Tomas Bata University in Zlín
Faculty of Applied Informatics

Doctoral Thesis

Deep Learning Methods Applied in Computer Vision

Využití metod hlubokého učení v počítačovém vidění

Author:	**Ing. Alžběta Turečková**
Branch of study:	P3902 / Engineering Informatics
Supervisor:	doc. Ing. Zuzana Komínková Oplatková, Ph.D.

Zlín, 2023

To my daughter, Lýdie:
This is what I have been doing all the while you were asking. I love you.

First and foremost, I would like to sincerely thank my husband, Tomas,
who made this journey possible from beginning to end.
I am incredibly grateful to my father, sister, and our entire family;
I consider myself fortunate to have you all in my life.
My deepest appreciation goes to my supervisor, Zuzka, for her guidance on my
winding path into the world of research and for her endless patience.
Finally, I would like to thank my colleagues from A.I.Lab, namely Michal,
Adam, Tomáš, Roman, and Anežka, for their steadfast support and inspiration.

"A ship in port is save, but it's not what ships are made for.
Sail out to see and do new things."
Grace Hopper

SUMMARY

This Doctoral Thesis investigates the significant role of data handling in the practical application of deep learning techniques for object detection in high-resolution images. The study examines the impact of attention mechanisms and introduces novel data processing methodologies, namely Artificial Size Slicing Aided Fine Tuning (ASSAFT) and Artificial Size Slicing Aided Hyper Inference (ASSAHI). Despite the potential of attention mechanisms observed in medical imaging, the practical application of similar principles in the custom-made Tomato360 dataset does not prove to be beneficial. On the other hand, a substantial improvement in object detection performance in the Tomato360 dataset was achieved through the newly proposed ASSAFT and ASSAHI techniques. The research underlines the challenges of deploying deep learning techniques in real-world scenarios; concretely, the final proposed solution is utilized and evaluated for estimating crop yields in tomato greenhouses.

ABSTRAKT

Tato disertační práce zkoumá významnou roli zpracování dat při praktickém použití technik hlubokého učení pro detekci objektů v obrazech s vysokým rozlišením. Práce zkoumá dopad mechanismů pozornosti a představuje nové metody zpracování dat, konkrétně Artificial Size Slicing Aided Fine Tuning (ASSAFT) a Artificial Size Slicing Aided Hyper Inference (ASSAHI). Přes úspěšné použití mechanismů pozornosti při zpracování medicínských dat, praktické uplatnění podobných principů v nově vytvořeném datasetu Tomato360 se neukázalo prospěšné. Na druhou stranu, významné zlepšení kvality detekce objektů v datasetu Tomato360 bylo dosaženo prostřednictvím nově navržených technik ASSAFT a ASSAHI. Práce dokumentuje výzvy spojené s nasazením technik hlubokého učení v reálných aplikacích; konkrétně je finální navržené řešení využito pro odhad sklizně rajčat ve skleníku.

TABLE OF CONTENTS

LIST OF FIGURES **8**

LIST OF TABLES **9**

LIST OF ABBREVIATIONS **10**

1 INTRODUCTION **11**

1.1 BASIC COMPUTER VISION TASKS 12

1.2 APPLYING COMPUTER VISION TO AGRICULTURE 14

1.3 SMALL OBJECT DETECTION IN HIGH-RESOLUTION IMAGES 16

2 AIMS OF DOCTORAL THESIS **18**

3 LITERATURE REVIEW **19**

3.1 DEEP CONVOLUTIONAL NEURAL NETWORKS 20

3.1.1 Artificial Neural Network Layers 21

3.1.2 GPU Acceleration 22

3.2 OBJECT DETECTION 23

3.3 SEMANTIC SEGMENTATION 25

3.4 ATTENTION MECHANISM IN CNN 28

3.4.1 Attention Gates 28

3.4.2 Spatial Attention Mechanisms in Deep Networks 31

3.5 APPLICATIONS OF DEEP LEARNING IN AGRICULTURE 32

3.6 PROCESSING HIGH-RESOLUTION IMAGES 35

3.6.1 Cropping Image Patches 35

3.6.2 Multiple Detection Suppression 37

4 METHODOLOGY **38**

4.1 DATASET COLLECTION AND PREPROCESSING 39

4.1.1 Dataset Normalization 39

4.1.2 Image Augmentation 40

4.1.3 Slicing Aided Fine-tuning 41

4.1.4 Artificial Size Slicing Aided Fine-tuning 42

4.1.5 Dataset Splitting 43

4.2 MODEL TRAINING 44
4.2.1 Loss Function 44
4.2.2 Optimizers 45
4.2.3 Learning Rate 45
4.3 MODEL INFERENCE 46
4.3.1 Slicing Aided Hyper Inference 46
4.3.2 Artificial Size Slicing Aided Hyper Inference 46
4.3.3 Image stitching 49
4.4 VALIDATION METRICS 52
4.4.1 Metrics operating with the confusion matrix values 52
4.4.2 Object detection evaluation according to COCO challenge 54
4.4.3 Evaluation metrics for segmentation 55
5 RESULTS 57
5.1 CREATION OF THE TOMATO360 DATASET 57
5.1.1 Field Data Collection 59
5.1.2 Video converting 60
5.1.3 Data Annotation 62
5.1.4 Dataset statistics 63
5.1.5 Dataset challenges 63
5.2 ATTENTION GATES FOR TUMOR SEGMENTATION 65
5.2.1 Methodology 66
5.2.2 Experimental Evaluation and Discussion 67
5.2.3 Visualization of the Attentional Maps 70
5.3 SPATIAL ATTENTION FOR TOMATO DETECTION 72
5.3.1 Methodology 72
5.3.2 Experimental Evaluation and Discussion 74
5.4 ASSAFT AND ASSAHI IN TOMATO DETECTION 76
5.4.1 Methodology 77
5.4.2 Experimental Evaluation and Discussion 79
5.5 PRACTICAL APPLICATIONS IN TOMATO GREENHOUSE 83
5.5.1 Whitefly Detection 84
5.5.2 Tomato Detection and Counting 86

5.5.3 Estimating Yield of Tomato Crops 88

6 DISCUSSION **90**

6.1 Fulfillment of the Doctoral Thesis Aims 91

7 IMPACT OF WORK ON SCIENCE AND PRACTICE **94**

7.1 Limitations and Future Directions 96

8 CONCLUSION **97**

REFERENCES **99**

PUBLICATIONS OF THE AUTHOR **110**

CURRICULUM VITAE **113**

LIST OF FIGURES

1.1 Illustration of the fundamental computer vision tasks 13
1.2 Example images representing the challenging small object detection 17
3.1 The schema of encoder-decoder network structure. 26
3.2 A block diagram of additive attention gate (AG) [62] 29
3.3 Illustration of different attention terms [100] 31
3.4 Attended residual block [100] 32
4.1 Five-fold cross-validation splitting method. 44
4.2 Visualization of Artificial Size Slicing Aided Hyper Inference (ASSAHI) procedure proposed in this thesis. 48
5.1 A frame from a 360-degree camera before any preprocessing. . 60
5.2 Projection of a 360-degree camera frame into two planes, only the vertical part is utilized in the final image reconstruction. . 61
5.3 Example images documenting difficulties of Tomato360 dataset 64
5.4 A CNN segmentation model with attention gates and deep supervision. 67
5.5 Examples of attention maps (AM) obtained from attention gates in the three topmost levels 70
5.6 Visualization of attention maps (AM). 71
5.7 Example of whitefly detection on yellow sticky tag 85
5.8 Precision, recall, and F1-score comparison for different percentiles of tomato vertical position. 88

LIST OF TABLES

5.1 Distribution of tomato ripeness categories in the Tomato360 dataset. 62
5.2 Comparison of the proposed VNet-AG-DSV to the state-of-the-art network with similar parameters presented by [42]. 68
5.3 Performance comparison. 69
5.4 Architecture comparison (test on slices) - **tomato one class** . 75
5.5 FLOP in trillions (tera, 10^{12}) and number of parameters in millions (mega 10^{6}) for each of tested architecture, tested on the input image of size 1024x1024pixels. 76
5.6 Foreground segmentation mean accuracy (mACC) and mean Intersection over union (mIOU results) for semantic segmentation model trained to obtain object groups positions needed for test image Artificial Size Slicing Aided Hyper Inference (ASSAHI). 80
5.7 Evaluation of proposed ASSAFT and ASSAHI methodologies on a Tomato360 dataset. 81
5.8 Comparison of performance metrics between Faster-RCNN DCN and Tood DCN model, using the ASSAHI method with FCN-Unet or ground truth mask (gtmask) as segmentation input. For each model, results are shown for two different training schemas (12 and 24 epochs). 82
5.9 Comparison of crop estimates made by an agronomist and by the proposed model. 89

LIST OF ABBREVIATIONS

AI	Artificial Intelligence
ASSAFT	Artificial Size Slicing Aided Fine Tuning
ASSAHI	Artificial Size Slicing Aided Hyper Inference
bbox	bounding box
COCO	Common Objects in Context, a large-scale dataset.
CV	Computer Vision
FLOP	Floating-Point Operation
FLOPS	Floating-Point Operation Per Second
GFLOPS	Giga Floating-Point Operations Per Second
IOU	Intersection Over Union
IOS	Intersection Over Smaller Area
DL	Deep Learning
GREEDYNMM	Greedy Non-maximum merging
M	Mega, Millions, 10^6
NMM	Non-maximum merging
NMS	Greedy non-maximum suppression
SAFT	Slicing Fine Aided Tuning
SAHI	Slicing Hyper Aided Inference
T	Tera, trillions, 10^12
YST	Yellow Sticky Tag

1 Introduction

There is an enormous amount of image/video data created each moment. The internet, social networks, security cameras, and the data from medical scanners are just a few examples of image data resources. Although the processing of visual information is natural for us humans, and we can do it with ease, the amount makes the manual analysis of all data impossible. Understanding information encoded in images is critical and may be helpful in many social areas. For example, automatic emergency braking is already standard equipment in new cars; a machine can preventively screen a patient's medical images, letting a doctor check only suspicious samples; or a security camera can start an alarm when an unattended child falls into a swimming pool.

