NHL STENDEN
computer vision & data science

Aphid Recognition using Multi-Scale Feature Representations with Vision Transformers

NHL Stenden Lectoraat in Computer Vision & Data Science

Fredrik-Otto Lautenbag

Supervisors: Henry Maathuis, Maya Aghaei Gavari

Abstract— In this paper, Multi-Scale and Single-Scale architectures were evaluated to improve the classification of aphids among other insects. Early aphid identification is necessary to determine whether the presence of aphids impacts the effectiveness of pesticides. The objective is to prevent the crop (particularly seed potatoes in this study) from being contaminated with viruses transmitted by aphids. In order to classify our dataset, three distinct deep-learning models (ResNet, Vision Transformers, and Cross-Attention Multi-Scale Transformers) are evaluated. In addition to the difference in scale, two designs were included: Convolutional Neural Networks and Vision Transformers. The collected dataset used during this study contains photos of aphids and other insects. Since the difference between aphids and non-aphids is negligible, the annotations were enhanced multiple times by cleaning the data in cooperation with domain experts. Prior to the comparison, grid searches are performed on all selected models to identify the optimal parameters. The Cross-Attention Multi-Scale Vision Transformer, which is based on the Vision Transformer but expanded to a Multi-Scale architecture, achieved the greatest F1-score on aphids (84.88 percent) and lowest standard deviation among multiple experiments (1.06 percent). The Multi-Scale method demonstrates applicable performance for classifying aphids. Several recommendations are made to further improve classification performance.

Index Terms—classification, aphids, insects, Vision Transformers, ViT, Cross-ViT

1 Introduction

The Netherlands has a large agricultural sector. In 2020, this sector accounted for about 18 percent of exported goods [1]. Among these exported goods are potatoes, with the majority of them being seed potatoes [2]. An export seed potato starts its growth in Dutch soil. Planting selected potatoes yield seed potatoes, which are then run through this process again. When the seed potato reaches a certain biological specification, it can be grown into a potato for consumption. The final process can take place in another country where the potato is imported. This process produces consumption potatoes, which are used to feed both people and animals. Once the seed potato has been planted abroad, problems may occur if the initial growing process in the Netherlands was disrupted. As a result, a portion of the world's food supply is reliant on the conditions present when seed potatoes are in their early stages of growth.

The main threat to the seed potato during its growth is to be infected with a virus [3]. The transmission of the virus in a seed potato batch starts to show up weeks after the potato has been harvested and stored. Before exportation to other nations, a third-party organization examines seed potatoes for possible infections. Only a few samples from a batch are tested for infections because it is not feasible to examine all seed potatoes. All samples in a batch are rejected and consequently downgraded if a predetermined number of them are found to be infected. A rejected batch is a major setback because energy and money was invested in the seed potato growing process before the inspection can take place.

To reduce the number of infected potatoes, virus spread has to be minimized. The viruses found on potatoes are mainly spread by aphids [4]. These tiny insects transmit viruses through plants. Given that aphids can fly, crops can become infected rapidly. Early detection allows for timely pest management. Using pesticides to control pests is becoming increasingly difficult as pesticides become less effective due to regulations. Regulations are supposed to protect the environment and biodiversity. The available resources must therefore be used more efficiently, therefore preventive use is no longer suitable [5].

Pest control seems most effective if pesticides are used on plants where aphids are present. It is necessary to accurately detect the presence of aphids on the crops in order to research if applying pesticides at the right moment will improve the extermination process of aphids. Currently, the detection of aphids is done by placing yellow sticky plates next to the field's crops. Aphids are drawn to the yellow color, which causes them to land on the plate and stick to it. The plates will be disassembled at specific times for analysis. Analysis of the plates is difficult due to other stuck insects on the plate and the fact that humans are currently responsible for counting the aphids on the sticky plates. Due to the fact that aphids are tiny and challenging to see with the naked eye, inspection is time-consuming and accuracy depends on the inspector. Using a system based on computer vision, the preceding procedure could be automated. Such a system could also reduce the time between detecting the presence of aphids and initiating pest control, increasing the effectiveness of the procedure.

Personnel for the agriculture industry is currently difficult to find [6]. If automatic checks for the presence of aphids are possible, pest control could be applied more quickly. As a consequence, fewer personnel will be required to inspect for aphids' presence and plants infected with diseases [3]. In addition, the laborious procedure for determining the presence of aphids should be automated.

Our research investigates the classification performance of separating aphids from other possibly present insects in the field. As a result, the distinction between aphids and non-aphids is the main focus of this work. Since recent Vision Transformers (ViT)-based architectures show promising performance on classification problems [7], these are evaluated in our research. Due to the big variation in insect size and thus image size, the Multi-Scale approach appears to be a suitable fit for classifying aphids. Whereas the Convolutional Neural Networks (CNN) and ViT models are both single-scale structures, the Cross-Attention Multi-Scale Vision Transformer (Cross-ViT) [8] model is used to evaluate the Multi-Scale effect.

- *Fredrik-Otto Lautenbag is a Computing Science & Data Science student at the NHL Stenden University of Applied Sciences, E-mail: fredrik.lautenbag@student.nhlstenden.nl.*
- *Henry Maathuis is a researcher at the NHL Stenden Lectoraat in Computer Vision & Data Science, E-mail: 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.*

These aspects are covered in the following main research question: *Do Multi-Scale feature representations with Vision Transformers improve classification performance to discern aphids from non-aphids?*

- *What is the classification performance of a conventional CNN architecture?*
- *What is the classification performance of a single-scale ViT architecture? Does it outperform the single-scale CNN baseline?*
- *Can a Multi-Scale ViT architecture outperform a single-scale ViT architecture?*

2 State of the art

The first part begins with a brief description of aphid detection before diving into the state-of-the-art computer vision techniques. Similar studies, such as [9] and [10], are discussed in the second part, which explains the practical implications of these studies in detail. The third part describes the Convolutional Neural Network (CNN) architectures, which are chosen as a baseline. This architecture has primarily been used in studies such as [11] for classification and detection.

The second part describes the Convolutional Neural Network (CNN) architectures, which are chosen as a baseline. This architecture has primarily been used in studies such as [11] for classification and detection.

The Vision Transformers (ViT) architectures are described in the last part [7]. Cross-Attention Multi-Scale Vision Transformers (Cross-ViT) [8] introduce a Multi-Scale approach to enhance the performance of this architecture. Using this architecture, the Multi-Scale aspect is compared.

