computer vision
& data science

Detection of Aphids on Sticky Plates using YOLOv5 with Image Tiling

NHL Stenden Lectoraat in Computer Vision & Data Science

Sander Klut

Supervisors: Henry Maathuis, Maya Aghaei Gavari

Abstract—The health of seed potato crops is being threatened by viruses like the Potato Virus Y (PVY). In the last five years, more than 15% of harvested seed potatoes and sugar beans were infected with the Potato Virus Y. There is an empirical relationship between the number of aphids and the number of infected plants [1]. Early detection of aphids creates the opportunity to instantly use pesticides. Currently, manually counting aphids is very time-consuming and the use of Convolutional Neural Networks, to detect aphids and accordingly decide the necessity of applying pesticides, could offer a solution. This study considers a single-stage object detection model for the real-time detection of aphids on yellow sticky plates with YOLOv5. A dataset has been acquired with images of yellow sticky plates with aphids and other insects. Because the aphids are tiny, images with a high resolution are being used. To reduce the needed computing power, tiles are selected from the images which are then processed by the model. The network is competent in selecting tiles that always contain either an aphid or an insect, named positive tiling. The YOLOv5l-model with positive tiling is the best-performing model with an F1-score of 0.525. Possible extensions of this study are being discussed, together with suggestions for future research.

Index Terms—Deep Learning, YoloV5, Insect Detection, Aphids, Agriculture, Yello Sticky Plates, Image Tiling

1 INTRODUCTION

According to Projectplan POP3+ Fryslân: Innovation in Aphid Detection [2], research of Wageningen University & Research (WUR) has shown that the health of seed potato crops is being threatened by viruses. Viruses like the Potato Virus Y (PVY) cause economic harm and jeopardize the strong export position of Dutch seed potatoes and sugar beans.

To maintain this strong export position the health and quality of the product are paramount. In the last five years, more than 15 percent of harvested seed potatoes and sugar beans were infected with the PVY virus [1]. This diminishes the value of the product and thus results in less profit. It is believed that viruses such as PVY are spread by insects, with aphids in particular. This assumption is supported by the direct relationship between the number of aphids and the percentage of infected crops in the field [1].

Early detection of aphids creates the opportunity to use pesticides locally when needed, resulting in better protection of the crops and less use of pesticides. Here the speed of the detection process is crucial, since the control of the aphids needs to happen before the further spreading of deceases.

Currently, the process of detecting aphids is time-consuming and not as efficient as it could be. With yellow sticky plates, insects are attracted and caught, after which the aphids are manually identified and counted. The results of these counts determine the number of pesticides used for the entire field, while aphids could be only present in certain regions. Here the use of Convolutional Neural Networks, to detect aphids and accordingly decide the necessity of applying pesticides, could be a solution.

For object detection with Convolutional Neural Networks, there are two main approaches that can be considered. The single-stage-models approach performs the detection of objects and the classification at the same time, and the two-stage-models approach has an object detection model and a different classification model. In this study, the single-stage approach is being used.

The aim of this study is to design a single-stage Object Detection and Classification Model able to detect and identify insects and aphids on yellow sticky plates. Hence, the main research question is the following:

Can a single-stage object detection and qualification model be used to robustly find aphids on yellow sticky plates?

The professorship of Computer Vision Data Science has collected a dataset containing aphids and insects, the 'Aphids and Other Insects' dataset. The dataset consists of high-resolution images that take a large amount of computing power to process these images. Image tiling can be used to crop tiles from the images, which are then processed.

Another possible difficulty with the dataset is that there is an imbalance between the number of aphids and insects. The effect of this imbalance will be studied.
Therefore, the sub-questions are the following:

- *Does image tiling benefit the detection of tiny insects?*
- *Would an imbalanced dataset influence the detection results of tiny insects per class?*

- *Sander Klut is a Mechanical Engineering student, doing a minor in Computer Vision and Data Science at the NHL Stenden University of Applied Sciences, E-mail: Sander.klut@student.nhlstenden.com.*
- *Henry Maathuis is a researcher at the NHL Stenden Lectoraat in Computer Vision & Data Science, Email: henry.maathuis@nhlstenden.com.*
- *Maya Aghaei Gavari is a researcher at the NHL Stenden Lectoraat in Computer Vision & Data Science, E-mail: maya.aghaei.gavari@nhlstenden.com.*

2 State of the Art

The target of this study is to create a real-time network based on YOLOv5 specified in the detection of small objects. Section 2.1 discusses numerous object-detecting networks based on YOLOv5 that have been established.

