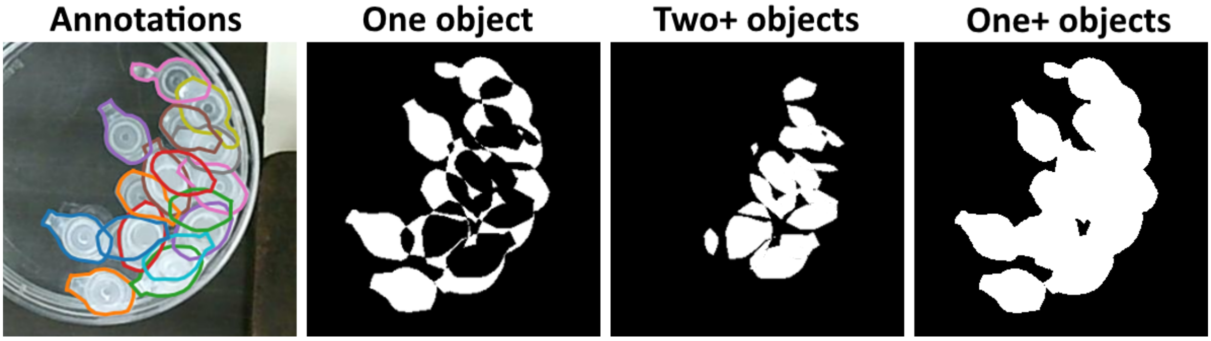


Figure 9. Visual representation of training labels and corresponding pixel classification tasks. From left to right: baseline image with annotations, binary mask for pixels which are covered by exactly one object, binary mask for pixels which are covered by two or more objects, binary mask for pixels which are covered by one or more objects.

A 2 x 2 confusion matrix was calculated for each pixelwise class. The cells of the confusion matrix were categorised into four parameters:

- true positives (TP) - correctly predicted to be objects of interest
- false positives (FP) - **in**correctly predicted to be objects of interest
- true negatives (TN) - correctly predicted **not** to be objects of interest
- false negatives (FN) - **in**correctly predicted **not** to be objects of interest

Precision, recall, F1 score, Matthew's correlation coefficient (MCC) and pixelwise IoU are metrics that are calculated using TP, FP, TN, and FN values derived from a confusion matrix. They are calculated using the following formulas:

$$Precision = \frac{TP}{TP+FP}$$

$$Recall = \frac{TP}{TP+FN}$$

$$F1\ score = 2 \cdot \frac{precision \cdot recall}{precision + recall}$$

$$MCC = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}$$

$$Pixelwise\ IoU = \frac{TP}{(TP+FP+FN)}$$

2.5 Experiments

The computation power (NVIDIA Tesla V100 32 GB GPUs) used for training the models used in the experiments was provided by the High Performance Computing Center[12] located at the University of Tartu. The following experiments were devised and carried out to achieve this work's aims.

2.5.1 Comparison of Ground Truth Generation Approaches

The purpose of this experiment was to create two datasets using different approaches and then compare the quality and similarity of these datasets. Two datasets were created: the first dataset was created following the direct manual annotation approach and the second dataset was created following the layered annotation approach. Both datasets contained 27 images. The annotations created from these two datasets were then compared against each other using the F1 score, MCC and pixelwise IoU. Once the comparisons were finished, augmentations and data splitting were applied to both datasets for use in further experiments.

2.5.2 Evaluating U-Net for Overlapping Object Segmentation

The goal of this experiment was to test U-Net's capabilities of detecting objects that overlap and to see how the two different datasets (Experiment 2.5.1) affected the detection results.

U-Net model training was repeated five times on both of the datasets using randomly initialised weights to assess the uncertainty of model predictions. As U-Net is not directly applicable for instance segmentation, it was used as a pixel classification instead with each class corresponding to the number of overlapping objects on any pixels (Table 1). The models were trained with a batch size of 32 for 1000 epochs. Early stopping was applied after 30 epochs of no improvements on validation loss. Each model was evaluated on the test sets from both datasets.

2.5.3 YOLOv5 Model Architecture Selection

This experiment was dedicated to finding out which model size offered the best results since YOLOv5 instance segmentation comes packaged with many different model sizes. The sizes, ranked from largest to smallest, are L, M, S, and N. The size affects the amount and size of the layers.

Two instances of the YOLOv5 instance segmentation model were trained for each of the packaged L, M, S, N model sizes. A model was trained on both of the previously created datasets (Experiment 2.5.1) for each of the model sizes. The models were trained with a batch size of 32 for 1000 epochs. Early stoppage was applied after 100 epochs of no improvements on model fitness, which is a weighted combination of metrics built into YOLOv5.