2.1 Insect detection

There are numerous approaches to the detection of insects. Additionally, the most interconnected research was conducted on the detection of aphids. Aphids are primarily found on crops, as this is where the problem originates. In [12], an attempt is made to directly detect aphids on wheat crops. We use a yellow sticky plate as a background in our work because potato plants do not provide a consistent capturing scene. Since detection of aphids is rare, research is also conducted on insect detection using computer vision. Since sticky yellow plates are also utilized in [9]. Due to their setup's location in a greenhouse, however, the number of insect species is limited to two. These species, namely thrips and white-flies, are easily distinguishable to the human eye.

2.2 Classification with CNN

In [11], classification performance on images with insects is evaluated by using pre-trained models such as AlexNet, ResNet-50, ResNet-101, VGG-16, and VGG-19. These models are trained, validated, and tested on three insect datasets: NBAIR [13], Xie1 [14], and Xie2 [15]. The layout of the models and hyperparameters is optimized to achieve a maximum classification accuracy of around 96 percent. This result could be found because pre-processing steps, like Canny Edge Detection, are made. In addition, augmentations, such as scaling and transposing, are made to enhance the model's performance. The pre-processing and augmentations could be valuable for our research to achieve better performance. Although performance is not fully comparable, since [11] uses multiple classes for classification. The used datasets have a different number of classes: NBAIR dataset provides 40 classes of insects, whereas Xie1 and Xie2 provide 24 classes of insects. Due to our investigation into the viability of aphid detection around potato plants, a self-collected dataset is utilized to report classification performance for this particular scene. Existing datasets were considered prior to collecting our own, but differences in conditions, such as background, orientation, and annotations, led to the collection of our own.

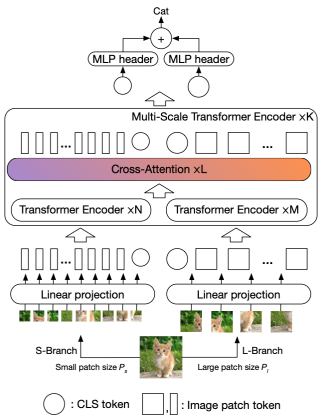


Fig. 1: An illustration of the Cross-ViT architecture. Reprinted from [8]

In [16], in addition to research on the identification of insects, research has been conducted on the classification of aphid life stages. The stages of an aphid's life cycle are as follows: the lifecycle starts as lymph, then the lymph emerges as an aphid without wings, and after a period of time, an aphid develops wings. Important to note about this study is that the aphids were collected using a special acquisition setup. A modified paper scanner captures the required high-resolution image. Aphid life stage recognition is feasible only with high resolution, similar to that of paper scanners. The research demonstrates that high-detail images are required to predict the aphid life stages or species.

2.3 Image classification with Vision Transformers

In [7], Vision Transformers (ViTs) are introduced. Transformers were already used in Natural Language Processing (NLP) [17], but now the transformer architecture can be used for image classification, where ViTs were found. Instead of using self-attention layers, which could be used together with CNNs, the complete approach is based on NLP transformers. The most important part of the transformer principle is that the input should be made of tokens. To get tokens, which are words in a NLP approach, images are tokenized. Research indicates that ViTs outperform CNNs for classification on large datasets [7].

In [8], two plain ViT architectures are incorporated into a single model, resulting in the Cross-ViT architecture. Figure 1 illustrates the architecture of Cross-ViT. Since two images could now be simultaneously inserted into the model, the Multi-Scale effect is implemented. As with the original plain ViT model, the images are also presented as patches. Due to two inputs as opposed to a single input, the patches are tokenized out of different image sizes per sub-model. This process is shown in Figure 1, the image of the cat is cut out in multiple patches. Both sets of tokens are processed by two individual Transformers. For the approach to be effective, their outputs must be combined. There are four techniques for executing this so-called Attention. Cross-Attention is selected due to its methods' superior performance. Cross-Attention permits the combination of results without requiring a huge amount of computing power. The additional classification token (CLS) generated by each individual Transformer functions as an image summary. The CLS is exchanged between Transformers to achieve the effect of cross-learning. The tokens are then fed into two multilayer perceptron networks, which output a value per class. Multi-Scale approaches are novel for ViT, but they are used in CNN, specifically in Feature Pyramid Networks (FPN) [18].

In [19], guidelines explain how to properly utilize plain ViT architectures. Due to the fundamental differences between CNN and ViT based architectures, additional explanation from their research could prevent falling into the same pitfalls addressed in their research.

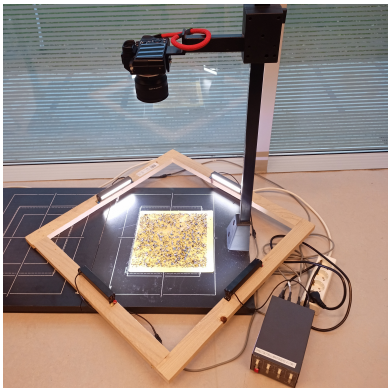


Fig. 2: Setup to capture yellow sticky plates with insects and aphids

Consequently, one of the recommendations to utilize pre-trained models has been considered. Regularization and augmentation are deemed less important when using pre-trained models. In addition to these factors, the selection of pre-training datasets is considered relevant. Guidelines for how to use Multi-Scale Cross-ViT are not given, therefore the proper application of Cross-ViT is a part of our research.

Our research differs from previous work on several aspects:

- Our dataset contains various species of insects. Since the objective is to count aphids, our classifier should classify the data into two classes: aphid or non-aphid.
- The data in our dataset has never been processed by ViTs or ViT based networks.
- We compare a conventional CNN architecture with ViT-based architectures on our insect dataset. Both evaluations act as the baseline for the Single-Scale to Multi-Scale comparison.

3 Materials and Methods

This section is divided in four parts. Section 3.1 describes how the dataset was collected and how it was structured. Section 3.2 continues by describing what models were evaluated. The following section continues by describing what pre-processing and augmentations were carried out on the data before it is inserted into the model. Section 3.4 reflects hardware-related elements of experiment execution. Finally, the utilized metrics to evaluate and compare experiment performance are discussed in Section 3.5.

3.1 Dataset

A self-collected dataset is utilized because the dataset must closely resemble the actual data collected in the field. In Section 3.1.1 the acquisition of the dataset is described, whereas Section 3.1.2 describes the structure of the dataset.