A study in detecting tiny pests from sticky traps is shown in section 2.2. As well as research being done on regional analysis such as the power of tiling for small object detection, shown in section 2.3.

2.1 Object detection with YOLOv5

In [3], the study on the single-stage object detection and classification model, YOLO is presented. Previous models were dependent on a two-stage approach in which, first, regions of interest are found after which a classifier processes these regions. You Only Look Once(YOLO) is a single model able to detect as well as classify objects. The network is now known for good performance in real-time object detection algorithms.

More recently, in 2020 a new version of YOLO was released, the YOLOv5 model [4]. While not written by the same author, since the network is based on the other YOLO versions and the same characteristics are applicable, the model is assumed to be its successor. YOLOv5, the more recent version of YOLO, has been shown to provide good performance in real-time object detection [5] [6] [7].

2.2 Field detection of tiny insects

In [8], a network using RCNN-models was constructed to detect thrips and whiteflies on yellow sticky plates in a greenhouse. The network was designed for real-time monitoring of the insect population in the greenhouse. The best-performing model of the study was the Tpest-RCNN-model, consisting of pest feature learning with VGG16, Region of Interest proposals, and classification.

2.3 Tiling for small objects

In ‘The Power of Tiling for Small Object Detection’ researchers have tried to identify pedestrians and traffic real-time with images from a micro aerial vehicle (MAV)[9]. For use on mobile GPU’s a balance must be sought between higher accuracy and less computational power. For the detection of pedestrians and vehicles onboard a MAV, with high-resolution imagery, PeleeNet is considered to be the most efficient on mobile GPUs.

Training models with high-resolution images is not efficient since the model is asked to analyze empty parts of images as well. The tinier the object to identify, the higher the resolution of the image should be to get an accurate model. To establish a fast-performing but accurate model, image tiling has shown to be a solution. [8] [9] [10]. Image tiling requires extra processing compared to using raw images, however, a lot less data needs to be loaded and therefore the model has a faster performance.

3 Materials and Methods

The requirements for the study are presented in this section, together with the specific techniques that have been applied.

The dataset of the professorship, the ’Aphids and Other Insects’ dataset, is discussed in 3.1. 3.2 explains what is considered the ground truth and how the data has been divided. The pre-processing of the data is discussed in 3.3, and the use of object detection models in 3.4. In 3.5, the used hardware is stated. How the results are shown, is discussed in 3.6.

3.1 Dataset Acquisition

In preparation for this study, the professorship has carried out experiments at seed potato farms across Friesland, the Netherlands. Since aphids and other insects appear to be attracted to bright yellow colors, yellow sticky plates were placed at the farms to catch these insects.

A dataset of images of yellow sticky plates has been collected by the professorship, which contains both aphids and other insects. For four months, every two weeks these yellow sticky traps were collected and placed at ten Frisian farms, resulting in 80 yellow sticky plates. The collected traps were used for acquiring images.

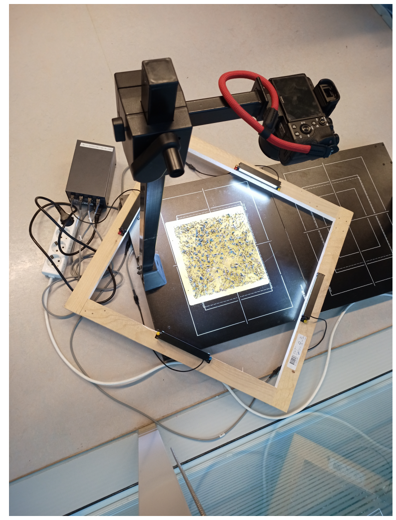


Fig. 1: Camera Setup for image acquisition

In a controlled environment, images of the yellow sticky traps were taken with a SONY a7 MARK II SLR-camera. An image of each quarter of the yellow sticky plates was captured with the camera creating images with a resolution of 6000 x 4000 pixels, creating four individual images per yellow sticky plate. Resulting in a total of 320 images. The yellow sticky plates that have been used are 250x250 mm. Other specifications regarding the camera are shown in table 1.