Each instance of the model was then used to predict instance segmentations of the objects on each image in the test split. These predictions were then compared to the ground truth of both datasets. The best model size was then selected based on metrics and a visual comparison of the prediction results.

[12] https://hpc.ut.ee/

2.5.4 YOLOv5 Instance Segmentation and Overlapping Objects

After the best performing YOLOv5 model size was found, the goal was to see how the prediction results on the test splits of the two previously created datasets (Experiment 2.5.1) compared to one another. The predictions of the two trained instances of the best performing model size were then compared with the ground truth of both datasets.

2.5.5 Comparison of Manual and Automatic Ground Truth Generation

The goal of this experiment was to see if it is possible to use CNNs to automatically generate ground truth for a dataset based on a small manually annotated dataset.

This experiment used only the dataset from the layered annotation approach. The leftmost quadrant was covered by a black rectangle on the layer images since some of the images had parts of the other layer's objects visible by the left border of the image. An instance of YOLOv5 was trained on the images and annotations of the separated layers. The model was trained with a batch size of 32 for 1000 epochs. Early stoppage was applied after 100 epochs of no improvements on model fitness. The trained model was then used to predict instance segmentations of the objects on the separated layers of the same test split used previously (Experiment 2.5.1).

3 Results and Discussion

The goals of the thesis were:

- Construct a robotic system for dataset acquisition allowing the generation of a layered annotation dataset.
- Assess the differences in quality between direct and layered annotation approaches.
- Assess the capability of U-Net and YOLOv5 to detect and segment overlapping objects.
- Explore the possibility of automatic ground truth generation using single layer images and the YOLOv5 model.

3.1 Comparison of Ground Truth Generation Approaches

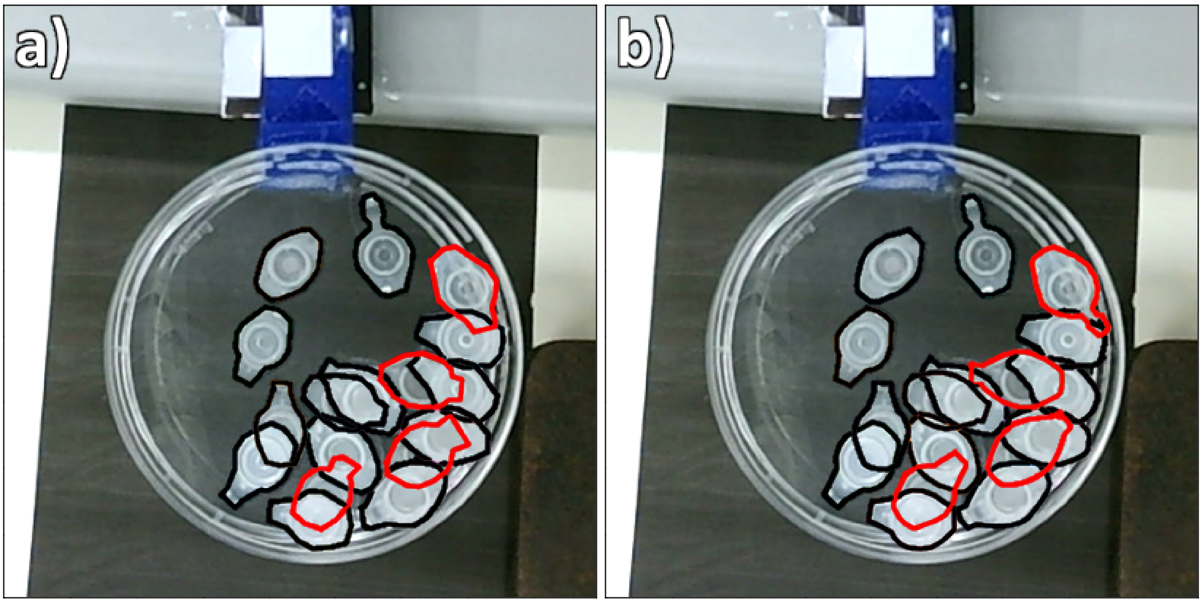


Figure 10. Example comparison of polygon annotation results. Object annotations that had a greatly improved size, shape and orientation accuracy have been highlighted in red. a) The results of the direct manual annotation approach. b) The results of the layered annotation approach.