3.1.1 Dataset acquisition

The subsequent section provides a detailed description of the acquisition setup. The yellow sticky plates are gathered from various locations in the northern part of the Netherlands in order to obtain relevant variance in the data. To acquire insects over a longer period of time, the plates were placed and collected over several weeks. The plates are captured using an acquisition setup after collection. The details of the plates and the acquisition setup is described in Table 1.

In order to obtain images with sufficient detail to differentiate aphids from non-aphids, the yellow sticky plate is not captured in a single image. Instead, the plates are imaged in four quarters to provide more detail. The distance between the yellow sticky plate and camera lens was approximately 60 centimeters. Illumination is applied from each corner to be able to capture the yellow sticky plate without shadows. The described configuration is shown in Figure 2.

Table 1: Acquisition parts

Part	**Detail**
Yellow sticky plate	around 25 x 25 cm
Size of tiles	around 13 x 13 cm
Camera	Sony A7 II
Camera lens	Tamron F053S
Resolution (per tile)	4000x4000 pixels
Color depth	14 bits
Illumination	Four LED bars

After the addition of annotations, the dataset became usable. Due to the minor differences between aphids and other insects, a domain expert created the annotations. The annotations are created by drawing bounding boxes around the insects and labeling them as either aphids or non-aphids. To obtain a dataset with a sufficient quality, multiple data cleaning iterations were performed. The data cleaning procedure is described in detail in Appendix A.

Because the photos of the plates contain a large number of insects, the photos could not be used for classification directly. A new dataset is created by taking crops of the aphids and non-aphids on the sticky plates. This subset was created using object detection based on the YOLOv5 [20] network, and this process is not part of this paper. Although the performance of object detection and classification together forms the performance of the complete pipeline. Our research only focused on classification of images between aphids and non-aphids.

3.1.2 Dataset structure

The images in the dataset belong either to the aphid or non-aphid class. In Figure 3a, images of different aphids are shown. In Figure 3b, shows images of various non-aphids. The dataset counts 6508 images in total, of which 682 are aphids and 5826 are non-aphids. Hence, there is an imbalance in the number of images per class. This effect will be tempered by a pre-processing technique described in Section 3.3.

The dataset is split into three parts: training, validation, and testing. The train split contains 56 percent of the images. The validation and testing part contain respectively 24 and 20 percent of the images. The distribution of the dataset is shown in Table 2. It is important to realize that the insects and therefore cutouts vary in size. Cutout sizes range from 45x45 pixels for the smallest to 505x505 pixels for the largest image. The cutouts' aspect ratio is consistently 1:1. Due to the fact that the task of making cutouts was part of related research, the cutouts' margins could be adjusted for pre-processing and augmentations in our pipeline. The margin is multiplied by the square root of two for both the height and the width. Thus, we were able to apply a random rotation between 0 and 45 degrees and crop the center without losing or adding pixels.

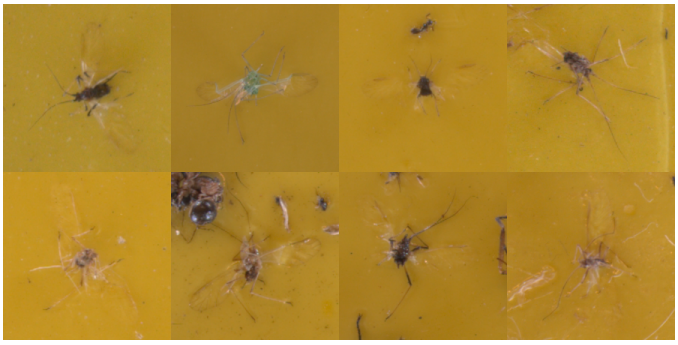
Table 2: Distribution of the dataset

Dataset split	**aphid**	**non-aphid**	**total**
train	379	3261	3640
validation	170	1404	1574
test	133	1161	1294

3.2 Classifier networks

Three models were selected to evaluate in the experiments. Pre-trained models, datasets on which they were trained, the number of parameters, and input resolution all played a role in narrowing down the search for the best possible models. Since Cross-ViT, in particular, is still in its early stages, the number of pre-trained networks available is limited. As a result, a Cross-ViT model[1]

[1]The "crossvit-small-240" model can be found in https://github.com/IBM/CrossViT/



(a) Examples of aphids stuck on the yellow sticky plate

(b) Example of non-aphids stuck on the yellow sticky plate

Fig. 3: **Two classes in the dataset: aphid and non-aphid**: (a) Aphids have injecting mouthparts in the form of a relatively long, segmented rostrum. The wings are are larger as the body (b) The non-aphid class contains a wide variety of objects. Typically, it could be other insects, but it could also be grass or other plant life, etcetera.

introduced in Section 2.3 was chosen first. The model has 26.3 million parameters and was pre-trained on the ImageNet-1K dataset, and the patch sizes are 12 and 16. Two regular ViTs[2] are used in addition to the Cross-ViT model. The patch sizes per model are 16 and 32, allowing us to include the effect of this parameter on our dataset in the experiment. Furthermore, the input format is the same, and the models are pre-trained on ImageNet-1K. The options considered for the baseline CNN were a ResNet or VGG architecture. The number of parameters, which should be similar to the models mentioned earlier, led to the selection of the ResNet architecture. One of the smallest VGG-based architectures, VGG-11, already has an excessive number of parameters around 133 million. As a result, ResNet-50, a model that has already been trained on ImageNet-1K, is selected as the baseline. Table 3 compares the number of parameters in each model. This explains why the smaller models were chosen for both the Cross-ViT and ViT architectures. First, because the number of parameters is similar to the chosen baseline. Second, the number of parameters in the models influences the time required for experiments. In addition to the number of parameters, the image

Table 3: Model parameters

model	**parameters [million]**
resnet50	23.5
vit_small_patch16_224	21.7
vit_small_patch32_224	22.5
crossvit_small_240	26.3

resolution and dataset on which pre-training was performed were factors in model selection. As shown in Table 3, the number of parameters for the selected models is comparable. The resolution of the image is set to 224 by 224 pixels. As mentioned before, all models are pre-trained on ImageNet-1K.