Table 1: Camera Specifications

Component:	**Specifications:**
Camera	Sony A7II
Camera lens	Tamron F053S
Camera Resolution	6000x4000 pixels
Sensor Size	35.8x23.9 mm
Color depth/Bitrate	14-bit uncompressed Raw
Illumination	Low angle illumination

3.2 Ground Truth and Dataset division

3.2.1 Ground Truth

To create square images, all images captured by the camera have been cropped. The images of 6000x4000 pixels, have been cropped to a wanted size of 4000x4000 pixels.

After cropping, the data was annotated to create a ground truth. Annotation software was used to annotate the insects by creating a bounding box around the surface of the insects[11]. After which the insects were manually annotated as an ‘Aphid’ or ‘Non-aphid’. The bounding boxes and class names in combination with the images are considered the ground truth. Object detection methods use this to train a model.

3.2.2 Train, Validation and Test Dataset

From the images combined with their annotations, the ’Aphids and Other Insects Dataset’ has been established. The images have been divided into three groups: a group for training, one for validation, and one for testing.

The division of the dataset ensures the network will perform well on data it has never seen before. If the validation and test are already shown to the network it will perform better on this specific data, but

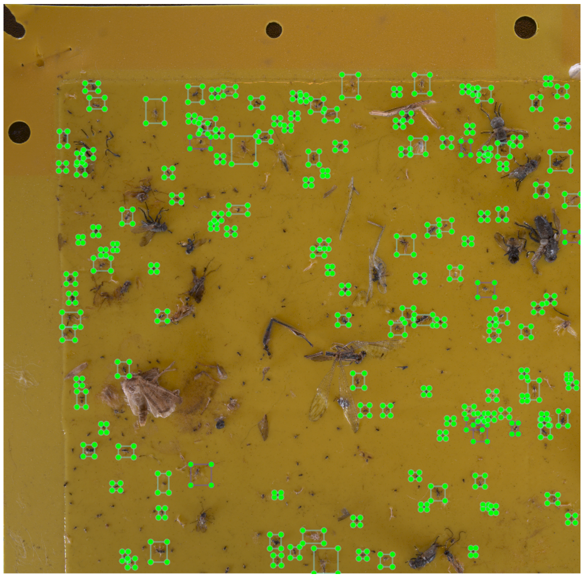


Fig. 2: Image with annotations made with LabelImg annotation software.

probably worse on data, it has never seen before. The principle were the network is performing well only because it has seen the specific data before, and only on this data, is called overtraining.

For the purpose of finding the effect of the annotations on the results, another two groups are introduced in which the objects are again annotated, only now following other annotating rules. In the qualitative results of the general dataset, misclassifications in the annotations were noted, as well as annotations bigger than the objects. Insects with parts missing were not annotated. In the newly annotated data, all misclassifications have been adjusted and all insects were annotated.

The number of images per group and amount of aphids and non-aphids is shown in Table 2.

- Four tiles from the validation group are again annotated for End-Validation
- Four tiles from the Testing group are again annotated for the Improved Testing group.

The larger testing dataset is now neglected and the better-annotated End-Validation and Improved Testing groups are used instead.

Table 2: Dataset division

Model	Images	Non-Aphids	Aphids
Training	102	3302	402
Validation	47	1427	181
Testing	22	1200	136
End-Validation	4	369	50
Improved Testing	4	369	50

3.3 Pre-processing

3.3.1 Tiling

Because images with a high resolution are being used, tiles, which are smaller cutouts or crops, are selected from the images which are then processed by the model. Because of the small size of the insects, aphids and non-aphids are difficult to distinguish. High-resolution images, taken from a close distance of approximately 500 mm contain a lot of detail of the insects, making it easier for the models but processing these images does require a lot of computing power. Images containing more detail can be processed with the same computing power by selecting tiles from the images, instead of going over whole plates. The tile size for the baseline is set at 640x640 pixels.

The possibility to create specific tiles containing at least one aphid or non-aphid, has been integrated. Without this, the images contain regions with no insects during training, tiles could be selected without any aphids or non-aphids on them. If this happens often, it prevents the network from learning what an aphid or non-aphids looks like. To overcome this problem, the coordinates of a bounding box from the ground truth are picked and a tile around it is created. Thereby resulting in tiles always containing an aphid or a non-aphid.