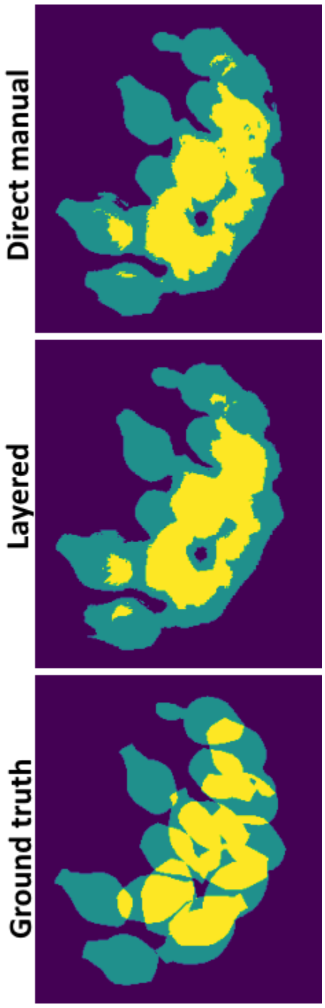
By the visual comparison of the two approaches (Figure 10) it became evident that the direct manual annotation caused multiple mistakes. For example, the hinges of the topmost red objects on both images are of different sizes and the hinges of the second topmost red objects are on opposite sides. These examples showcase why the metrics differ by a substantial margin (Table 2). The metrics may even overestimate the ground truth similarity as many objects did not have any overlaps. This proves that the novel data annotation approach has a significant impact on the obtained quality.

Table 2. F1 scores between the direct manual annotation dataset and the layered annotation dataset.

Class	One object on pixel	Two+ objects on pixel	One+ objects on pixel
F1 score	0.86	0.79	0.96
MCC	0.85	0.79	0.96
pw IoU	0.76	0.66	0.93

3.2 Evaluating U-Net for Overlapping Object Segmentation

According to the pixelwise metrics the prediction results for model instances trained on either dataset were very similar (Figure 11). The effect of random weights initialization on the outputs of the trained models was small. The maximum standard deviation among all classes were 0.02 for precision, 0.04 for recall, 0.02 for the F1-score, 0.01 for MCC, and 0.02 for pixelwise IoU. All U-Net instances were able to successfully determine if a pixel belonged to an object. All instances were also able detect the general regions where objects overlapped, but all trained models had problems pinpointing the borders of these regions. Surprisingly, none of the model metrics surpassed the corresponding metrics of the ground truth comparison experiment. The layered annotation approach helped to create higher quality ground truth but this alone was not enough to train a better model with the training samples currently available. This may indicate that a substantially larger dataset is needed to achieve a good prediction of overlapping pixels.



Approach	Direct manual	Layered
Class \ Metric	mean ± σ	mean ± σ
One object		
Precision	0.825 ± 0.012	0.816 ± 0.006
Recall	0.864 ± 0.012	0.847 ± 0.011
F1-score	0.844 ± 0.003	0.831 ± 0.004
MCC	0.828 ± 0.004	0.814 ± 0.004
Pixelwise IoU	0.730 ± 0.005	0.711 ± 0.005
Two or more objects		
Precision	0.772 ± 0.019	0.745 ± 0.022
Recall	0.714 ± 0.041	0.737 ± 0.022
F1-score	0.741 ± 0.016	0.741 ± 0.009
MCC	0.732 ± 0.015	0.731 ± 0.009
Pixelwise IoU	0.589 ± 0.020	0.588 ± 0.011
One or more objects		
Precision	0.953 ± 0.004	0.942 ± 0.001
Recall	0.965 ± 0.002	0.965 ± 0.004
F1-score	0.959 ± 0.001	0.953 ± 0.002
MCC	0.953 ± 0.001	0.946 ± 0.002
Pixelwise IoU	0.922 ± 0.002	0.910 ± 0.003

Figure 11.. Cropped visualisations of predictions compared to the ground truth and a table of the averages and standard deviations of pixelwise metrics where purple pixels are the background, cyan pixels are pixels with one object and yellow pixels are pixels with two or more objects. The metrics were calculated over the predictions of all five instances trained on either dataset. The metrics were calculated against the ground truth of the layered approach dataset.

Upon closer inspection of U-Net results it became apparent that U-Net started to correlate pixel brightness with object overlaps (Pearson r = 0.46) (Figure 12). As the objects on the images were placed on a dark background, the areas where the semi-transparent caps were overlapping had less visible background, causing those pixels to become brighter. This effect was amplified by the fact that the objects were not uniformly transparent as the inner rims of the caps were less transparent. Such an effect shows that the model relies on low level features and this may explain the lack of quality in segmenting the overlapping objects as overlapping object segmentation is expected to rely on higher level features.