3.3 Data pre-processing and augmentations

To carry out various experiments with the models mentioned in Section 3.2, data pre-processing and augmentations are applied. Data handling and formulating the results from the model are covered in this section. Because the data from both classes is not balanced, balancing is performed in the pipeline. Both aphids and non-aphids are sampled such that an equal number of aphids and non-aphids are presented to the model. Data balancing is only enabled during training and validation, ensuring that metrics obtained during testing are not disrupted. Another feature of the pipeline is augmenting the data, these functions are only activated during training and validation if the selected experiment requires it. However, normalization and a combination of resizing and cropping are always performed to provide the models with images in the correct format. Due to the data balancer, as described in Section 3.3, images of aphids are repeated. Augmentation ensures images are not exactly the same for the model, so repeating images is possible to resolve the class imbalance. Besides this, the color of some yellow sticky plates differs a bit. To not make this small detail overpowering, augmentations like ColorJitter and Multiplicative Noise are included. The implemented augmentations are mentioned in Table 4 and used during training and validation. Augmentations are applied with a probability of 0.5, except the MultiplicativeNoise augmentation, which has 0.9. Since the ColorJitter and MultiplicativeNoise are configured with additional settings, these are mentioned in Table 5 and 6.

Table 4: Applied augmentations to pipeline

Augmentation	**Description**
Rotate	Rotates the image randomly (0-45 deg)
HorizontalFlip	Randomly flip over horizontal axis
VerticalFlip	Randomly flip over vertical axis
ColorJitter	Randomly vary colors
MultiplicativeNoise	Randomly apply noise

Table 5: ColorJitter Augmentation settings

Setting	**Value**
Brightness	0,1
Contrast	0,1
Saturation	0,2
Hue	0,01

Table 6: Multiplicative Noise augmentation settings

Setting	**Value**
Multiplier	0,98 - 1,02
Elementwise	TRUE

Table 7: Optimizer settings

Type	Stochastic Gradient Descent (SGD)
Momentum	0.9
Weight decay	0.0005

The optimizer's configuration is detailed in Table 7. Loss is computed utilizing Categorical Cross Entropy Loss is used with the

[2]vit-base-patch16-224 and vit-base-patch32-224 can be found at https://github.com/rwightman/pytorch-image-models/

default parameters from the Torch library [3]. A maximum of 50 epochs are performed, but an early stopping with threshold of 15 epochs is configured. Only the model with the lowest validation loss percentage is saved and tested. The settings for the learning rate scheduler are modified within the grid search. The start learning rate is used to initiate the training process. Patience represents the number of epochs without improvement after which the learning rate is reduced. As a result, once patience runs out, the learning rate drops by 0.1 factor. Furthermore, the scheduler utilizes the Torch library's [4] default settings.

3.4 Hardware resources

The specifications of the virtual machine (VM) where the experiments are executed are shown in Table 8.

Table 8: Hardware specifications

Hardware	VM in NHLStenden datacenter
CPU	Intel Xeon Gold 6338
RAM	60GB
GPU model	NVIDIA A40-16C
GPU memory	16GB
CUDA version	11.6

3.5 Evaluation metrics

The naming is set up as follows: the aphid is considered the positive class, while the non-aphid is considered the negative class. True (T) is considered a correct prediction, while false (F) is incorrect. These values are calculated into more comparable values, namely accuracy, precision, recall, and F1-score. Accuracy is the ratio between good predictions the total number of instances:

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \tag{1}$$

Precision is the ratio between the actual aphids and all predicted aphids:

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

The ratio of predicted aphids to the actual amount of aphids is known as recall:

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

Both values are represented in the F1-score. This is the harmonic mean between recall and precision.

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

The F1-score is regarded as the most important metric because it incorporates precision and recall. Due to ambiguity regarding the significance of a balance between precision and recall for our approach, the F1-score is used as the best indicator of the networks' performance. Other metrics, however, are calculated and included in the study.

4 Experiments & Results

This experiments and results section describes the experiments conducted to answer the research questions, along with the results obtained. The first set of experiments is determining the configurations for the models. In section 3.2 the baseline, ViT, and Cross-ViT model configurations are described. The specifics of this grid search are detailed in section 4.1. After determining the optimal configurations, the final experiments were conducted on the test set. These concluding experiments and their results are outlined in Section 4.2.

4.1 Grid search configurations

Due to compute limitations, a selection of grid search parameters was chosen to validate. Hyperparameter optimization has been performed by adjusting these parameters during multiple runs to find the best due to their distinctive architectures, ResNet and ViT-based models do not use the same ranges. The set of values for each parameter is indicated in Tables 11 and 12 in Appendix B. This yields a total of 108 configurations. Each configuration was performed four times to obtain an average and standard deviation. The F1-score was used to quantify the performance of each configuration. The average of these four equal configurations is derived by combining the scores from each experiment with equal parameters. The parameters of the experiment with the highest average F1-score were picked. In Table 9, the configuration with the best average F1-score is demonstrated, and the corresponding F1-score is attached as well.

4.2 Experiments with selected configurations

These experiments utilize the configurations described in Section 4.1. The experiments will be evaluated using ResNet-50, ViT with 16 and 32 patch sizes, and Cross-ViT models. In order to assess the performance under different orderings of the images, each experiment is conducted four times. At this stage, all experiments are evaluated using the same test set, as described in 3.1.2.

4.3 Results

All scores are calculated using the equations outlined in Section 3.5. Table 9 lists the configurations with the highest F1-score for the models that have been verified. These outcomes are the result of the experiments detailed in Section 4.1. The values of the discovered hyperparameters are also summarized. In addition to selecting the configuration with the highest F1-score, models that are close to the best configuration in terms of performance are also analyzed. In order to conduct this analysis, the configurations with the highest performance are summarized in tables. These tables are attached in Appendix C. Resnet-50, ViT16, and Cross-ViT are evaluated on the test set. We chose ViT16 out of both plain ViT models since the model with the smaller path size achieved higher performance. Table 10 displays the final results of the experiments discussed in Section 4.2. In addition to these results, Figure 4 describes the misclassified images found in all four Cross-ViT experiments.

5 Discussion, Conclusion & Future Work

This section discusses the outcomes of the experiments, draws conclusions regarding their relevance to the research topics, and outlines concepts for the next steps of the project.

5.1 Discussion

The first part of the discussion compares the CNN-based architectures with the ViT-based architectures. The second part details the selected configurations during the grid search and how they relate to the recommendations in the state-of-the-art. The third part describes the context of the incorrect classifications. The last part discusses the impact of data cleaning.