This also introduces the possibility of balancing the number of aphids versus non-aphids seen by the model during training. The aphid dataset contains approximately ten times as many insects as aphids, meaning the dataset is imbalanced. A probability for sampling can be chosen to counteract this imbalance. The probability decides how often the network should create a tile containing an aphid, or a tile containing either a non-aphid. When an image is chosen with no aphids, a tile with an insect is created.

Because all created tiles can contain more than just one aphid or insect, a class probability of 50 percent still results in an imbalance. For the baseline, the probability is set at 50 percent. If an image contains insects and aphids, the chance the model picks either a tile with an insect or a tile with an aphid is 50 percent. Some images contain no aphids, in that case, the model always picks a tile with an insect.

3.3.2 Augmentations

Furthermore, the possibility for the use of augmentations is added, expected to create a more reliable model in uncontrolled environments and enlarge the dataset. Augmentations modify an original image slightly, creating a new image with the same characteristics but differing from the original. Because of this, the same images look different every time they go through training.

Horizontal and vertical flipping is considered, next to a color jitter and applying noise multipliers. Table 3 shows the hyperparameters of the color jitter. Table 4 shows those of the multiplicative noise.

Table 3: Hyperparameters for color jitter

Color Jitter:	Brightness	Contrast	Saturation	Hue
Value:	0.02	0.02	0.02	0.02

Table 4: Hyperparameters for multiplicative noise: min_mult = minimum multiplier, max_Mult = maximum multiplier.

Multiplicative Noise	min_Mult	max_Mult	Random sample
Value:	0.98	1.02	True

3.3.3 Normalising Images

The images are normalized to values between 0 and 1. For 16-bit images, this is done by subtracting mean = 0 and dividing by standard deviation as shown in equation 1.

$$normalized\_image = \frac{(original\_img - mean)}{(standard\_deviasion)} \tag{1}$$

3.4 Object Detection Model

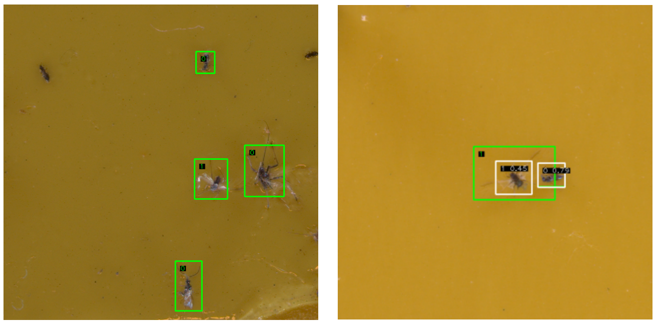
3.4.1 YOLOv5

Several YOLOv5 models are available. In this research, the models YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x are considered.

The YOLOv5 models are pretrained on the Common Objects in Context [12] dataset. During pre-training, images are used with a size of 640x640 pixels. Because of this, a tile size of 640x640 could turn out to be ideal in training.

3.4.2 Training & Validation

Random tiles are created during training & validation. The tile size is one of the hyperparameters, standard set at 640x640 pixels. Figure 3a shows a tile created in training. Figure 3b shows a tile created in validation with the made predictions in white. Every two epochs of training, the model will validate whether learning has improved. The other hyperparameters are the number of tiles per image, batch size, learning rate, and the patience of the scheduler. For the baseline, the following values are used: Number f tiles of 2 tiles per image, a batch size of 2, a learning rate of 0.001, and no patience.



(a) **Cropped tile with annotations in green during training.** (b) **Cropped tile with ground truth in green + predictions in white.**

Fig. 3: Cutout tiles in training and validation

3.4.3 Testing

Images are divided into fixed tiles and are run through testing. The tile size used for testing is 640x640 pixels. The tile size should always be the same in training and testing. The image with Bounding Boxes of the Ground Truth and the Predictions is reconstructed and saved. Figure 4 shows a testing image with ground truth in green and predictions in white. The yellow lines show the borders of the individually analyzed tiles.