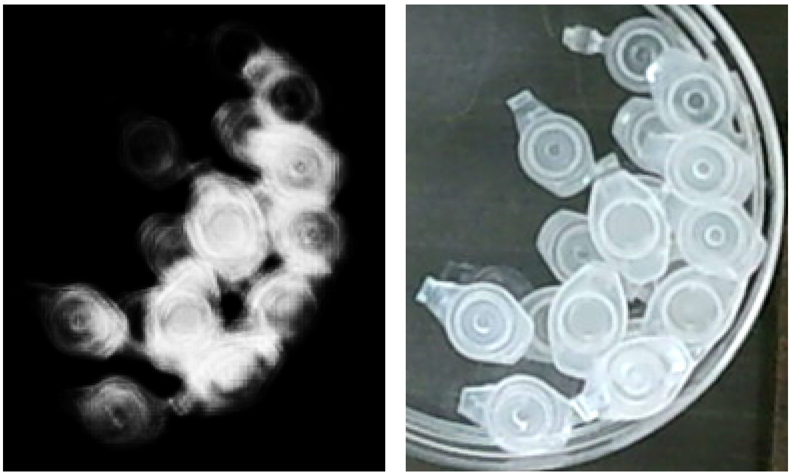


Figure 12. An example comparison between a prediction's confidence of overlap and the base image.

3.3 YOLOv5 Model Architecture Selection

Pixelwise metrics were not able to differentiate between the quality of different size YOLOv5 models. No clear pattern was found in the metrics (Table 3). For example the best metrics for the class of only one object per pixel were with the L size YOLOv5 instance, while the S sized instance outperformed the others when using the class of two or more objects per pixel.

Table 3. The pixelwise metrics of the different sized YOLOv5 models. The metrics were calculated against the ground truth of the layered approach dataset. The results of the instances that were determined to have the best performing size are highlighted in bold.

Approach	Direct manual				Layered			
Metric \ Size	L	M	S	N	L	M	S	N
One object								
Precision	**0.70**	0.72	0.71	0.72	**0.72**	0.73	0.76	0.70
Recall	**0.76**	0.68	0.65	0.72	**0.77**	0.60	0.58	0.61
F1-score	**0.73**	0.70	0.68	0.72	**0.74**	0.66	0.66	0.66
MCC	**0.70**	0.67	0.65	0.69	**0.72**	0.63	0.63	0.63
Pixelwise IoU	**0.57**	0.54	0.51	0.56	**0.59**	0.49	0.49	0.49
Two or more objects								
Precision	**0.29**	0.45	0.47	0.44	**0.31**	0.48	0.49	0.50
Recall	**0.11**	0.25	0.36	0.17	**0.10**	0.34	0.48	0.21
F1-score	**0.16**	0.32	0.41	0.25	**0.15**	0.39	0.48	0.29
MCC	**0.16**	0.31	0.39	0.26	**0.16**	0.38	0.46	0.31
Pixelwise IoU	**0.09**	0.19	0.26	0.14	**0.08**	0.25	0.32	0.17
One or more objects								
Precision	**0.95**	0.97	0.96	0.96	**0.95**	0.95	0.96	0.94
Recall	**0.84**	0.80	0.83	0.77	**0.81**	0.75	0.80	0.69
F1-score	**0.89**	0.88	0.89	0.86	**0.87**	0.84	0.87	0.80
MCC	**0.88**	0.86	0.88	0.84	**0.86**	0.83	0.86	0.78
Pixelwise IoU	**0.81**	0.78	0.80	0.75	**0.77**	0.73	0.77	0.66

The visual evaluation of the different models showed slight differences in terms of quality (Figure 13). It was evident that the L size YOLOv5 instance was outperforming the others in this task as the predictions were the most similar in terms of the actual shapes, sizes and orientations of the objects. Based on this the L sized YOLOv5 model was concluded to be the best performing model.

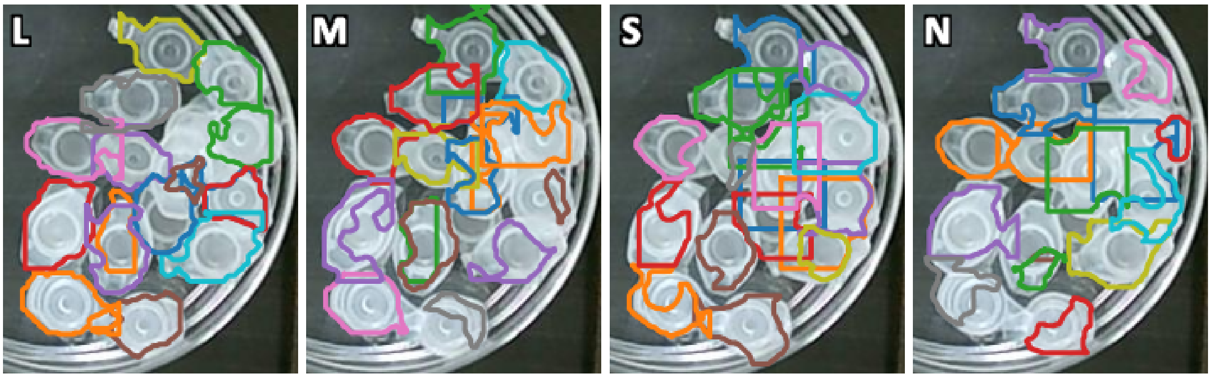


Figure 13. An example image for visual comparison of the predictions made by the YOLOv5 models trained on the layered dataset. The size of the model is noted on the top left of each image.