The state-of-the-art indicates that ViT-based architectures should perform better than CNN-based architectures. Comparing the F1-scores of the ResNet-50 and ViT16 models proves this. In addition to its higher F1-score, the standard deviation of this ViT16 model's F1-score is lower. Since the weights of these models were both initialized from pre-trained models, it appears that the training process of the ViT16 model is more consistent than that of the ResNet-50 model. However, additional experiments are required to confirm this. The robustness of training also applies to the ViT32 model, but its performance is not superior to that of the ViT16 model. Since the difference between these models is patch size, we can say that the ViT16 model, which divides images into smaller patches, is

[3] Configuration at: https://pytorch.org/docs/stable/generated/torch.nn.CrossEntropyLoss.html

[4] Configuration at: https://pytorch.org/docs/stable/generated/torch.optim.lr_scheduler.ReduceLROnPlateau.html

(a) Predicted as an aphid (b) Predicted as an aphid (c) Predicted as an aphid (d) Predicted as an aphid

Fig. 4: **Miss classified images**: (a) Multiple insects (parts) disturb the classifier. (b and c) both are clearly not aphids, so the prediction is constantly incorrect. (d) The domain expert is not sure about the class to which this image belongs.

Table 9: Configurations with best performance on validation set per model

Model	Start LR	Patience	Batch size	Augmentations	F1-score
ResNet-50	0.05	5	16	Enabled	84.95% ±2.09%
ViT16	0.0001	20	8	Disabled	86.51% ±2.15%
ViT32	0.0001	10	8	Disabled	85.33% ±0.77%
Cross-ViT 12/16	0.0005	10	8	Enabled	85.33% ±1.37%

Table 10: Performance on test set per model

model	accuracy	precision	recall	F1-score
ResNet-50	95.11% ±1.69%	90.41% ±2.56%	71.12% ±8.26%	79.43% ±5.70%
ViT16	95.81% ±0.61%	88.72% ±1.84%	75.29% ±4.45%	81.37% ±2.04%
Cross-ViT 12/16	96.54% ±0.28%	94.36% ±2.34%	77.18% ±2.20%	**84.88% ±1.06%**

more sensitive to subtle differences in the images. In this study, we found that small patch sizes work best for self-attention in images with small objects, like the insects in our dataset. Combining two patch sizes, as in the Cross-ViT model, increases the efficiency of self-attention. Cross-ViT is superior to ResNet-50 and plain ViTs as shown by the higher F1-score. In addition, the ViT's F1-score has the lowest standard deviation, indicating that this model is also the most robust. Despite the fact that the minor difference in validation performance between ViT16 and Cross-ViT indicates the opposite, the standard deviation should be included in the comparison to demonstrate that the difference is minimal. There are discrepancies between the F1-score distributions of the ResNet-50 and plain ViT models, as ResNet-50 outperforms ViT16 in terms of precision with standard thresholds. The opposite is applicable to recall. However, the Cross-ViT model is more powerful, so selection based on these metrics is unnecessary. Cross-ViT outperforms the other models in terms of all average scores and standard deviations.

For pre-trained plain ViT-based models, augmentations do not improve performance. The state-of-the-art has previously shown this [19]. However, we discovered that augmentations improve the performance if pre-trained Cross-ViT models are used. Table 22 suggests that about 1 percent of the increase in the F1-score is attributable to augmentations.

Table 9 indicates that ViT-based models outperform the ResNet-50 model on the validation split, unless the performance of the models in the grid search cannot be used to draw conclusions. As described in Section 4.3, the appendices contain additional results. These results demonstrate that the configurations with particular hyperparameters obtained through these grid searches are optimal. This additional validation is performed on the batch size and augmentations. If a chosen hyperparameter value dominates the best-performing experiments, it can be assumed to be the correct parameter. Furthermore, the selection was based on averages, so that an unintentional outlier would have less influence. Since the best performing configurations vary based on the start learning rate, the grid search result for this parameter is validated in a different manner. The selected start learning rate should not be on a grid search limit and within the selected limits. Since all initial learning rates were within the specified limit, the identified values are considered as close as possible. Since the learning rate and patience are directly related, only the learning rate is selected for verification.

As demonstrated in Figure 4, the model consistently misclassifies images. To describe the context, four of them are displayed. As stated previously, images should only contain one insect. Due to the density of insects on the yellow sticky plates, that is impossible. The use of yellow sticky plates will result in misclassified images due to the presence of multiple insects or insect parts. Particularly Figure 4a depicts numerous insect parts, demonstrating the difficulty of classifying such images. Misclassifications are only reported for the test split of the dataset, but it is reasonable to assume that similar cases are also present in the other splits. Consequently, we can say that our dataset still contains imperfect images, which impacts the classification performance.

Since the results are based solely on the most recent set of cleaned data, performance claims due to data cleaning are impossible. Since it is evident that the quality of the annotations has improved throughout the cleaning iterations, we anticipate that there is still room for improvement. Due to the time constraints of our research, two iterations were conducted. In addition, the primary purpose of our research is to compare multiple models, so the data quality achieved after two cleaning iterations was deemed adequate.

5.2 Conclusion

This study proposed a comparison of the CNN baseline, the standard ViT model, and the Cross-ViT model. Prior to discussing the results, the research questions were answered.

What is the classification performance of a CNN baseline?. ResNet-50 is chosen as CNN baseline. The mean F1-score is 79.43 percent, and the standard deviation is 5.7 percent. This indicates that 71.12 percent of the aphids were discovered. And 90.41 percent of

the predicted aphids are actually aphids.

In order to answer the question, *What is the classification performance of a single-scale ViT architecture?* The classification performance of the single-scale ViT architecture is represented by the ViT16 since its smaller patch size provides higher sensibility to small objects.

Does it outperform the single-scale CNN baseline? The ViT16 outperforms the CNN baseline with an F1-score that is 1.94 percent higher and a standard deviation that is 3.66 percent lower.

Can a Multi-Scale ViT architecture outperform a single-scale ViT architecture?, to answer this question, the performance of the Cross-ViT model is evaluated. The achieved F1-score is 84.88 percent with a standard deviation of 1.01 percent. Cross-ViT, being a Multi-Scale ViT architecture, outperforms ViT16, a single-scale ViT architecture, by 3.51 percent on the F1-score. In addition to the substantial difference in the F1-score, the Cross-ViT architecture has a 0.98 percent lower standard deviation. Cross-ViT performance is consequently more consistent than ViT16 performance.