Computer vision is an interdisciplinary field concerned with processing image data to gain a high-level understanding of a scene. In other words, it tries to mimic and automate the task the human visual system can do. Computer vision deals with the automatic extraction, analysis, and understanding of useful information from a single image or a sequence of images. To achieve this goal, it uses algorithms often involving artificial intelligence. The image data are available in many forms, such as video sequences, views from multiple cameras, or multi-dimensional data from a medical scanner [23, 25].

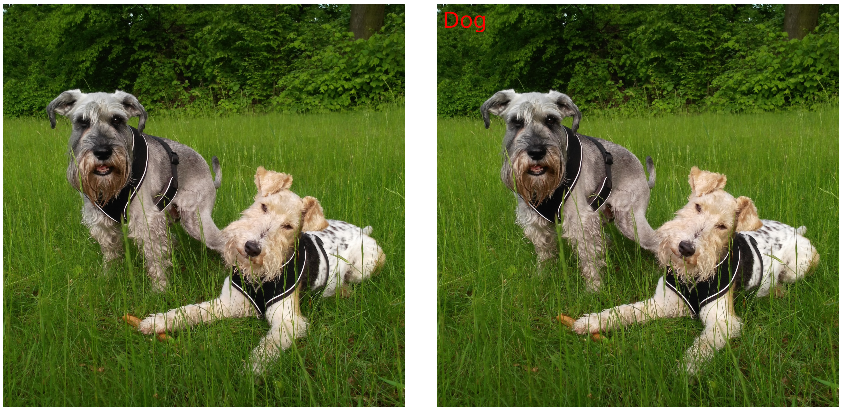
This introduction section continues by discussing fundamental computer vision tasks from a computer science perspective. Building upon this foundation, the text delves into the application of computer vision in agriculture, a domain with significant potential for technological advancements. One critical challenge that remains unsolved is small object detection in high-resolution images. Small objects often exhibit low contrast, partial occlusion, and complex spatial arrangements, making their accurate detection a formidable task. Motivated by the desire to address this challenge, this thesis focuses on developing novel approaches to tackle small object detection in high-resolution images, intending to enhance precision and efficiency in agricultural applications. Outlining these motivations establishes the rationale for the subsequent section that details the specific aims of this dissertation, which include the creation of a custom dataset, incorporation of attention mechanisms within different CNN architectures, and the development of a tailored processing pipeline for handling high-resolution images and small object detection.

1.1 Basic Computer Vision Tasks

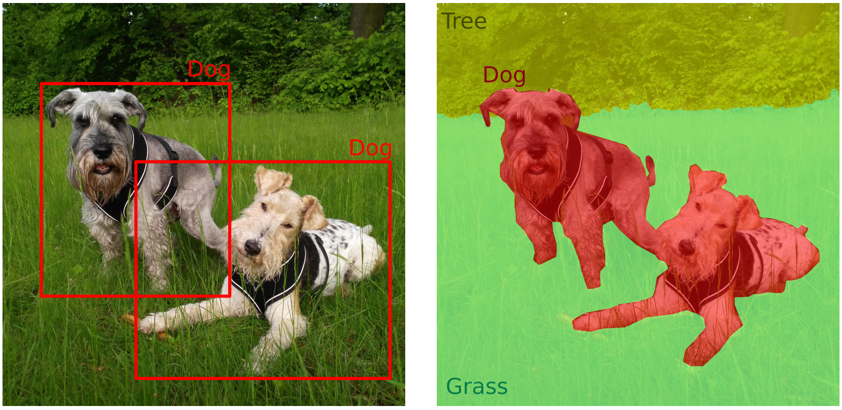
Computer vision seeks to interpret images in much the same way humans do. To approach such a complex problem, it can be divided into five subtasks with increasing complexity: classification, object detection, semantic segmentation, instance segmentation, and panoptic segmentation. For illustration, please see Fig. 1.1. The journey of computer vision has been characterized by gradual advances in tackling these tasks.

Classification is the simplest task, which involves assigning an entire image to a single class based on its content. For instance, if the image of two dogs lying in the grass from Fig. 1.1 is considered, a classification model would label the entire image as ’dogs’ or ’park’. Image recognition has witnessed the first significant developments of a deep convolutional neural network (CNN), leading to substantial advancements in the computer vision field. In 2012, the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) served as a catalyst for progress in this field. The current models for image classification often assign the top five classes present in the image. These models are able to classify many different images with a startling preciousness, achieving even higher consistency than various human operators [1].

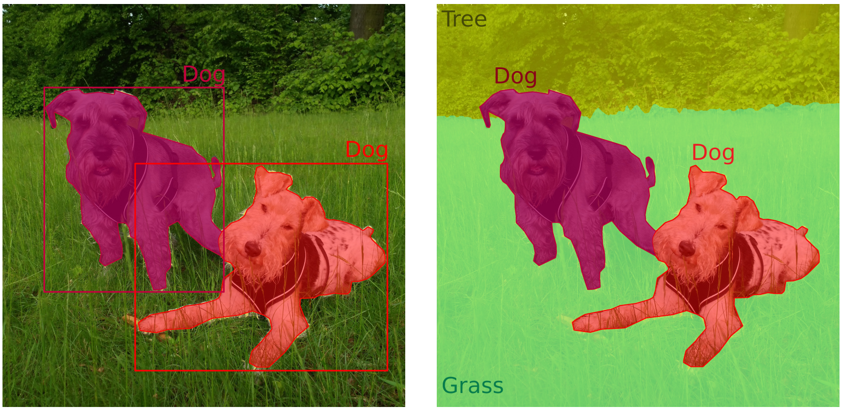
The task of object detection advances this further by identifying and locating objects within an image, marking them with bounding boxes. For the image of two dogs in a park, an object detection model would locate and label each ’dog’ and draw a rectangle around them. It is closer to a real-world application but simultaneously is more complex. In 2014, Girshick et al. introduced the RCNN framework [30], which combined region proposal algorithms with convolutional



Original Recognition/Classification



Object Detection Semantic Segmentation



Instance Segmentation Panoptic Segmentation

Fig. 1.1 Illustration of the fundamental computer vision tasks from a computer science point of view.

neural networks (CNNs), revolutionizing object detection accuracy. Since object detection is one of the themes discussed in this thesis, subsection 3.2 describes the historical development and current state of the art of object detection methods in more detail.

Another approach to understand information coded in an image is semantic segmentation. Here the model classifies each pixel of the input image. The output then consists of the binary or multi-label mask assigning each input image pixel a class. For our image of two dogs in a park, each pixel would be classified as either part of the 'dogs', the 'grass' or 'tree'. Early attempts at semantic segmentation were made using traditional image-processing techniques such as thresholding. In 2014, the introduction of fully convolutional networks (FCNs) by Long et al. [55] revolutionized semantic segmentation by enabling end-to-end trainable architectures. The subsection 3.3 describes the fundamental principle of segmentation models and the current state of the art.

Instance segmentation combines elements of object detection and semantic segmentation. This technique distinguishes different instances of the same class, meaning it could differentiate and label each individual 'dog' in the park, even if they are close to each other.

Finally, panoptic segmentation [46] unifies semantic and instance segmentation, providing a holistic understanding of the image. It simultaneously labels all pixels according to their semantic class and identifies individual instances of objects. As such, it distinguishes 'background' classes like 'grass' and also labels 'foreground' objects such as each individual 'dog'. This results in a highly detailed and informative representation of the image scene.

1.2 Applying Computer Vision to Agriculture

Precision agriculture aims to establish an effective crop management system, leveraging accurate monitoring of plant health and crop physiological status to inform cost-effective fertilization, pesticide application, and plant management strategies [27]. A crucial component of this concept is acquiring detailed, frequent information about a representative amount of plants or field sections. For instance, the Ohio Agricultural Research and Development Center operates a web-based decision support system for greenhouse climate control [76], while other studies further refine this approach by examining optimal degrees of microclimate parameters, such as air temperature and relative humidity [74].