3.4 YOLOv5 Instance Segmentation and Overlapping Objects

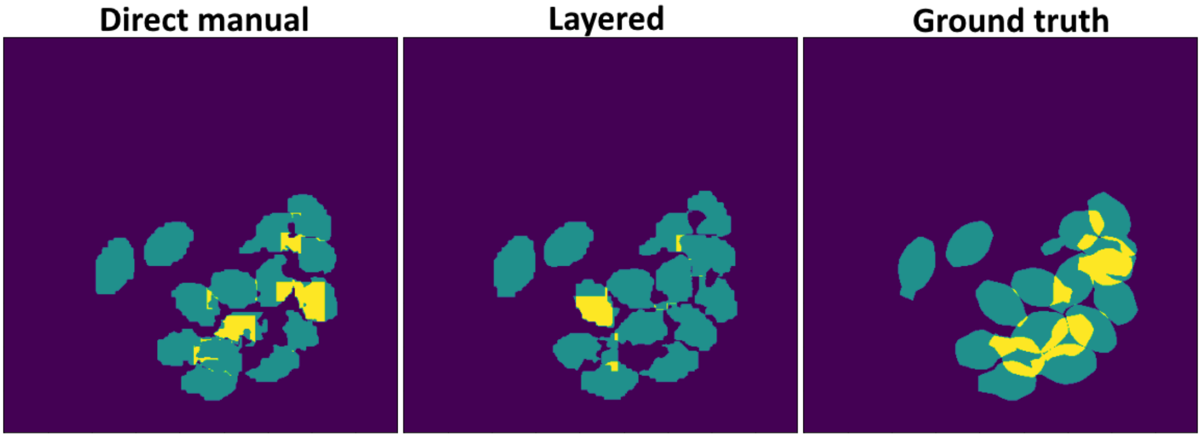


Figure 14. Visualisation of the YOLOv5 instances' results. Purple pixels are the background, cyan pixels are pixels with one object and yellow pixels are pixels with two or more objects.

YOLOv5 L models were trained on the direct manual and layered datasets. Visual comparison of the two trained models shows that neither was successful at segmenting overlapping objects (Figure 14). The model trained on the layered dataset was better at detecting the edges of the objects but it was very conservative when it came to predicting object overlaps. On the other hand, the model trained on the direct manual dataset was more liberal with predicting object overlaps. Both trained models had a tendency of predicting object segmentations that were formed from the parts of multiple different objects (Figure 15). These malformed object segmentations increased the pixelwise metrics but resulted in degradation of segmentation morphology. It is likely that the training dataset was too small for the YOLOv5 models to converge to a stable result in the segmentation task. This also shows that the development of more dedicated metrics is necessary for future extensions of this work.

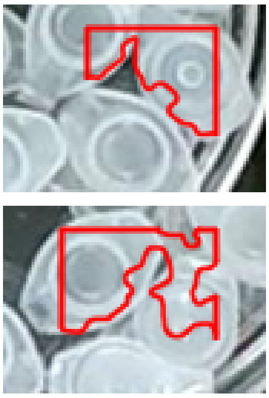


Figure 15. Examples of instance segmentations malformed from multiple objects.

3.5 Comparison of Manual and Automatic Ground Truth Generation

According to the pixelwise metrics the annotations generated from the predictions of the YOLOv5 are high in quality reaching even the levels of agreement between the two ground truth generation methods. Based on previous results (Table 4) it may be hypothesised that more advanced models may give even higher quality results.

Table 4. The pixelwise metrics of annotations generated by YOLOv5 against manual annotations.

Automatic (YOLOv5) vs Manual (Layered)					
One object		Two or more objects		One or more objects	
Precision	0.92	Precision	0.17	Precision	0.95
Recall	0.88	Recall	0.30	Recall	0.95
F1 score	0.90	F1 score	0.22	F1 score	0.95
MCC	0.89	MCC	0.22	MCC	0.95
Pixelwise IoU	0.82	Pixelwise IoU	0.12	Pixelwise IoU	0.91

The annotations created by the predictions are sufficient for usage as ground truth by themselves as they surpass the quality of the overlap prediction models. In practice, transfer learning could be applied to further improve the quality of automatic ground truth generation utilising the full extent of the available data. Small errors are still visible but they are localised to the lower right and upper left part of the objects which could be caused by imperfections in the alignment of the ground truth.