Since the experiments were conducted on our insect database, we can conclude those Multi-Scale representations of features with Vision Transformers significantly improve classification for this problem. Due to the limited number of data cleaning iterations, the performance obtained on this particular dataset could be improved further. The classification component is used within a larger project in which the goal is to first localize insects and afterward classify them as aphid or non-aphid. To ensure that the performance is adequate for aphid detection, it is necessary to evaluate the overall performance.

5.3 Future Work

The models used in this research were obtained from the TIMM library [5]. Since the suitable models for our research were limited, we anticipate improvements in the classification performance if there are no limitations in terms of model comparability. Due to the active community surrounding ViTs, it is anticipated that additional pre-trained models will be available in the near future. In addition to using models with other datasets as a source for pre-training, models with more parameters can be used to search for performance improvements. Due to the number of model parameters that were attempted to be matched in our research, it is likely that quality was compromised. Implementing the findings of this study and enhancing the pipeline includes selecting the optimal Cross-ViT model, which will most likely improve classification performance. As stated previously, there is room for improvement in the incorrect image labels. If the model can be trained on better data, the performance of classification will likely improve. To further reduce the number of incorrect labels, additional steps can be taken, such as executing more data cleaning cycles. However, we have discovered that clean-up in collaboration with a domain expert is labour-intensive. The optimal solution appears to be an environment in which only aphids are captured.

Our research only verifies the Multi-Scale effect with ViTs. Evaluating Multi-Scale based on CNN makes the comparison more complete. A model like the Feature Pyramid Networks, which were mentioned earlier, could provide the required metrics for comparison.

Since our research indicates inconsistency between classification performance on the train and test splits of the dataset, it would be preferable to report performance using the entire dataset. Cross-Validation, which utilizes all of the data across multiple iterations, appears to be a viable solution.

An object detector generated the selection of the dataset used in our research. The overall performance of the pipeline is dependent upon both the object detector and this classification system. We discovered that the used object detector can further be improved, as shown by the misclassifications. Consequently, the input to our classification algorithm is inadequate. An alternative to object detection was investigated to provide quality input to our network. Instead of doing object detection on a plate where lots of insects are stuck. The insects are captured in flight, and the advantages and other details are described in Appendix D under the heading *Line scan setup*.

[5] https://timm.fast.ai/

Acknowledgements

- This project is financially supported by SNN and performed within the POP3+ Fryslân project Innovatie luizendetectie

References

[1] Gerben Jukema and Pascal Ramaekers. *De Nederlandse agrarische sector in internationaal verband - editie 2022*. Number 2022-001 in Rapport / Wageningen Economic Research. Wageningen Economic Research, 2022. Project number: BO-43-115-007. - Project code 2282500352.

[2] J.J. van Hoogen. Pootgoedexport in tonnen oogst 2021, Jul 2022.

[3] J.A.L.M. Kamp, P.M. Blok, G. Polder, J.M. van der Wolf, and Henk Jalink. *Smart Ziekzoeker 2015: Detectie van virus- en bacteriezieke pootaardappelen met behulp van vision- en sensortechnologie*, volume 703 of *PPO/PRI rapport*. Praktijkonderzoek Plant and Omgeving (Applied Plant Research) Business Unit AGV, 2016. Dit project is mogelijk gemaakt door: - Topsector Agro and Food – onderdeel programma "op naar precisielandbouw 2.0"; - BO-Akkerbouw / LTO Nederland - Kverneland - Agrico - HZPC - NAK - Phenovation BV.

[4] Florence Olubayo, A Kibaru, Huria John, Rose Njeru, and Muo Kasina. Management of aphids and their vectored diseases on seed potatoes in kenya using synthetic insecticides, mineral oil and plant extracts. j innov dev strateg 4:1-5. 01 2010.

[5] Leo van Overbeek, Kirsten Leiss, Johanna Bac-Molenaar, Marie Duhamel, and Sanae Mouden. *Literatuuroverzicht Plantweerbaarheid: Om inzicht te verkrijgen hoe plantweerbaarheid tot stand komt en de rol van het metaboloom en microbioom daarin*. Number WPR-1043 in Report / Stichting Wageningen Research, Wageningen Plant Research, Business unit Glastuinbouw. Stichting Wageningen Research, Wageningen Plant Research, Business unit Glastuinbouw, 2022.

[6] Nienke Oomes. Dutch labour market shortages and potential labour supply from africa and the middle east: Is there a match?, Dec 2020.

[7] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *CoRR*, abs/2010.11929, 2020.

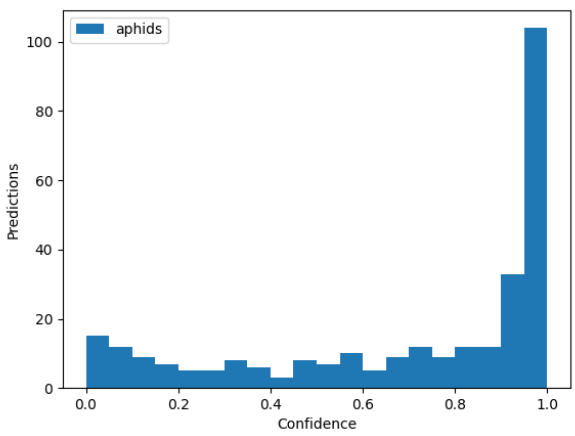
[8] Chun-Fu Chen, Quanfu Fan, and Rameswar Panda. Crossvit: Cross-attention multi-scale vision transformer for image classification. *CoRR*, abs/2103.14899, 2021.

[9] Wen-yong 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.

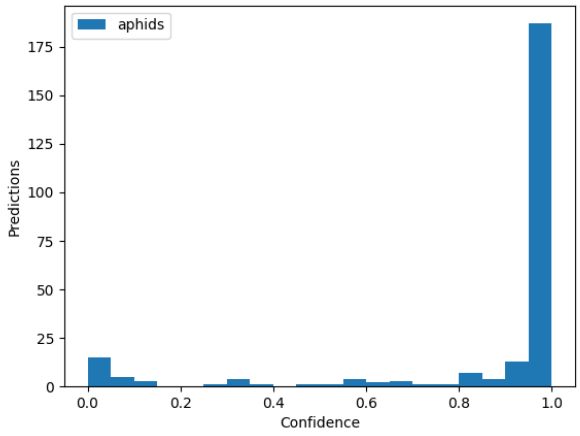
[10] Mu Qiao, Jaehong Lim, Chang Woo Ji, Bu-Keun Chung, Hwang-Yong Kim, Ki-Baik Uhm, Cheol Myung, Jongman Cho, and Tae-Soo Chon. Density estimation of bemisia tabaci (hemiptera: Aleyrodidae) in a greenhouse using sticky traps in conjunction with an image processing system. *Journal of Asia-pacific Entomology - J ASIA-PAC ENTOMOL*, 11:25–29, 03 2008.