Despite the wealth of data gathered through various sensors monitoring the plants, augmenting this information with new sources can always improve the accuracy of decision-making processes. In this context, computer vision techniques emerge as valuable tools, capable of generating insights not easily derived from traditional sensor data.

This thesis aims to contribute to this evolving landscape by developing a novel tomato fruit detection and counting method using deep learning applied to computer vision. By determining the size and location of fruits within the greenhouse, this system can be utilized for precise harvest prediction. Discrepancies between accurate harvest predictions and actual yields can reveal previously unidentified plant stress conditions in greenhouses, enabling early interventions.

Moreover, harvest prediction is crucial for optimal crop management and significantly impacts the commercial aspects of tomato cultivation in greenhouses. The tomato delivery contracts are based on tomato harvest predictions in the following week. Greenhouse management is often committed to delivering a specific volume of tomatoes on particular days or weeks. Therefore, any deviation between the predicted and actual harvest can lead to substantial commercial losses and logistical challenges. This reinforces the necessity for accurate, reliable prediction models – one of the primary goals of this work is to tackle the large scale of fruit counting in the context of a commercial tomato greenhouse.

A comprehensive review of the state of the art in this field, detailing the latest applications and advancements of deep learning in agriculture, is presented in section 3.5. This will provide the necessary context and groundwork for understanding the innovations brought in this study.

1.3 Small Object Detection in High-Resolution Images

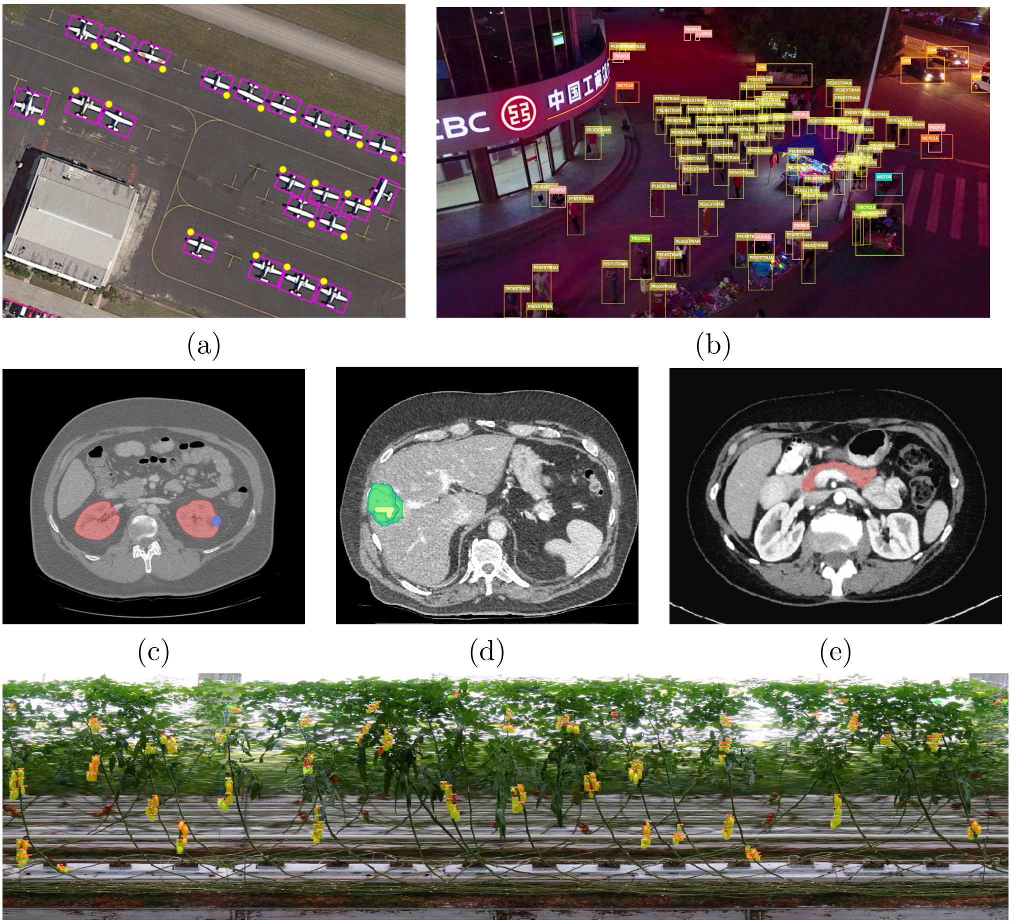
Detecting small objects within high-resolution images remains an open problem in the field of computer vision and constitutes an active area of research. Despite their significant achievements, current deep learning techniques often struggle to identify and correctly classify small objects due to their limited spatial resolution and the complex backgrounds within which these objects are located.

While this problem is frequently encountered in fields such as satellite imagery analysis and surveillance systems, its relevance extends to various sectors, including medical imaging and agriculture. Fig. 1.2 presents example images representing the challenging small object detection, a satellite imagery example, an image from a drone, and a picture of a tomato row in a greenhouse (from a custom dataset named Tomato360, created as part of this thesis).

The disparity in scale between the overall image size and the target objects of interest presents significant difficulties. Conventional convolutional neural networks (CNNs) may struggle with such tasks due to factors such as the reduced feature representation of small objects, the overwhelming dominance of background features, and the increased computational demands associated with processing such large images.

One possible approach to address these challenges is the incorporation of attention mechanisms into the model architecture. Attention mechanisms have the potential to improve the model's focus on the small but crucial parts of the input, thereby enhancing its performance in detecting small objects.

The aim of this work is to explore these possibilities, devising methods that improve the accuracy and efficiency of small object detection within high-resolution images. Section 3.6 offers a detailed account of existing research and advancements in the field of small object detection, while section 3.4 describes different approaches to incorporate an attention mechanism, setting the stage for the innovative approaches proposed in this dissertation.



(a) (b)

(c) (d) (e)

(f)

Fig. 1.2 Example images representing the challenging small object detection, (a) satellite image: DOTA dataset [22], (b) surveillance: Visdrone dataset [99], (c-e) - medical images: KiTS dataset [59], LiTS dataset [5], MSD dataset, pancreas segmentation [78], (f) Tomato row in a greenhouse: dataset created and analyzed in this thesis.

2 Aims of Doctoral Thesis

The following items are proposed as aims of the dissertation:

1. **Appraise the current state of the research area:** Specifically, deep learning methods applied in computer vision with a particular focus on small object detection and segmentation in high-resolution images.
2. **Develop and curate a custom dataset in a tomato greenhouse:** The creation of a custom, real-world dataset aims to demonstrate the transfer of AI technologies from theory to practical implementation. This involves acquiring, collecting, and labeling high-resolution images that capture the challenges specific to this domain.
3. **Investigate and compare the effectiveness of attention mechanisms:** Explore possibilities of incorporating attention mechanisms into different convolutional neural network (CNN) architectures. Compare their performance in terms of accuracy and computational efficiency.
4. **Develop an enhanced deep learning pipeline:** Design and develop a novel processing pipeline tailored to handle the challenging task of small object detection in high-resolution images.
5. **Evaluate the proposed pipeline on the custom dataset:** Apply and test the developed processing pipeline on the custom dataset from the tomato greenhouse. Measure its performance against existing standard techniques used for small object detection. Assess and compare the proposed pipeline's accuracy, robustness, and efficiency.
6. **Analyze the impact and practicality of the proposed methods:** Conduct a comprehensive analysis to understand the impact of incorporating attention mechanisms and the newly developed processing pipeline on small object detection in high-resolution images. Evaluate their practicality in real-world scenarios, considering factors such as computational requirements, scalability, and generalizability.

3 Literature Review

This chapter brings a comprehensive review of the state-of-the-art methods and techniques relevant to the research of this work. The review starts with an exploration of Deep Learning (DL) in Computer Vision (CV), including the fundamentals of Convolutional Neural Networks (CNNs), their basic layer structure, and the role of graphics processing unit (GPU) acceleration in enhancing computational performance. Then the state-of-the-art techniques in object detection and semantic segmentation are discussed in separate sections since they are relevant to the experiments realized in this dissertation.

The literature review progresses to cover attention mechanisms within CNNs, a promising advancement that aids in the detection of small objects within high-resolution images. These mechanisms, utilized in fields such as medical semantic segmentation and object detection, allow models to focus on regions of interest within an image, potentially improving its performance.

The application of Deep Learning in agriculture, particularly in tasks like fruit detection and counting, is also addressed. This area is of significant interest given the increasing need for automated and efficient farming practices. Accurate fruit detection and counting can provide precise crop estimates, aid in planning, and even help identify diseases and pest infestations early. Again, this is relevant to the experiments realized in this thesis, which deals with a custom-made dataset captured in a tomato greenhouse.

The final section dissects the challenges and techniques associated with processing high-resolution images. Due to their large-scale nature, effectively handling these images necessitates specialized approaches to ensure that models can efficiently process and extract meaningful information. The last section examines strategies from current literature for managing high-resolution images, specifically within the context of small object detection. The insights gathered from this review inform the design of a tailored deep learning pipeline, addressing the challenges posed by small object detection in high-resolution images.