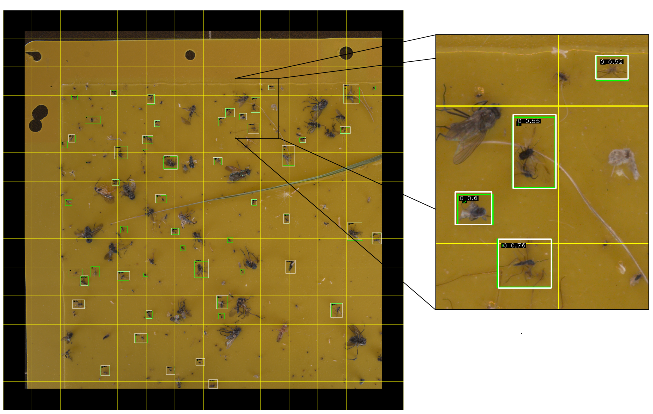


Fig. 4: Object detection and classification during Testing. Green = Ground truth, White = Predictions

3.5 Hardware

The machine used to carry out experiments consists of an Intel® Core(TM) i9-7960X CPU @ 2.80GHz (8 cores) CPU and an NVIDIA® GeForce RTX 2070(8.000 Gigabyte Memory) GPU.

3.6 Evaluation Metrics

The performance of the models is evaluated with intersection over union(IoU). With IoU, the ground truth bounding box is compared to the predicted bounding box. A threshold of 0.5 is used, meaning an intersection of 50 percent between the ground truth- and the predicted bounding box is needed to label the prediction as a correct one.

To compare the performance of the models, the precision, recall, F1-score, and mAP-score(Mean Average Precision), on the aphids and Non-aphids, are considered. As well as the precision, recall, and F1-score on aphids only. A confusion matrix has been made for each class to identify where the models are struggling the most. With the data results of the performance, choices can be made between models.

During the experiments, the decision has been made that above all the precision, recall, and F1-score regarding only aphids should be considered. The results regarding only aphids are considered to be more useful since they will show the feasibility of detecting aphids with the models. The models are compared using F1-score on aphids, showing the overall strength of the models in this class.

Precision is the score for the ratio between the number of correct predictions and the total nu of predictions.TP being the True Positives, meaning the fraction of Aphids detected over all objects. FP being the False Positives, meaning non-aphids detected as aphids. Equation 2

$$Precision = \frac{TP}{TP+FP} \tag{2}$$

In Equation 3 the equation for recall is shown. Recall is the score for the ratio between the number of correct predictions and the total number of the ground truth. TP being the True Positives, FN being False Negatives.

$$Recall = \frac{TP}{TP+FN} \tag{3}$$

The F1-score arises from the average between the precision and the recall. As shown in Equation 4

$$F1 = \frac{2*Precision*Recall}{Precision+Recall} \tag{4}$$

The confusion matrices show the four values for each model for True Positives(TP), True Negatives(TN), False Positives(FP), and False Negatives(FN). For each class, there is a matrix. Table 5 shows the matrix for aphids. Table 6 shows the matrix for other insects.

Table 5: Confusion Matrix: Aphids

	True Aphid	**No Aphid**
Predicted Aphid	TP	FP
Predicted No Aphid	FN	TN

Table 6: Confusion Matrix: Insects

	True Insect	**No Insect**
Predicted Insect	TP	FP
Predicted No Insect	FN	TN

4 EXPERIMENTS AND RESULTS

Experiments have been executed on the influence of different hyper-parameters and experimental settings, of the end-to-end tiled YOLOv5 Tiled Object Detection model, on the 'Aphids & Other Insects Dataset'. The first experiment compares the different YOLO-models. YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x are compared. In the second experiment, different parameters for training and validation are considered, being the batch size, learning rate, and patience. In the third experiment, the effect of batch size, tile size, tiling, and use of augmentations is shown. The last experiment considers using different probabilities for positive tiling. The number of tiles per image of 2 is used for all experiments.

4.1 Experiment 1: Different YOLO-models

Four different YOLOv5-models, YOLOv5s, YOLOv5m, YOLOv5l, YOLOv5x, are compared. All hyper-parameters are equal for the four models, having a batch size of 2, the number of tiles per image is 2, and a learning rate of 0.0001. The experiments have been running for 100 epochs. No scheduler was used in this experiment. A summary of the experiment is shown in Table 7.

Table 7: Experiment 1: considering different YOLO-models, Batch = Batch size, Learning rt = Learning rate

	Model	Batch	No. Tiles	Learning rt.	Epochs
YOLOv5	s/m/l/x	2	2	0.0001	100

4.1.1 Results Experiment 1

Table 8 shows the results of experiment 1. From the F1-score in the table can be concluded that larger YOLOv5-models are outperforming the smaller models on the 'Aphids & Other Insects Dataset'. This is as expected since the larger YOLOv5-models are in fact larger algorithms requiring also more computing power.