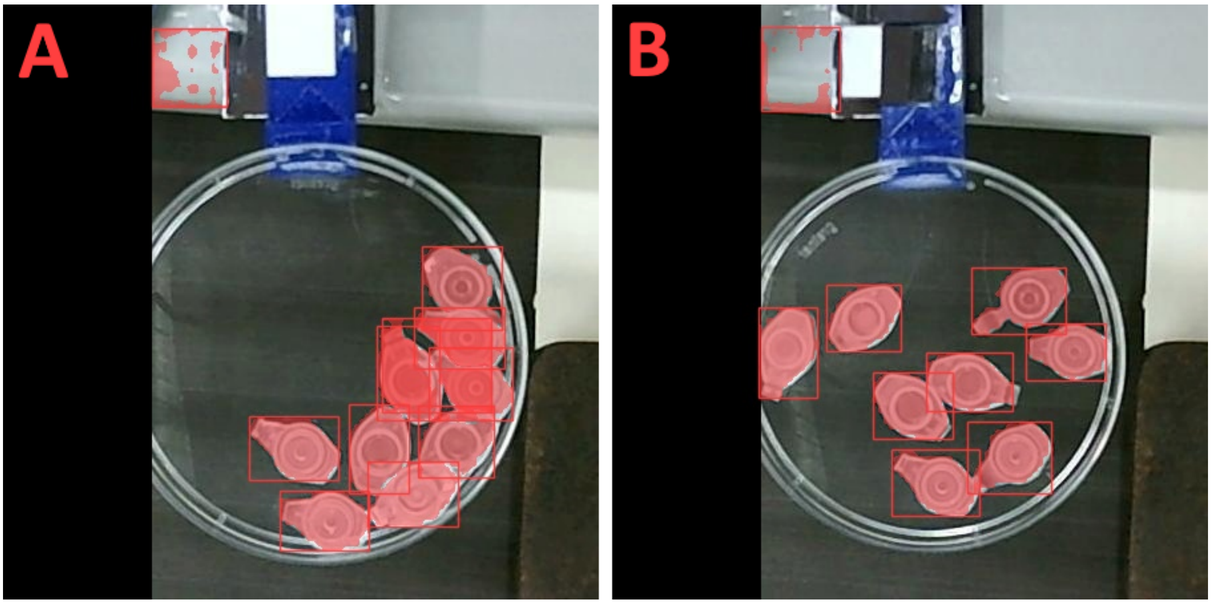


Figure 16. Two example images taken from the results generated by YOLOv5. A) An image containing a false positive and an example of how objects may have more than one annotation to represent them. B) An image containing a false positive and showcases of YOLOv5's capabilities regarding instance segmentation on objects without overlap. Metrics were calculated using only the pixels on the Petri dish.

Conclusions and Future Perspectives

The detection and segmentation of overlapping objects on images is a prevalent problem in CV. Objects overlapping with each other is a challenging segmentation task, causing CNNs to make predictions in an area it can not directly observe. A general solution to this problem has not been proposed in the scientific literature.

The main goals of this thesis were to develop a new layered data acquisition approach for images with overlapping objects, to compare how the layered approach affected data quality in comparison to the direct manual approach, to observe how the different datasets affected different CNNs predictive quality, and to see if this new data acquisition approach could be automated.

A robotic system was devised to acquire data. Using the robotic system a set of images were generated from two layers of objects. This resulted in three sets of images: overlapping layers, the top object layers, the bottom object layers. Two datasets were created based on these image sets.

The performed experiments showed that the direct ground truth generation method has substantial faults as was hypothesised. U-Net generally outperforms YOLOv5 models but is not usable for object detection. This opens the avenue to combine the YOLOv5 and U-Net models to obtain a combined pipeline, which is likely to outperform the individual models. It was discovered that the current datasets probably did not represent the worst case scenario as a low number of overlapping pixels as well as highly regular objects still allowed to generate reasonable predictions using the direct approach. This, however, may not be true for all possible use cases. It was also evident that the models mostly relied on low level features indicating the need for larger datasets in the future to find the true limitations of this method. The automated ground truth generation gave promising results and could be applied in the future for the generation of such large datasets.

The thesis has large future perspectives. For example it would be possible to study more heterogeneous objects with variable size, shape, and transparency. It would also be possible to evaluate a large number of different models as well as the influence of dataset size on the model's performance. The large datasets themselves could be compiled into open-source benchmarks as there is currently a clear lack of such datasets.

References

[1] IBM. What is computer vision? https://www.ibm.com/topics/computer-vision (21.04.2023)

[2] Janai J, Güney F, Behl A, Geiger A. Computer vision for autonomous vehicles: Problems, datasets and state of the art. Foundations and Trends® in Computer Graphics and Vision. 2020 Jul 5;12(1–3):1-308.