3.1 Deep Convolutional Neural Networks

Deep learning is a term most commonly connected to models composed of multiple processing layers, such as deep neural networks, deep belief networks, recurrent neural networks, and convolutional neural networks. Its composite structure allows the models to progressively extract features with an increasing level of abstraction from the raw input. These models dramatically improved the state-of-the-art in many fields such as speech recognition, language translation, computer vision, and many others [47].

The computer vision research area is recently mainly occupied by deep convolutional neural networks (deep CNN) [38]. A convolutional neural network (CNN) is a shift-invariant or space-invariant artificial neural network [96]. CNNs are regularized versions of multilayer perceptrons. Groups of neurons share their weights which give the model translation invariance characteristics. The name "convolutional neural network" cames from a mathematical operation called convolution employed through the network instead of general matrix multiplication. The convolution is a special kind of linear operation.

The connectivity pattern between neurons in CNN is biologically inspired and resembles the organization of the animal visual cortex (the part of a brain dedicated to processing visual information) [40, 70]. Individual neurons react to stimuli only in a restricted region of the visual field. This region is known as the receptive field. The receptive fields of different neurons partially overlap, covering the entire visual field. Similarly, the output matrix values from the convolution operation are calculated from overlapping regions of the input.

CNN requires relatively low preprocessing of input data compared to classical image processing methods, where the initial filters for feature extraction are hand-designed. CNN learns these filters directly from the training data. This independence from prior knowledge and human effort is a significant advantage.

3.1.1 Artificial Neural Network Layers

A CNN consists of an input and an output layer, as well as multiple hidden layers. A series of convolutional, pooling, fully connected, and non-linearity layers are typically present in a CNN model.

The convolution Layer (CL) convolves the input and passes its result to the next layer. The input is a tensor with shape *(batch size)* × *(image width)* × *(image height)* × *(image channels)*, after passing through the layer, the image becomes abstracted to a feature map, with shape *(batch size)* × *(feature map width)* × *(feature map height)* × *(feature map channels).* In a basic setup, a CL has three hyper-parameters:

- the size of a convolutional kernel, which defines the receptive field,
- the stride parameter, which defines how much the receptive fields overlap,
- and the number of input and output channels.

The Pooling Layer (PL) reduces the dimensions of the data, usually combining multiple values of neighboring image pixels into one output pixel. PL typically involves maximum operation, choosing only the highest value of input pixels, or average operation, which uses the average value from input pixels. In a basic setup, a PL has two hyper-parameters:

- the size of a kernel, which defines how much the output is down-sampled, typically 2 × 2,
- the pooling operation, i.e., maximum, minimum, average...

The Fully Connected Layer (FCL) connects every neuron in one layer to every neuron in another layer. It is typically present at the end of the network where it converts the processed features into the final model decision.

The Non-linearity Layer (NLL) applies the non-saturating activation function, such as ReLU (a rectified linear unit 3.1), the saturating hyperbolic tangent 3.2, or sigmoid function 3.3. These layers effectively remove negative values from an activation map by setting them to zero and thus increases the nonlinear properties of the overall network.

$$f(x) = max(0, x) \tag{3.1}$$

$$f(x) = thanh(x), f(x) = |thanh(x)| \tag{3.2}$$

$$\sigma(x) = (1 + e^{-x})^{-1} \tag{3.3}$$

All the above-mentioned layers are usually connected in a feed-forward manner forming the final model architecture. Authors of [34] introduced the skip connections, also linking not directly neighboring layers. These shortcuts help information to flow through a network, increasing both the training speed and the final network performance. The design of model architecture, i.e., the number of layers, their order, and the layout of skip connections, is a subject of extensive research.

3.1.2 GPU Acceleration

The term deep learning goes back to 1986 when it was introduced to the machine learning community by Rina Dechter [20]. However, only the advances in hardware have enabled renewed interest in deep learning. In 2009, Nvidia introduced the possibility to train deep learning neural networks with Nvidia graphics processing units (GPUs). GPUs proved to be well-suited for the matrix/vector computations involved in machine learning and speed up training algorithms by orders of magnitude, reducing running times from weeks to days [57].

Significant additional impacts in the image and object recognition had the paper [16] dealing with fast implementations of CNNs with max-pooling on GPUs. In 2011, a deep convolutional neural network-based approach trained on GPU achieved superhuman performance in a visual pattern recognition contest for the first time [16], and from then on, deep CNN became basically the gold standard in all computer vision tasks [38].

3.2 Object Detection

Object detection has a rich history of development within the computer vision community. The main goal of object detection is to identify and locate objects within an image, and it has become more sophisticated over time.

The early years of object detection were dominated by manual feature extraction and machine learning classifiers. Techniques such as Scale-Invariant Feature Transform (SIFT) [56] and Histogram of Oriented Gradients (HOG) [19] were used to describe and capture patterns in images that could be used to detect objects. These features would then be fed into a machine learning algorithm such as Support Vector Machines (SVM) to classify whether an object is present or not.

In 2001, Viola and Jones [85] introduced a novel approach that combined Haar-like features with a cascaded classifier, making real-time face detection possible. This was a major milestone, providing a robust solution to a real-world problem.

However, these traditional methods struggled to scale with the diversity and complexity of real-world images. They were engineered to work under specific conditions and were sensitive to variations in the object's scale, pose, and appearance.

The introduction of Convolutional Neural Networks (CNNs) to image recognition revolutionized the field of object detection, too. CNNs' ability to learn rich, hierarchical representations from raw pixel data made them ideally suited to the task. Subsequently, the focus shifted to applying these methods to object detection. A classical object detector that consists of two parts was created: the first module acted as a region proposal, and the second module was a classifier.

Girshick et al., in works [29, 30] was the first to successfully adapt such a structure utilizing CNNs to the recognition task, but it was computationally expensive due to the independent processing of a large number of region proposals. Fast RCNN [28] addressed this issue by introducing a technique known as Region of Interest (RoI) pooling to share computations. Faster-RCNN [67] went a step further by including a Region Proposal Network (RPN), enabling the detection network to suggest potential object bounding boxes. In 2023, Faster-RCNN is still one of the most popular object detection network architectures, especially in custom computer vision problem solving, due to its availability, training stability, and a good ratio of inference speed and detection precision.

Still, object detection methods have continued to evolve. Modern successful object detector architectures consist of a single feed-forward convolutional neural network (CNN) that directly predicts classes and anchor offsets without the need for a second stage per-proposal classification operation. These detectors allow a few potential bounding boxes to be considered as raw object locations and require predicting an offset to the actual location of the object. Simultaneously, they predict scores for object categories, effectively combining the steps of region proposal and classification. This approach was first proposed in 2016 by You Only Look Once (YOLO) [66]. The Single Shot Detector (SSD) [54] presents a similar approach but adds layers of feature maps for each scale. The improvement of the SSD detector by combining it with the state-of-the-art classifier (Residual-101 [34]) is presented in work [26].

To provide a concrete illustration of how models perform in terms of accuracy and evaluation speed, consider the following example. In 2018(1), the state-of-the-art general-purpose detector was RetinaNet [52]. Its best model can detect

---

(1) Starting from 2019, the COCO object detection challenge only features the detection task with object segmentation output (that is, instance segmentation).

more than 80 categories with mean average precision (mAP)$^{2)}$ 55.2 at 0.5 intersection over union (tested on COCO$^{3)}$). With this setting, the model reached the evaluation speed 122 ms on an Nvidia M40 GPU.

The field of object detection continues to develop, with ongoing research seeking to enhance speed, accuracy, and robustness. Object detection systems have wide-ranging applications, from autonomous driving and video surveillance to medical imaging and augmented reality.

3.3 Semantic Segmentation

Many of the classic image segmentation methods consist of some thresholding variant. If the image pixel satisfies a condition regarding its level of color or brightness, it is assigned a class. This straightforward approach faces many difficulties if the object of interest has a variable appearance or the scene's lighting is uneven. This trouble can sometimes be fixed by applying the adaptive threshold and other additive conditions. Another conventional technique for solving the segmentation problem is the region-growing method – watershed algorithm. This approach was used, for example, in the task of cell nuclei detection and segmentation [50]. Many region-growing algorithms result in over-segmented images, i.e., too many object regions are formed.

Aside from thresholding, there are edge-based segmentation techniques. In this approach, the edges are detected first, and only then the segmented regions are located. This approach is especially useful when looking for an object with a stable shape, i.e., the human eye or iris. Hough transform is a method that detects distinctive contours in the image, i.e., lines or circles [41].

In 2014, the introduction of fully convolutional networks (FCNs) by Long et al. [55] revolutionized semantic segmentation by enabling end-to-end trainable architectures. Since then, the research area dealing with semantic segmentation

---

2) Common evaluation metrics used in computer vision are introduced in sec. 4.4.

3) For more information about this dataset please visit http://cocodataset.org/#home.