Although this would suggest the use of the largest model, the size of the model is, due to available computational power, also inversely proportional to the batch size. This is further discussed in experiment 2. All models will still be considered in experiments 2 and 3.

Table 8: Results Table 1: Aphids and Non-aphids

Model	Precision	Recall	F1-Score
YOLOv5s	0.76	0.67	0.71
YOLOv5m	0.75	0.68	0.72
YOLOv5l	0.78	0.67	0.72
YOLOv5x	0.76	0.70	0.73

4.2 Experiment 2: Basic hyperparameters

Since the different YOLOv5-models are expected to deliver different results, depending on different parameters, all YOLO-models have been compared in combination with batch size, learning rate, and patience. Patience introduces a scheduler that reduces the learning rate when the model is not improving for a given number of epochs, with patience being the number of epochs. The batch size in combination with tile size is limited by the amount of memory of the GPU. In this study, the maximum of the (batch size * the number of tiles for each model being: YOLOv5s-model<=16, YOLOv5m-model<=8, YOLOv5l-model<=4, YOLOv5x-model<=4. For all experiments, a number of tiles of 2 per image has been used.

The best batch size, learning rate, and patience will be determined by the experiment. The learning rate and patience will be used in experiment 3. All batch sizes are still considered in experiment 3, due to find if there is a relation between batch size and tile size.

YOLO Model	Batch Size	Learn Rate	Patience
s/m/l/x	2/4/8	0.01/0.001/0.0001	0/20/40

4.2.1 Results Experiment 2

The best results of experiment 2 are shown in table 9. All best results have a learning rate of 0.001 and a patience of 20 epochs.

In table 9 can be seen that larger batch sizes result in higher F1-scores. At a certain point, the network has seen most of the aphids in the dataset at least once. A possible explanation for the results could therefore be that the network is beginning to overtrain. Overtraining is the principle where the network is trained too specifically on the data at hand during training. Because of this, the network would perform worse on data it has never seen before.

Table 9: Results experiment 3. Pre_A = Precision Aphids, Rec_A = Recall Aphids, F1_A = F1-score Aphids.

Model	Batch	Epochs	Pre_A	Rec_A	F1_A
YOLOv5s	2	400	0.382	0.559	0.454
	4	400	0.449	0.419	0.433
	8	400	0.430	0.418	0.424
YOLOv5m	4	400	0.431	0.520	0.471
YOLOv5l	2	400	0.455	0.549	0.485

4.3 Experiment 3: Tile size, Tiling & Augmentations

The best YOLO-models, learning rate, and patience have been used as the baseline for experiment 3. This means that the batch size is still a variable. 24 experiments have been carried out differing in batch size, tile size, tiling, and augmentations.

Table 10: Experiment 3

Batch size	Tile size	Tiling	Augmentation
2/4/8	320/640	Tiling/Positive Tiling	No/Yes

4.3.1 Results experiment 3

The results are shown in table 11. YOLO-models YOLOv5m and YOLOv5l are used in combination with a learning rate of 0.001 and patience of 20 epochs.

As can be concluded from table 11, tiles with a size of 640x640 pixels seem to be preferable above tiles with a size of 320x320 pixels. The YOLOv5l-model with a batch size of 2 seems to outperform all other experiments with an F1-score on aphids of 0.525. Furthermore, no advantages have been found in the use of augmentations.

4.4 Experiment 4: Probability

Experiment 4 considers the change in class probability. During the tiling process, one of the hyperparameters is the class probability, deciding how often a tile is created around an aphid and how often around a non-aphid. A 50% probability for each is compared with a 100% probability of creating a tile with an aphid. The 100% probability counters the imbalance between the number of aphids and non-aphids seen by the network.

4.4.1 Results experiment 4

While it was expected that removing the imbalance by changing the probability would result in better performances of the network, table 12 shows that there is no difference between the results of the different probabilities. However, during these experiments, another interesting point came forward. When looking at the F1-scores on aphids, the probability of 100% aphids, seems to train much faster regarding the detection of aphids.