[3] Mahendran R, Jayashree GC, Alagusundaram K. Application of computer vision technique on sorting and grading of fruits and vegetables. J. Food Process. Technol. 2012 Dec;10:2157-7110.

[4] Zeng Z, Xie W, Zhang Y, Lu Y. RIC-Unet: An improved neural network based on Unet for nuclei segmentation in histology images. Ieee Access. 2019 Feb 1;7:21420-8.

[5] Esteva A, Chou K, Yeung S, Naik N, Madani A, Mottaghi A, Liu Y, Topol E, Dean J, Socher R. Deep learning-enabled medical computer vision. NPJ digital medicine. 2021 Jan 8;4(1):5.

[6] Nelson, Joseph 2020. How to Select the Right Computer Vision Model Architecture https://blog.roboflow.com/yolov3-vs-mobilenet-vs-faster-rcnn/ (06.05.2023)

[7] Hafiz AM, Bhat GM. A survey on instance segmentation: state of the art. International journal of multimedia information retrieval. 2020 Sep;9(3):171-89.

[8] LeCun Y, Bengio Y, Hinton G. Deep learning. nature. 2015 May 28;521(7553):436-44

[9] Mukherjee D, Jonathan Wu QM, Wang G. A comparative experimental study of image feature detectors and descriptors. Machine Vision and Applications. 2015 May;26:443-66.

[10] IBM. What is a neural network? https://www.ibm.com/topics/neural-networks (06.05.2023)

[11] IBM. What is Supervised Learning? https://www.ibm.com/topics/supervised-learning (06.05.2023)

[12] O'Shea K, Nash R. An introduction to convolutional neural networks. arXiv preprint arXiv:1511.08458. 2015 Nov 26.

[13] Lee HS, Kim K. Simultaneous traffic sign detection and boundary estimation using convolutional neural network. IEEE Transactions on Intelligent Transportation Systems. 2018 Mar 8;19(5):1652-63.

[14] Mahyari TL, Dansereau RM. Deep learning methods for image segmentation containing translucent overlapped objects. In2019 IEEE Global Conference on Signal and Information Processing (GlobalSIP) 2019 Nov 11 (pp. 1-5). IEEE.

[15] Toda Y, Okura F, Ito J, Okada S, Kinoshita T, Tsuji H, Saisho D. Training instance segmentation neural network with synthetic datasets for crop seed phenotyping. Communications biology. 2020 Apr 15;3(1):173.

[16] Zafari S, Eerola T, Sampo J, Kälviäinen H, Haario H. Segmentation of overlapping elliptical objects in silhouette images. IEEE Transactions on Image Processing. 2015 Oct 19;24(12):5942-52.

[17] Andrews, Gerard 2021. What Is Synthetic Data? https://blogs.nvidia.com/blog/2021/06/08/what-is-synthetic-data/ (06.05.2023)

[18] Papas M, Jarosz W, Jakob W, Rusinkiewicz S, Matusik W, Weyrich T. Goal‐based caustics. InComputer Graphics Forum 2011 Apr (Vol. 30, No. 2, pp. 503-511). Oxford, UK: Blackwell Publishing Ltd.

[19] Hu A, Murez Z, Mohan N, Dudas S, Hawke J, Badrinarayanan V, Cipolla R, Kendall A. FIERY: future instance prediction in bird's-eye view from surround monocular cameras. InProceedings of the IEEE/CVF International Conference on Computer Vision 2021 (pp. 15273-15282).

[20] Ronneberger O, Fischer P, Brox T. U-net: Convolutional networks for biomedical image segmentation. InMedical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18 2015 (pp. 234-241). Springer International Publishing.