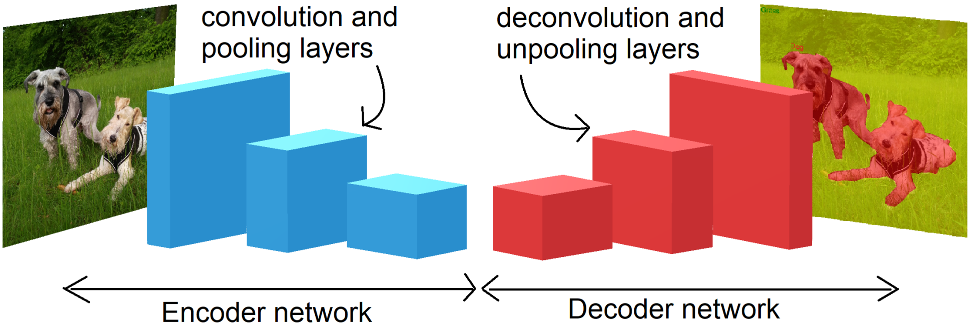


Fig. 3.1 The schema of encoder-decoder network structure.

is mainly occupied by deep convolutional neural networks [38]. A general semantic segmentation architecture often consists of an encoder network followed by a decoder network. The encoder is often a pre-trained classification network like VGG [77] or ResNet [34] followed by a decoder network. The task of the decoder is to semantically project the discriminative features (lower resolution) learned by the encoder onto the pixel space (higher resolution) to get a dense classification.

The schema of the encoder-decoder network structure is in Fig.3.1. The encoder usually consists of convolutional and pooling layers; the image resolution gradually decreases while the number of feature channels increases. The decoder usually incorporates unpooling and deconvolutional layers. It reversely mimics the encoder structure, increasing the image resolution while decreasing the number of feature channels, for more information about layers used in convolutional neural networks, refer to section 3.1.

The presence of several convolutional and pooling layers in the network architecture brings the problem that the resolution of the output feature maps is downsampled. Therefore, the resulting object boundaries are relatively fuzzy. A variety of network extensions directed to address this issue have been proposed in the literature, including SegNet [4], Unet [71], and DeepLab [14].

The Unet model series, starting with the original Unet proposed by Ronneberger et al. [71], has become highly influential in the field of biomedical image segmentation. Unet's architecture, mirroring the encoder-decoder structure, introduced a new approach where the decoder part uses high-resolution features from the encoder part directly. This skip-connection scheme allows the model to utilize both high-level semantic and low-level spatial information effectively, thereby enhancing the precision of segmentation results. This basic structure has been adopted and improved in numerous subsequent models, proving its effectiveness in various scenarios. One such very successful implementation is nnUnet presented by Isensee et al. in 2020 [42], which sets a new state of the art in the majority of tasks it was evaluated on, outperforming all respective specialized processing pipelines. Even more interesting is that the strong performance of nnUnet is not achieved by a new network architecture, loss function, or training scheme (hence the name nnUnet - "no new Unet") but by replacing the complex process of manual pipeline optimization with a systematic approach based on explicit and interpretable heuristic rules.

The DeepLab model series, introduced by Chen et al. [11], has significantly contributed to the semantic segmentation domain. The series begins with DeepLabv1, which introduced atrous (dilated) convolutions to control the resolution of feature responses in the network explicitly. In subsequent versions, DeepLabv2 [12], DeepLabv3 [13], and DeepLabv3+ [14], the authors incrementally incorporated various enhancements like atrous spatial pyramid pooling (ASPP), encoder-decoder structure, and depthwise separable convolutions, effectively pushing the state-of-the-art in semantic segmentation.

These advancements in Unet and DeepLab series have opened up new possibilities for semantic segmentation tasks. Their significant performance in complex segmentation tasks sets a benchmark for future models, while their design principles provide valuable insights for the development of more sophisticated segmentation algorithms. One such research direction is the incorporation of the attention mechanism, which is discussed in the following section 3.4.

3.4 Attention Mechanism in CNN

Many research papers have incorporated attention into artificial CNN visual models for image captioning [92], classification [58, 91] and segmentation [10]. For example, in the case of Recurrent Neural Networks (RNN), [95] presents an RNN model that learns to sequentially sample the entire X-ray image and focus only on salient areas. In these models, attention could be divided into two categories: hard and soft attention. As described by [92], hard attention is when the attention scores are used to select a single hidden state, e.g., iterative region proposal and cropping. Such an attention mechanism is often non-differentiable and relies on reinforcement learning to update parameter values, making training quite challenging. On the other hand, soft attention calculates the context vector as a weighted sum of the encoder's hidden states (feature vectors). Thus, soft attention is differentiable, and the entire model is trainable by back-propagation.

The attention modules which generate attention-aware features presented by [87] was the state-of-the-art object recognition performance on ImageNet in 2017. The work [39] presents a Criss-Cross Network (CCNet) with a criss-cross attention module and achieves the state-of-the-art results on Cityscapes test set and ADE20K validation set, respectively. The paper [33] combines deep CNN architecture with the components of attention for slice-level predictions and achieves 81.82% accuracy for the prediction of hemorrhage from 3D CT scans, matching the performance of a human radiologist. Other boosted CNN with attention and deep supervision (DAB-CNN) [44] achieves state-of-the-art results in automatic segmentation of the prostate, rectum, and penile bulb.

3.4.1 Attention Gates

Medical image segmentation, specifically automatic abdominal organ segmentation from CT images, presents significant challenges for deep CNN models [37]. One such challenge is how to automatically locate the anatomical structures in the target image because different organs lay close to each other and can also

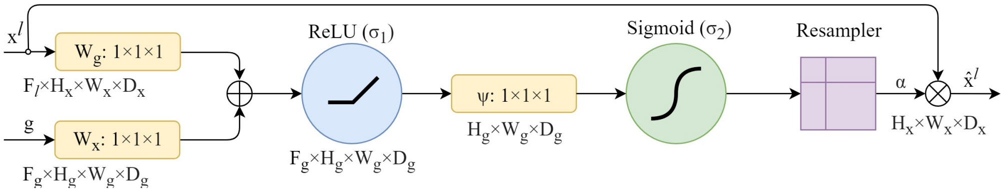


Fig. 3.2 A block diagram of additive attention gate (AG) [62]. Input features (x^l) are scaled with the attention coefficients (α) computed in AG. Spatial regions are selected by analyzing both the activations and the contextual information provided by the gating signal (g) which is collected from a coarser resolution scale. Attention coefficients are resampled to match the resolution of (x^l) by trilinear interpolation.

overlap. Moreover, among individual patients exists a considerable variation in the location, shape, and size of organs. Furthermore, abdominal organs are characteristically represented by similar intensity voxels as identified surrounding tissues in CT images. The other challenge is to determine the fuzzy boundaries between neighboring organs and the soft tissues surrounding them.

The task of detecting cancerous tissue in an abdominal organ is even more difficult because of the large variability of tumors in size, position, and morphology structure. Results are quite impressive when the focus is on organ detection; an example of this is [42], achieving Dice scores[4)] of 95.43 and 79.30 for liver and pancreas segmentation. On the other hand, these values drop dramatically when the focus is on detecting the tumor, where values are as low as 61.82 and 52.12 for their respective (liver and pancreas) tumor classes. There is also a high variability on tumor classification depending on the organ, e.g., [94] presents Dice scores of 93.1 and 80.2 when the organ is the kidney and its tumor detection, respectively.

On the other hand, all the organs have a typical shape, structure, and relative position in the abdomen. The model could then benefit from an attentional mechanism consolidated in the network architecture, which could help to focus

---

[4)] Dice score [21] is a common metric used in medical image segmentation, value 100 means 100% conformity with ground true, for more information about this metric see sec. 4.4

specifically on the organ of interest. One successful application of attention in the context of medical image segmentation incorporated the idea of attention gates (AG) [62]. Attention gates identify salient image regions and prune feature responses to preserve only the activations relevant to the specific task and suppress feature responses in irrelevant background regions without the requirement to crop the region of interest. The principle of the attention gates is tested in the results section of this thesis, in section 5.2. Therefore, a more detailed description of the attention gates is provided here for reference.