Table 11: Results experiment 3. Pre_A = Precision Aphids, Rec_A = Recall Aphids, F1_A = F1-score Aphids.

Model	Batch	Tile	Aug	Prec_A	Rec_A	F1_A
m	2	640	No	0.488	0.463	0.475
		640	Yes	0.576	0.390	0.465
	4	640	No	0.492	0.456	0.473
		640	Yes	0.389	0.478	0.429
		320	No	0.416	0.507	0.457
l	2	640	No	0.473	0.588	0.525
	2	640	Yes	0.593	0.397	0.476

Table 12: Experiment 4

YOLO-model	Batch size	Probability
s/m/l/x	2/4/8	50/50 , 100/0

5 Discussion, Conclusion and Future Work

This section discusses the main findings of the study and states the conclusions that have been drawn from these findings, after which possible options for future research are given.

5.1 Discussion

The study suggests that the use of larger YOLO-models results in better performances on the 'Aphid Dataset'. Overall, a larger batch size results in better performances and faster training of the network, as shown in table 8 and 9. The batch size in combination with the number of tiles per image is limited by the amount of memory of the GPU.

It is believed that even higher batch sizes in combination with the YOLOv5x-model would result in even better performances. To accomplish these results, more computing power is needed, as well as a larger dataset. From the fact that larger batch sizes result in faster training, in combination with the occurrence of better performances on non-aphids with larger batch sizes at the end of the training, the assumption is that the network needs more aphids to train on.

The study has also shown that positive tiling, where only tiles are selected containing either an aphid or a non-aphid, outperforms regular tiling.

The use of augmentations has not proven to benefit the results on the 'Aphid dataset', as can be concluded from table 11. Further experiments must show if there are other augmentations than the ones discussed in the paper, that do benefit the results. More test results on this part would especially be useful since augmentations can counter the disadvantages of a small dataset.

Concerning the main hyperparameters, a learning rate of 0.001 and a scheduler with patience of 20 epochs have been shown to give the best results in this setting (table 9). A tile size of 640x640 pixels is preferred over a tile size of 320x320 pixels (table 11). During pretraining images of the same size have been used. Using a tile size with the same amount of pixels as during the pretraining of the models could be an explanation for better results.

The experiments done regarding the class probabilities in positive tiling, have shown that a balance between the number of objects per class results in faster training (table 13). Because in the remaining part of the tile, around the chosen aphid or non-aphid, there could be more aphids or non-aphids, it is difficult to perfectly balance the number of objects seen per class.

A solution could be considered which would perfectly balance the classes. Taken into account has to be that the process of tiling would still have to be random. A more obvious solution could be to create a model with only one class for aphids.

Table 13: Results experiment 4. Pre_A = Precision Aphids, Rec_A = Recall Aphids, F1_A = F1-score Aphids.

Model	Epochs	Probability	Prec_A	Rec_A	F1_A
YOLOv5s	100	50/50	0.356	0.451	0.398
	100	100/0	0.358	0.495	0.415
	400	50/50	0.430	0.418	0.424
	400	100/0	0.333	0.574	0.421

The model seems to do some false predictions on parts of insects that are stuck to the yellow sticky plates. Furthermore, regarding the 'Aphids & Other Insects Dataset' there is still room for improvement. Attempts have been done in improving the annotations for the testing data. Due to time restrictions, no further improvements to the dataset have been done. Especially in the data for training, aphids are annotated as non-aphids and the other way around.

5.2 Conclusion

This section answers the research questions. The conclusion states the main highlights of the results and relates them to the goals of the study.

This study proposes a single-stage object detection model as a solution to find aphids on yellow sticky plates. Primary, the sub-questions are answered, after which the answer to the main research question is stated.

5.2.1 What would specifically benefit the detection of tiny objects?

For the detection of tiny objects images with a high resolution have been used. Processing high-resolution images takes a lot of computing power. Hardware limitations make it impossible to process full images. For this reason, the model crops tiles from the images which are then processed. The use of high-resolution images with image tiling has given promising results. More complex algorithms seem to outperform less complex algorithms.

5.2.2 Would an imbalanced dataset influence the detection results per class?

Around farms, in general, there are a lot more insects than aphids. The collected yellow sticky plates have approximately ten times as many insects on them as aphids. This creates a natural imbalance in the dataset.