[11] K. Thenmozhi and U. Srinivasulu Reddy. Crop pest classification based on deep convolutional neural network and transfer learning. *Computers and Electronics in Agriculture*, 164:104906, 2019.

[12] Tao Liu, Wen Chen, Wei Wu, Chengming Sun, Wenshan Guo, and Xinkai Zhu. Detection of aphids in wheat fields using a computer vision technique. *Biosystems Engineering*, 141:82–93, 01 2016.



(a) First data cleaning iteration



(b) Second data cleaning iteration

Fig. 5: **Data cleaning**: Both histograms shows the normalized output probabilities into the aphid class. (a) First iteration (b) Second iteration

[13] Bing Liu, Luyang Liu, Ran Zhuo, Weidong Chen, Rui Duan, and Guishen Wang. A dataset for forestry pest identification. *Frontiers in Plant Science*, 13:857104, 07 2022.

[14] Chengjun Xie, Jie Zhang, Rui Li, Jinyan Li, Peilin Hong, Junfeng Xia, and Peng Chen. Automatic classification for field crop insects via multiple-task sparse representation and multiple-kernel learning. *Computers and Electronics in Agriculture*, 119:123–132, 2015.

[15] Chengjun Xie, Rujing Wang, Jie Zhang, Peng Chen, Wei Dong, Rui Li, Tianjiao Chen, and Hongbo Chen. Multi-level learning features for automatic classification of field crop pests. *Computers and Electronics in Agriculture*, 152:233–241, 2018.

[16] Elison Alfeu Lins, João Pedro Mazuco Rodriguez, Sandy Ismael Scoloski, Juliana Pivato, Marília Balotin Lima, José Maurício Cunha Fernandes, Paulo Roberto Valle da Silva Pereira, Douglas Lau, and Rafael Rieder. A method for counting and classifying aphids using computer vision. *Computers and Electronics in Agriculture*, 169:105200, 2020.

[17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. 06 2017.

[18] Tsung-Yi Lin, Piotr Dollár, Ross B. Girshick, Kaiming He, Bharath Hariharan, and Serge J. Belongie. Feature pyramid networks for object detection. *CoRR*, abs/1612.03144, 2016.

[19] Andreas Steiner, Alexander Kolesnikov, Xiaohua Zhai, Ross Wightman, Jakob Uszkoreit, and Lucas Beyer. How to train your vit? data, augmentation, and regularization in vision transformers. 2021.

[20] Hyun-Ki Jung and Gi-Sang Choi. Improved yolov5: Efficient object detection using drone images under various conditions. *Applied Sciences*, 12(14), 2022.

A Data cleaning

Data cleaning is done to reduce the number of incorrectly annotated images. Incorrectly annotated data causes problems during the training process, and the models' classification performance suffers. Due to the large number of images in our dataset, letting the domain experts review all the images was not feasible. Therefore, the normalized output probability scores of a baseline model are used to select images for extra review. Figures 5a and 5b shows the normalized output probability in the aphid class. Images with a confidence rating below a certain threshold should be sent for extra review. Figure 5 demonstrates the normalized output probability is less distributed in the results of the second data cleaning iteration.

B Grid searches

The following hyperparameters are tested on the ResNet-50 model on the validation split of our dataset during the grid search. The examined values for the parameters are described in Table 11.

Table 11: Grid search CNN-based models

Parameter	Values
Start LR	{0.0001, 0.0005, 0.001, 0.005, 0.01, 0.05}
Patience	{5, 10, 20}
Batch size	{8, 16, 32}
Augmentations	{Enabled, Disabled}
Network	{ResNet-50}

To determine the optimal configuration for ViT and Cross-ViT, a grid search similar to the one mentioned in Table 11 is conducted. The considered values for the parameters are depicted in Table 12.

Table 12: Grid search ViT-based models

Parameter	Values
Start LR	{1e-05, 5e-05, 0.0001, 0.0005, 0.001, 0.005}
Patience	{5, 10, 20}
Batch size	{8, 16, 32}
Augmentations	{Enabled, Disabled}
Network	{ViT 16, ViT 32, Cross-ViT 12-16}

C TOP 10 CONFIGURATIONS PER MODEL

Tables 13, 16, 19, and 22 presents the ten best configurations based on F1-score per model. The following tables (14, 15, 17, 18, 20, 21, 23, 24) depicts the distribution of batch size and augmentation parameters.

Table 13: Top 10 configurations for ResNet-50 model sorted on F1-score

Start LR	Patience	Batch s.	Aug.	F1-score
0.05	5	16	Enabled	84.95% ±2.09%
0.05	10	32	Enabled	84.81% ±3.33%
0.01	5	8	Enabled	84.39% ±1.93%
0.005	20	16	Enabled	83.94% ±3.42%
0.005	5	16	Enabled	83.82% ±1.04%
0.005	20	32	Enabled	83.76% ±1.93%
0.005	5	8	Enabled	83.74% ±1.41%
0.001	10	8	Disabled	83.73% ±4.83%
0.01	5	32	Disabled	83.67% ±2.39%
0.01	20	16	Enabled	83.56% ±2.11%

Table 14: Distribution of the batch size within the top 10 ResNet-50 model

Batch size	Amount within top 10
8	3
16	**4**
32	3

Table 15: Distribution of the augmentations within the top 10 ResNet-50 model

Augmentations	Amount within top 10
Enabled	**8**
Disabled	2

Table 16: Top 10 configurations for ViT16 model sorted on F1-score

Start LR	Patience	Batch s.	Aug.	F1-score
0.0001	20	8	Disabled	86.51% ±2.15%
0.0001	10	16	Enabled	86.12% ±0.79%
1e-05	20	8	Enabled	85.63% ±2.02%
0.0001	5	8	Disabled	85.46% ±4.06%
0.0005	10	32	Disabled	85.30% ±1.05%
0.0001	5	8	Enabled	85.15% ±0.48%
0.0001	10	16	Disabled	85.11% ±1.71%
5e-05	10	8	Disabled	85.09% ±1.50%
0.0005	5	32	Disabled	85.04% ±3.90%
0.0005	20	32	Enabled	84.75% ±1.37%