Attention coefficients, $\alpha_i \in [0,1]$ emphasize salient image regions and significant features to preserve only relevant activations specific to the actual task. The output of Attention Gates (3.4) is the element-wise multiplication of input feature maps and attention coefficients:

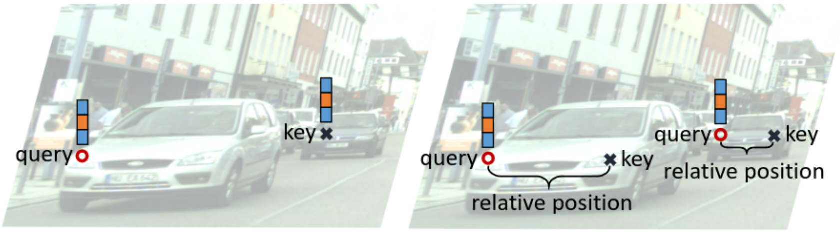
$$\hat{x}^l_{i,c} = x^l_{i,c} \cdot \alpha^l_{i,c} \tag{3.4}$$

where $\alpha^l_{i,c}$ is the attention coefficient (obtained using equation 3.6, below), and $x^l_{i,c}$ is pixel i in layer l for class c. $x^l_i \in R^{F_l}$ where F_l corresponds to the number of feature-maps in layer l. Therefore, each AG learns to focus on a subset of target structures. The structure of an attention gate is shown in Fig. 3.2. A gating vector g_i is used for each pixel i to determine the regions of focus. The gating vector contains contextual information to reduce lower-level feature responses. The gate uses additive attention (3.5), formulated as follows [62]:

$$q^l_{att} = \psi^T(\sigma_1(W^T_x x^l_{i,c} + W^T_g g_{i,c} + b_g)) + b_\psi \tag{3.5}$$

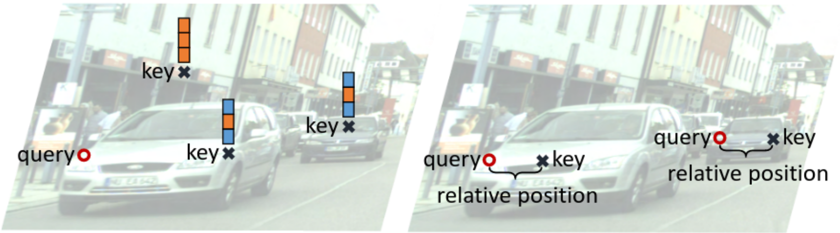
$$\alpha^l_{i,c} = \sigma_2(q^l_{att}(x^l_{i,c}, g_{i,c}, \Theta_{att})), \tag{3.6}$$

where $\sigma_1(x^l_{i,c}) = max(0, x^l_{i,c})$ is rectified linear unit. AG is characterised by a set of parameters Θ_{att} containing: linear transformations $W_x \in R^{F_l \times F_{int}}$, $W_g \in R^{F_g \times F_{int}}$, $\psi \in R^{F_{int} \times 1}$ and bias terms $b_\psi \in R$, $b_g \in R^{F_{int}}$. $\sigma_2(x^l_{i,c}) = \frac{1}{1+exp(-x^l_{i,c})}$ corresponds to a sigmoid activation function. The linear transformations are computed using channel-wise $1 \times 1 \times 1$ convolutions of the input tensors. All the AG parameters can be trained with the standard back-propagation updates.



(a) query and key content

(b) query content and relative position



(c) key content only

(d) relative position only

Fig. 3.3 Illustration of different attention terms [100]. The color bar above a sampling point denotes its content feature. The existence of content features and/or relative position indicates that the term uses them for attention weight calculation.

3.4.2 Spatial Attention Mechanisms in Deep Networks

The landmark work of Transformer [84] set a new standard, and its latest variants use relative positions instead of absolute positions for better generalization ability [75, 18]. The Transformer attention presented in [18] has attention weights expressed as a sum of four terms $(\epsilon_1, \epsilon_2, \epsilon_3, \epsilon_4)$. Specifically, these factors are (1) the query and key content, (2) the query content and relative position, (3) the key content only, and (4) the relative position only. In the vision, the key and query refer to visual elements, but aside from that, a formulation similar to Transformer attention introduced in [18] can be used. The visual meaning of each attention factor is illustrated in Fig. 3.3.

The paper [100] follow on the success of the work of [18] and presents an empirical study of spatial attention mechanisms in Convolutional Neural Networks (CNNs). The authors propose a new spatial attention module that can be easily integrated into existing CNN architectures. They also provide a comprehensive

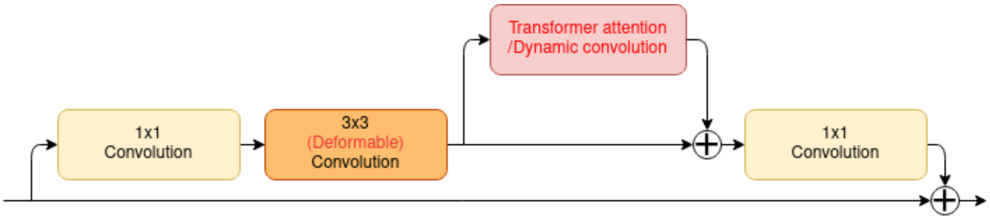


Fig. 3.4 Attended residual block [100], implementing spatial attention into feature extraction part of deep convolutional neural network for object detection. The modules added to existing blocks are marked in red text.

comparison of their proposed module with existing spatial attention mechanisms. For the object detection task, the authors recommend using a "query content and relative position" attention term as the most beneficial in terms of solution precision and efficiency. The incorporation of all attention factors might bring slightly better results but needs 30% more GFLOPS for a computation.

This spatial attention mechanism is incorporated in the results section 5.3, where a more detailed description of the implementation for the object detection task is provided too. Within the feature extraction part of the network, the attention mechanism extends a classical residual block common to most object detection networks utilizing ResNet[34] as a backbone. The attended residual block is visualized in Fig. 3.4. The modules added to existing blocks are marked in red text. The attention mechanism is also combined with the deformable convolutions [101], which position is illustrated in the diagram, too.

3.5 Applications of Deep Learning in Agriculture

In the continually evolving field of agriculture, cutting-edge technologies are ceaselessly integrated into farming practices to optimize productivity, enhance sustainability, and boost profitability. Among these technologies, computer vision holds a pivotal role in augmenting agricultural operations. A significant subset of applications pertaining to computer vision in agriculture pertains to harvest prediction and fruit counting. These tasks, when performed with precision, can profoundly influence crop management strategies and affect the economic aspects of cultivation. This section provides a comprehensive examination of the current state-of-the-art, detailing the application of computer vision techniques in agriculture, with a particular focus on their use in harvest prediction and fruit counting. The review of these methods and models will underpin the further development and discussion of this study's unique contribution to the field.

Many researchers have investigated fruit detection over the past several decades. Authors of [98, 31] brought a valuable overview and significant improvements in the field. Traditional image processing techniques have many constraints, making applying the algorithms to other fruits or environments hard. They often struggle with uneven illumination and incorrect leaf detection [88]. The growth and development of artificial intelligence techniques enabled the application of machine learning to computer vision tasks in agriculture, and it has practically dominated the field.

Authors of [86] present a multi-class fruit detection system utilizing an adjusted Faster-RCNN model. The paper focuses mainly on model adjustments and training process automation. For this purpose, it utilizes the artificially created dataset [61] where the fruits were filmed while rotating around a fixed axis on a white background. On this dataset, they achieved a mean average precision of 88.94%. Unfortunately, the results for the tomato class are not stated in the paper; even though it is present in the original dataset, the authors present results only for apple, mango, and orange.

The paper [93] proposes an improved YOLOv3-tiny architecture for tomato detection in real time. The experimental results show that the F1-score of the model is 91.92%, while the detection speed on a CPU can reach 25 frames/s. Another paper also utilizes the YOLO-based architecture, presenting a YOLO-tomato detector [53]. It modifies the well-known YOLOv3 architecture for object detection by adding dense connections between the layers for feature extraction and by applying the circular bounding box instead of the classic rectangle. YOLO-tomato model achieves an even better 93.91% F1-score.

The above-referred research only focused on tomato detection in images directly capturing a tomato or a tomato cluster. Compared to that, the paper [60] analyses the images of the whole tomato plant from a greenhouse, similarly as is proposed in this thesis. The authors of paper [60] utilized a Faster-RCNN with Resnet101 backbone model and obtained an F1-score of 83.67%. Compared to the previously stated research, the performance drop can be accounted to a more compilated scenario, including more small immature tomatoes and frequent obstruction by leaves. Although the authors provide the yield mapping of the whole image row by stitching the images together, the counting results were estimated only on single images, which should, by principle, bring additional errors into the final count: the tomatoes on the overlapped borders would be counted twice.

A few research papers deal with fruit counting on a large scale, i.e., accessing the whole crop counts. The authors of [63] attempt to produce a real-time pear fruit counter using a video captured by a mobile phone capturing the bottom side of the joint-tree pear orchard. It utilizes real-time object detection by the YOLOv4-tiny model and consequently applies the unique ID deep sort method for pear counting, achieving an F1-score of 87.85%. Compared to the scenario in the tomato greenhouse, pears do not form trusses/clusters, and the sky forms a relatively uniform background, making the detection process more straightforward.

Finally, the paper [8] presents a similar goal as this thesis proposes to solve, delivering an approach for detection, counting, and maturity assessment of cherry tomatoes but using multi-spectral images. Authors train deep CNN networks on images captured in a tomato greenhouse. Consequently, similar to the paper [63], they apply the deep sort algorithm to track the tomatoes in a video. The value F1-score is not stated; the IDF1-score achieved by the best solution is 51.4%. The results show that detecting and tracking objects in complicated and obscured scenes of tomato growth is very challenging.

The paper from the author of this thesis [A1] proposes a different solution, also utilized in the results part of this thesis. This approach eliminates the tricky part of object tracking, making the whole process easier. The objects are not detected in video frames, but rather an extra wide image capturing the whole tomato plant row was produced first. This big image is cut into overlapping patches, which are processed by an object detector, and the output predictions are then stitched together to form the final output. The extended study of the post-processing parameters was performed to ensure the best possible outcome of the stitching procedure. The following section 3.6 discusses different possibilities of how are such high-resolution images with small objects within processed in state-of-the-art research.