The probability of the network creating a tile with either an aphid or a non-aphid affects the ratio between classes. An imbalanced dataset is now autonomously balanced towards a desired ratio. A higher probability of a class has not been shown to increase results for this class. The higher probability does result in faster training for the specific class.

5.2.3 How could a single-stage object detection and qualification model be used to robustly find aphids on yellow sticky plates?

An end-to-end method to find aphids on yellow sticky plates has been established for use by farmers. A network of a YOLOv5-model in combination with image tiling has been shown to deliver optimistic results but not yet on a level that can be put in to use in the field. However, the feasibility of detecting aphids with the proposed methods has been shown.

The overall best result achieved in the study is an F1-score on aphids of 0.525. The model has a precision on aphids of 0.473, a recall on aphids of 0.588, and an overall F1-score of 0.458. The indication of the number of aphids in a field of crops can help farmers in their decision-making around the use of pesticides.

5.3 Future Work

This segment discusses the possibilities for future research after this study. The next steps are discussed in direct relation to this study. After which, potential future studies are considered.

5.3.1 Future work regarding this study

For future work, improving the data going into the model should be considered. A bigger training dataset is desirable. The dataset could be extended with more images or configurations for augmentations could be examined to give the idea of a larger dataset.

The annotations of the 'Aphids & Other Insects Dataset' should be considered enhancing. Instead of doing this manually, the network could also return images of objects the network is not certain about.

Due to time restrictions, no further improvements to the dataset have been done. Especially in the data for training, there is room for improvement. Some aphids are now still annotated as non-aphids and the other way around.

5.3.2 Potential future studies

Future research could be done on single-stage object detection. A model could be considered with only a class for aphids. In this study, the non-aphids class does not seem to increase performance. Furthermore, where on the model with two classes, the loss is a combined value of the aphid and non-aphid classes, is the loss in case of a one-class model only calculated of aphids.

In this study, different kinds of aphids are all considered to fit into one class. Some kinds of aphids differ from each other in color, shape, and size, potential research could be done in categorizing the aphids per kind.

Parallel to this study, a system is being developed, able to identify and analyze the aphid population with an in-flight setup at a seed potato farm. A test station has to be designed that is able to collect necessary data, such as images of videos, and send it to a remote location. From where the data can be accessed and processed by the software, specifically designed to regulate the number of pesticides used per region. The invented system will be designed in such a way it could be used to identify other insect species as well.

Acknowledgements

This project is financially supported by ELFPO and performed within the POP3+ Fryslân project Innovatie luizendetectie (Innovation in Aphid Detection). The author wishes to thank the consortium for their contributions to the study and expertise.

Europees Landbouwfonds voor Plattelandsontwikkeling: Europa investeert in zijn platteland

References

[1] Paula Westerman. Virus- en vectorbeheersing in pootaardappelen, 01 2022.

[2] PROJECTPLAN POP3+. Provincie fryslân maatregel 7: Samenwerking voor innovaties 2021, 2021.

[3] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection, 2015.

[4] Glenn Jocher. ultralytics/yolov5, 08 2020.

[5] Xiaohong Han, Jun Chang, and Kaiyuan Wang. Real-time object detection based on yolo-v2 for tiny vehicle object. *Procedia Computer Science*, 183:61–72, 2021.

[6] Yonghui Lu, Langwen Zhang, and Wei Xie. Yolo-compact: An efficient yolo network for single category real-time object detection, 08 2020.

[7] Achyut Morbekar, Ashi Parihar, and Rashmi Jadhav. Crop disease detection using yolo. *2020 International Conference for Emerging Technology (INCET)*, 06 2020.

[8] Wenyong Li, Dujin Wang, Ming Li, Yulin Gao, Jianwei Wu, and Xinting Yang. Field detection of tiny pests from sticky trap images using deep learning in agricultural greenhouse. *Computers and Electronics in Agriculture*, 183:106048, 04 2021.

[9] F. Ozge Unel, Burak O. Ozkalayci, and Cevahir Cigla. The power of tiling for small object detection, 2019.

[10] Delia M. Pinto-Zevallos and Irene Vänninen. Yellow sticky traps for decision-making in whitefly management: What has been achieved? *Crop Protection*, 47:74–84, 05 2013.

[11] Darren Tzutalin. Labelimg, 08 2022.

[12] Coco - common objects in context.