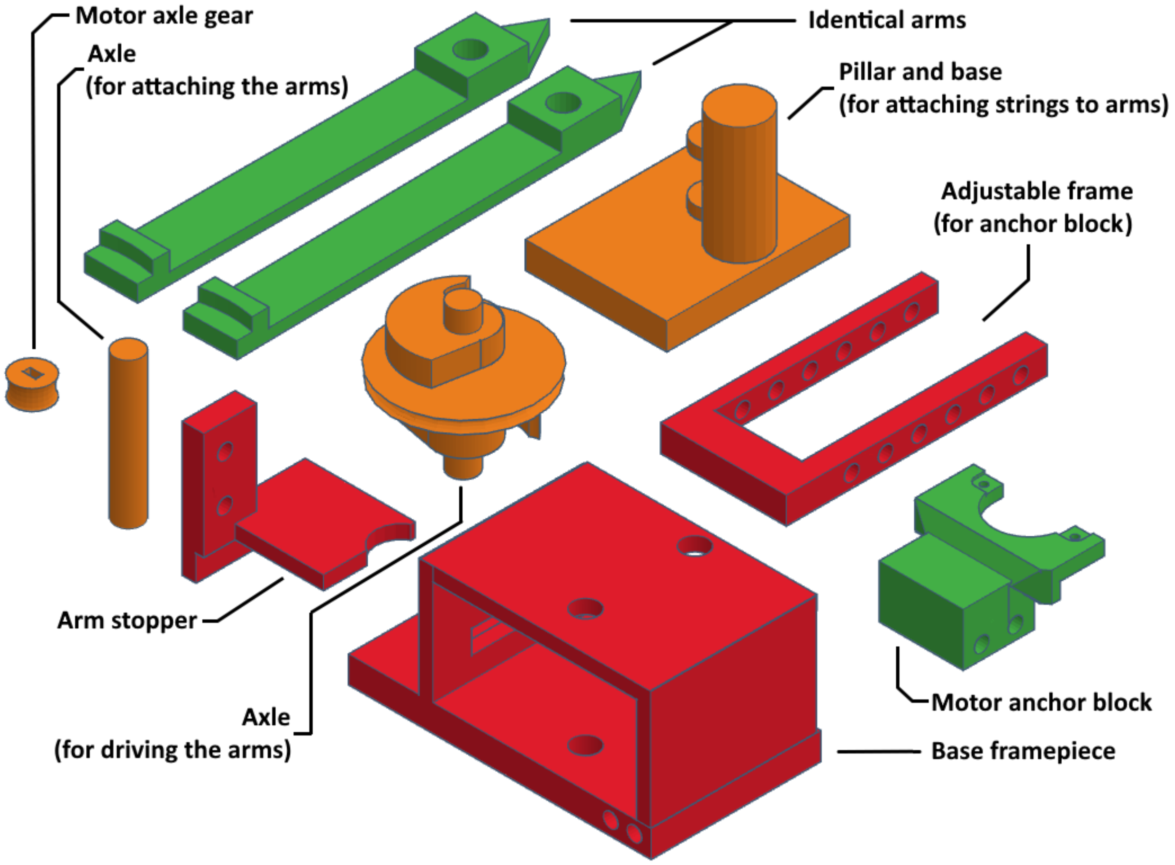
[21] Fishman D, Salumaa SO, Majoral D, Laasfeld T, Peel S, Wildenhain J, Schreiner A, Palo K, Parts L. Practical segmentation of nuclei in brightfield cell images with neural networks trained on fluorescently labelled samples. Journal of Microscopy. 2021 Oct;284(1):12-24.

[22] Jocher G, Chaurasia A, Stoken A, Borovec J, Kwon Y, Michael K, Fang J, Yifu Z, Wong C, Montes D, Wang Z. ultralytics/yolov5: v7. 0-YOLOv5 SOTA Realtime Instance Segmentation. Zenodo. 2022 Nov.

[23] Solawetz, Jacob 2020. What is YOLOv5? A guide for beginners. https://blog.roboflow.com/yolov5-improvements-and-evaluation/ (06.05.2023)

Appendix

I. 3D Modelled and Printed Parts of the Robotic System



II. Data Annotation Guidelines

The following list is a short list of guidelines which were followed during data annotation:

- Cover as much of the object as possible without including background pixels.
- Firstly, pinpoint all easily visible occluded objects and annotate them.
- Secondly, annotate all the topmost objects.
- Thirdly, if the annotated object count is less than the known object count, make the best reasonable guesses for any missed occluded objects.
- Overall, when annotating occluded objects, use best judgement about what the object's shape, size and orientation is.

III. Licence

Non-exclusive licence to reproduce the thesis and make the thesis public

I, Karl Suurkaev,

1. grant the University of Tartu a free permit (non-exclusive licence) to reproduce, for the purpose of preservation, including for adding to the DSpace digital archives until the expiry of the term of copyright, my thesis,

Seeing the forest behind the trees: A novel method for generating data for overlapping object segmentation

supervised by Tõnis Laasfeld, Kaspar Hollo, Dmytro Fishman.

2. I grant the University of Tartu a permit to make the thesis specified in point 1 available to the public via the web environment of the University of Tartu, including via the DSpace digital archives, under the Creative Commons licence CC BY NC ND 4.0, which allows, by giving appropriate credit to the author, to reproduce, distribute the work and communicate it to the public, and prohibits the creation of derivative works and any commercial use of the work until the expiry of the term of copyright.
3. I am aware of the fact that the author retains the rights specified in points 1 and 2.
4. I confirm that granting the non-exclusive licence does not infringe other persons' intellectual property rights or rights arising from the personal data protection legislation.

Karl Suurkaev

09/05/2023