3.6 Processing High-Resolution Images

High-resolution images present a complex problem in computer vision, particularly when the task involves detecting small, sparsely distributed objects. The combination of a vast amount of data in high-resolution images and the requirement to identify small details significantly escalates computational demands. This complexity can lead to extended processing times and increased resource needs, often making conventional object detection methods less effective or impractical.

These challenges are commonly seen in contexts such as aerial and surveillance imagery, where small objects of interest make up a tiny fraction of the total image data. The following paragraphs discuss selected research papers that propose potential solutions, emphasizing the need for novel techniques to effectively process high-resolution images in such challenging scenarios.

3.6.1 Cropping Image Patches

Many current methods utilize the notion of cropping high-resolution images into sequential subregions or chips for detection. Very successful implementation of such an approach was brought by [2]. This technique aids in data augmentation during the training phase and supports the inference phase, with the final prediction resulting from a combination of detections made in the original image and image patches. A computer vision library[5)] for performing large-scale object detection and instance segmentation is made available by the author of this paper, creating a stable reusable baseline for other methods.

Nevertheless, the patch-cropping technique is not without issues. The need for carefully chosen crop sizes presents one such challenge. Too-small patches risk excluding larger objects, while overly large patches may lack necessary detail. Balancing these factors often necessitates the sampling of different patch sizes, significantly increasing processing time and resource demands. Furthermore, objects may get sliced during image partitioning, complicating both training and detection, though overlapping patches can somewhat alleviate this issue.

Adopting an innovative approach, the creators of AMRNet [89] introduce an adaptive cropping schema that employs a scale statistic. Adjustments to patch size are made according to the object size, with padding used to enlarge patches and partitioning used to create smaller ones. While this adaptive scaling is only utilized during training due to the unavailability of object size information in the test set, the AMRNet still manages to achieve a significant increase in Average Precision metrics when compared to a then-current baseline on the Visdrone dataset.

The paper [83] proposes to use a reinforcement learning agent that adaptively selects the spatial resolution of each image, choosing to sample high and low-resolution patches. In particular, they trained two policy networks, using reinforcement learning with the dual reward of maintaining accuracy while maximizing the use of low-resolution images with a coarse detector. This increases the runtime efficiency by 2.2x but brings a need for complicated reinforcement learning in a training phase.

[5)] The python codes are available at github.com/obss/SAHI

Finally, a distinct strategy is proposed in [49], leveraging a density map prediction initially introduced in [97]. The authors developed the Density-Map guided Object Detection Network (DMNet), which generates a density map and learns scale information based on density intensities to form cropping regions. The DMNet capitalizes on the fact that an image's object density map displays object distribution in relation to the pixel intensity of the map. This pixel intensity variation reveals whether a region contains objects, providing valuable guidance for statistically cropping images.

3.6.2 Multiple Detection Suppression

Techniques suppressing repeated object detection play a crucial role in object detection, particularly in methodologies employing patch cropping. Object detection models often propose multiple bounding boxes surrounding a potential object within an image. However, this could lead to an issue known as multiple detections, where several bounding boxes are proposed for the same object. This is especially pronounced in patch-cropping methods, where each cropped patch is processed independently, potentially leading to overlapping predictions for the same object in adjacent patches.

To resolve this issue, Non-Maximum Suppression techniques are utilized. Non-maximum suppression (NMS) has been an integral part of many detection algorithms in computer vision for almost 50 years. It was first employed in edge detection techniques [72]. For human detection, Dalal and Triggs [19] demonstrated that a greedy NMS algorithm, where a bounding box with the maximum detection score is selected and its neighboring boxes are suppressed using a predefined overlap threshold improves performance over the approach used for face detection [85]. Greedy NMS still obtains the best performance when average precision (AP) is used as an evaluation metric and is therefore considered the gold standard and is employed in popular detector Faster-RCNN [68].

Over the years, some new approaches have been presented. Authors of [7] propose Soft-NMS, which attempts to address the problem of classic NMS, which suppresses even a correctly detected object if the object lies within the predefined overlap threshold. The algorithm decays the detection scores of all other objects as a continuous function of their overlap with the first most confident one. Hence, no object is eliminated. Just the confidence score is lowered. Another approach is presented in [15], where the algorithm judges the neighboring bounding boxes of each bounding box and combines the neighboring boxes that are strongly correlated with the corresponding bounding boxes. This approach was designed, for the instance segmentation task, showing a steady increase in accuracy.

The case of stitching image patches deals with a slightly different problem of multiple detections. The original wide image is split into overlapping patches; consequently, there are multiple detections in the overlapping areas of the final output. Therefore, aside from a classical suppression after model inference, post-processing suppression needs to be applied to the final stitched output, too. An extensive study of different post-processing suppression algorithms to achieve the best possible output is presented in the paper from this thesis author [A1].

4 Methodology

In order to develop a successful deep convolutional neural network model, an extensive and complex workflow is necessary. The quality of the established pipeline frequently has a significant impact on the final model results [42]. This part, therefore, summarizes the essential building blocks of a robust workflow, discussing the actual trends in the application area of computer vision. On top of that, a novel method for small object detection in high-resolution images is proposed. The novel methodology consists of two novel techniques, namely: Artificial Size Slicing Fine Tuning and Artificial Size Slicing Hyper Inference.

4.1 Dataset Collection and Preprocessing

Although one of the advantages of deep CNN models is their ability to extract significant features without the need for extensive, human-designed preprocessing, the proper understanding of training data and its appropriate adjustments are essential for successful model development. Authors of [82] discussed the presence of bias in image data collection, defining its leading causes and consequences. The *capture bias* is connected both to the utilized device and to the acquisition conditions, i.e., point of view, lighting conditions, etc. The *label* or *category bias* is caused by high in-class variability and a poor class semantic definition, caused by a disagreement between different expert annotations. This bias can cause problems, especially in the medical field, where even the experts may disagree on the correct answer. Finally, there is the *negative set bias.* The negative set defines what the model takes as "the rest of the word". If this set is too extensive or unbalanced, it may cause the model to be overconfident and not very discriminative.

When using the publically available datasets, it is usually hard to change the category and the negative set bias since they are directly connected to the data collection and annotation, which usually are not available anymore at the moment of model training. On the other hand, the capture bias could be managed or at least decreased by image preprocessing and normalization.

4.1.1 Dataset Normalization

Dataset normalization consists of the manipulation of images from a dataset in a way, so they are generally more consistent, and for the model, it is easier to process the images in a unified manner [32].

- **Scaling:** scaling to a range means converting floating-point feature values from their natural range (for example, 100 to 900) into a standard range—usually 0 and 1 (or sometimes -1 to +1). This normalization is useful, if the estimated upper and lower bounds of the data are known, and if the data are relatively uniformly distributed across that range. An example can be the z-score normalization applied based on the mean and standard deviation of data values, i.e., pixel intensity values.

- **Clipping:** clipping serves to remove outliers from the data. For example, clip the data to [0.5, 99.5] percentiles or to plus-minus some multiplication of dataset standard deviation.

- **Log scaling:** log scaling computes the logarithm of data values to compress a wide range to a narrow range. It is especially useful if some of the values have many data points, while most other values have only few data points.

- **Special normalization techniques:** special types of data may require special normalization techniques. For example, volume medical image data often needs to be resampled to the median voxel spacing of the dataset, spline interpolation, or nearest-neighbor interpolation are commonly used for this purpose [6].

4.1.2 Image Augmentation

Convolutional neural networks are quite a powerful tool but are heavily reliant on huge amounts of data to avoid overfitting. Data augmentation techniques systematically enlarge the training dataset by explicitly generating more training samples. Therefore they are effective in improving the generalization performance of deep convolutional neural networks [48, 35]. The following list summarises the techniques most commonly used in computer vision: random rotations, random scaling, random elastic deformations, gamma correction augmentation, and mirroring. All the augmentation techniques can be eighter administered before the training, explicitly enlarging the number of training examples, or can be applied on the fly during training, saving memory usage but demanding higher processing time.

4.1.3 Slicing Aided Fine-tuning

In various fields, including satellite imagery analysis, surveillance systems, medical imaging, and agriculture, there can be situations where training images may be too large for a CNN model to process in one go. This problem, relating to small object detection in high-resolution images, is detailed in section 1.3. A common approach to address this issue is to either downsample the images or process the image in patches, which are sub-images extracted from the larger image. These sub-images can overlap, and their size can vary depending on the model and its application.

While downsampling or using larger patches allows the model to capture more contextual information, this comes at the cost of reduced detail. On the other hand, processing in smaller patches provides high-resolution details but at the expense of broader contextual information, as the entire visual scene is not immediately accessible. This necessitates finding a balance between the two extremes. Patch cropping, for instance, is frequently employed in medical image segmentation, which frequently deals with extensive and often multimodal data. Models in this area are typically trained on specific patch sizes that are tailored to the application at hand.