Table 17: Distribution of the batch size within the top 10 ViT16 model

Batch size	Amount within top 10
8	**5**
16	2
32	3

Table 18: Distribution of the augmentations within the top 10 ViT16 model

Augmentations	Amount within top 10
Enabled	4
Disabled	**6**

Table 19: Top 10 configurations for ViT32 model sorted on F1-score

Start LR	Patience	Batch s.	Aug.	F1-score
0.0001	10	8	Disabled	85.33%±0.77%
0.001	10	32	Disabled	85.00%±1.07%
0.0005	5	32	Disabled	84.55%±2.22%
0.0001	5	8	Disabled	84.49%±1.54%
5e-05	5	8	Enabled	84.21%±1.68%
0.0005	10	32	Disabled	84.18%±1.46%
0.0005	20	32	Enabled	83.66%±1.25%
0.001	5	16	Disabled	83.54%±1.68%
0.0005	5	16	Disabled	83.53%±0.63%
1e-05	10	8	Enabled	83.42%±1.76%

Table 20: Distribution of the batch size within the top 10 ViT32 model

Batch size	Amount within top 10
8	4
16	2
32	4

Table 21: Distribution of the augmentations within the top 10 ViT32 model

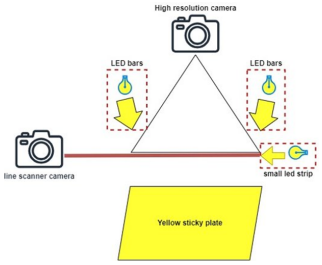
Augmentations	Amount within top 10
Enabled	3
Disabled	**7**

Table 22: Top 10 configurations for Cross-ViT model sorted on F1-score

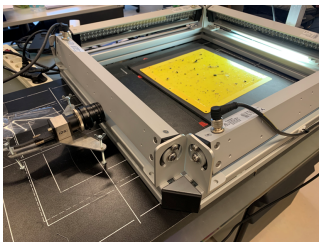
Start LR	Patience	Batch s.	Aug.	F1-score
0.0005	10	8	Enabled	85.33% ±1.37%
0.0001	10	8	Enabled	85.17% ±1.95%
0.001	20	32	Enabled	85.13% ±3.17%
0.0005	10	16	Enabled	85.10% ±2.66%
0.001	10	32	Enabled	84.90% ±2.44%
0.0005	10	8	Disabled	84.47% ±1.59%
0.0005	10	32	Enabled	84.02% ±5.23%
5e-05	20	16	Enabled	83.87% ±2.89%
0.0005	5	32	Enabled	83.82% ±2.62%
0.0005	20	8	Disabled	83.48% ±3.98%

Table 23: Best batch size configuration for the Cross-ViT model

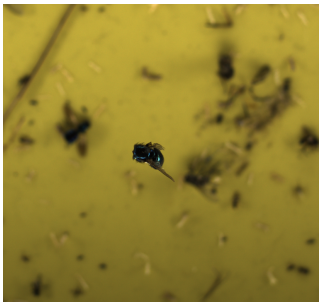
Batch size	Amount within top 10
8	4
16	2
32	4



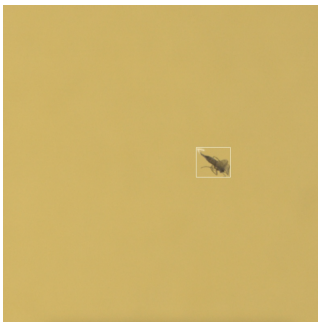
(a) Overview of the line scan camera approach



(b) Photo of the experimental line scan setup



(c) Image of falling insect



(d) Image of falling fruit fly annotated with bounding box

Fig. 6: **Line scan camera setup**: (a) The line scan camera scans the red line towards the small led strip. If something passes; the high resolution camera captures an image with the insect in focus. (b) The camera is on the left, pointing towards the light source (right between LED bar and aluminium profile). The LED bars around the yellow sticky plate are enabled when the high resolution camera captures the scene. (c) Image captured by high resolution camera, the falling insect is in focus and others are blurry (d) Same as previous but a bounding box around the fruit fly has generated with edge detection.

Table 24: Best augmentation configuration for the Cross-ViT model

Augmentations	**Amount within top 10**
Enabled	2
Disabled	**8**

D LINE SCAN SETUP

Due to the fact that there is room for improvement in the performance of the object detector, other techniques to obtain a cutout with an insect to insert in our classifier are being researched. The line scan setup shows promising results for this task. As shown in Figure 6a, the setup is similar to the one described in Section 3.1.1. The high-resolution camera is still capturing the yellow sticky plates (YSP) while illuminated by the four LED bars around the YSP. The LED bars are replaced with ones with flash capability, due to the timing should be perfect to capture the insect while it is falling. This is accomplished by setting the field of focus of the camera around five centimeters above the yellow sticky plate. The lifted field of focus allows us to capture the insects in flight and ensures that the stuck insects on the yellow sticky plate are blurred. Therefore the insect can easily be located with edge detection. The high-resolution camera could constantly capture the scene, but this has several disadvantages. Firstly the acquisition setup requires exceptional hardware, the camera should capture the scene with a very high frame rate. Because the resolution is also large, the amount of data to process and store will be enormous. Secondly, the amount of data to process after the data is captured and stored is immense and requires huge computing power. To tackle this problem, a high resolution camera should only capture the scene if there is actually an insect in the field of focus. A trigger signal should be given without significant delay to the high resolution camera. The line scan setup provides this trigger signal. In Figure 6a the red line displays the trigger field of the line scan camera. The red line lays in the field of focus of the high resolution camera. To be sure the line scan camera triggers only on objects in the field of focus, a small led strip is used as light source, this makes the setup robust. In Figure 6b the experimental setup is shown. The line scan camera is a IDS UI-3060CP-M-GL together with a Kowa HR978NCN-3H lens. This camera is configured to capture only one row of pixels to be able to provide the trigger signal as quickly as possible. A captured image with this setup is shown in Figure 6c. In Figure 6d, the falling insect is surrounded by a bounding box. The coordinates of the bounding boxes are acquired with edge detection, which makes this system fast and requires a little computing resource.