The term "Slicing Aided Fine-tuning" was introduced in paper [2] to describe a data processing pipeline that augments training images with cropped image patches. This process enriches the training data to include both original resolution images and image patches or crops in preparation for Slicing Aided Hyper Inference. The latter is a method that includes predictions based on both the original image and its cropped sections in the final prediction. Details of Slicing Aided Hyper Inference can be found in section 4.3.1. The best prediction performance might be achieved by applying different cropping sizes, though this method significantly increases computational requirements for both training and inference, as it enlarges the training dataset size and requires the model prediction multiple times: once for the original image and then for each image patch.

4.1.4 Artificial Size Slicing Aided Fine-tuning

Certain situations necessitate using image patches for both training and prediction as the input image data is too large for one-time model processing, and simple downsampling can degrade the image to the extent that object detection becomes impossible. This is the case with the custom-made Tomato360 dataset addressed in this work; for more details about this dataset, please refer to section 5.1. In a recent study [A1], the author of this thesis explores the impact of varying image patch sizes on prediction accuracy. To generalize the patch-cropping process, a novel proposal is made in this work to crop artificial-sized image patches centered around object groups, effectively utilizing the fact that tomatoes are growing in trusses. This strategy of augmenting the training data with these patches enhances the model's ability to discern and localize overlapping tomatoes.

During the training phase, generating these artificial-sized image patches is relatively straightforward. The application of this principle in the inference phase on test data is discussed in section 4.3.2; this section also provides a schematic of the newly proposed Artificial Size Slicing Aided Hyper Inference (ASSAHI) process, pictured in a Fig. 4.2. In the case of training data, where instance segmentation masks are available (or the area of boxes in scenarios lacking instance segmentation annotation), a foreground segmentation map was created from all object's instance masks. A binary dilation operation was employed to cluster objects into larger groups, and image patches were cropped in accordance with the position of each connected group. This method not only provides the model with detailed information about small objects in the image but also effectively avoids cutting objects during the patch cropping process.

However, implementing this solution poses a couple of challenges. Firstly, the dilation operation can be computationally demanding in scenarios involving numerous small objects in high-resolution images, especially with larger dilation operator sizes. To improve efficiency, the input image was downscaled, the dilation operation was applied, and the output was subsequently upscaled. This streamlined the process without causing significant damage to the final output. Secondly, the risk of cropping too small image patches around isolated objects in the image was mitigated by imposing a minimum crop size, ensuring such small objects were cropped with a broader context around them.

4.1.5 Dataset Splitting

In machine learning, the dataset is commonly split into three parts:

- **Training set** is a dataset of examples used for learning, that is to fit the parameters (weights) of a model.
- **Validation set** is a dataset of examples used to tune the training hyperparameters, e.g., the optimization parameters as a learning rate or definition of the moment to stop the training.
- **Test set** is a set of examples used only to assess the performance (i.e., generalization) of a trained model.

This division helps to verify if the model has learned the generalized features and can deal with unseen data or if it overfits the training set [32]. Both the validation and test sets should be independent of the training dataset but follow the same probability distribution as the training dataset. In the most simple case, three independent parts are separated randomly from the original dataset. These parts usually have different sizes and are used for training, validation, and testing, respectively. This method is called Holdout.

Cross-validation is a method of repeatedly splitting the training and validation set, so in the end, all the examples are used both for training and validation. The final model performance is estimated as mean performance achieved on all validation subsets. Fig. 4.1 illustrates the common five-fold cross-validation splitting method.

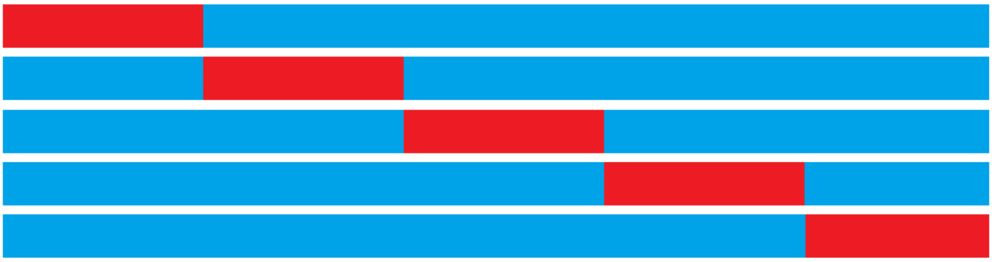


Fig. 4.1 Five-fold cross-validation splitting method, blue color illustrates a training set, red color illustrates a validation set.

4.2 Model Training

Machine learning algorithms learn from examples. The example data point passes through the model, and the final output is compared with the expected result. The difference between the two is usually estimated by some loss function and is called cost. Backpropagation is an algorithm widely used in the training of feedforward neural networks for supervised learning [32]. This algorithm allows the information from the cost to flow backward through the network, compute the gradient, and accordingly adapt the neuron weights [64].

The hyperparameters of training workflow influence the learning process massively — the following subsections highlight which parameters of training workflow should be examined.

4.2.1 Loss Function

The process of model training looks for a particular set of weights with which a convolutional neural network can make an accurate prediction. In order to measure the difference between the model predictions and the ground-true label, a loss function must be defined. The loss function heavily depends on the model application; the traditional loss function ranges from the mean squared error or multi-class cross-entropy loss used for image classification [1] over localization loss for bounding box offset prediction in object detection [67, 66, 52], to pixelwise cross-entropy loss or Dice coefficient loss for image segmentation [71, 42]. The loss function is often chosen with regard to final evaluation metrics, as its choice strongly affects the learning process.

4.2.2 Optimizers

After defining the loss function or cost, the optimizer finds parameter values that achieve the function's minimum. In deep CNN training, the gradient descent algorithm (GDA) creates the foundation of nearly all optimizing algorithms used in practice. GDA iteratively tunes the parameter values to reduce the cost. At every iteration, parameter values are adjusted according to the opposite direction of the cost gradient. Basic GDA is prone to a zigzag movement towards the optimal weights (which slows the learning process) and inclines to get stuck in the local optimum (not finding the global solution at all). In practice, more sophisticated algorithms are used, such as Stochastic Gradient Descent (SGD) with Momentum [69], RMSProp [81], or Adam Optimizer [45]. The last-mentioned combines the advantages of previous methods and becomes a gold standard in deep CNN model training, but Wilson et al. [90] showed that adaptive methods (such as Adam) do not generalize as well as SGD with momentum when tested on a diverse set of deep learning tasks. The optimal optimizer choice always depends on a model architecture as well as on data and a performed task.

4.2.3 Learning Rate

An important hyper-parameter of the training process is the learning rate (LR), which indicates the extent of adjustments made in each iteration step. The same LR can be applied throughout the whole training process, or its value may evolve over time. In the beginning, the learning rate is usually set to a higher value to speed up the training, but it should be decreased as the training gets closer to the global optimum in order to minimize the risk of missing the best solution. Different researchers use different learning rate tactics.

4.3 Model Inference

Model inference refers to a process where the model predicts unseen data that can be from a test set to access the model performance or during a real-world application of the model. The inference process should follow the data processing applied during the training as closely as possible.

The following subsections are devoted to the description of the process involved in Slicing Aided Hyper Inference and the proposed extension Artificial Size Slicing Aided Hyper Inference. The last subsection is devoted to the process of image stitching common to both techniques.

4.3.1 Slicing Aided Hyper Inference

As previously discussed in the section 4.1.3, Slicing Aided Fine-tuning, there are applications where an image may be too extensive for a CNN model to process all at once. In these cases, downsampling could degrade the image to the point where smaller objects become undetectable. Similarly, as the training was augmented by image patches, the inference might be enriched by predicting those patches. Paper [2] proposes Slicing Aided Hyper Inference, where the input image is predicted whole and then also the image is cut into several image patches, whose are predicted too. The final prediction then gathers all the predictions together. The problem of multiple prediction merging is described in the following section 4.3.3.

4.3.2 Artificial Size Slicing Aided Hyper Inference

In accordance with Artificial Size Slicing Aided Fine Tuning (ASSAFT) proposed in section 4.1.4, a similar extension for Slicing Aided Hyper Inference (SAHI) is provided. Following existing nomenclature, this innovative approach is referred to as Artificial Size Slicing Aided Hyper Inference (ASSAHI). To specify the placement of these artificially-sized patches, a mask that identifies object group positions within the original image is required. For this purpose, a semantic segmentation deep convolutional neural network might be trained. This network does not require ultra-precise segmentation annotations but should reliably detect clusters of smaller objects utilizing a greater context of the image. Hence, it should be trained on complete input images or relatively large image patches if slicing is necessary. The example implementation of this method on the Tomato360 dataset is discussed in the results section of this work, in section 5.4. Once the input image mask is generated, slicing follows the same rules established in the ASSAFT. The mask undergoes dilation, and image patches are situated around each detected object/group within the masked image. Those patches are predicted by the model and utilized in the final prediction output creation. The entire ASSAHI procedure is graphically represented in Fig. 4.2.

ASSAHI presents two key benefits. Firstly, it feeds the model with detailed information about smaller objects within the image, enabling more precise detection, which is especially beneficial in scenarios where objects overlap. Concurrently, it effectively prevents objects from being cropped/segmented during patch slicing. Furthermore, in a dataset where object groups sparsely populate the input data, ASSAHI can save a substantial amount of computation resources, which would otherwise be spent by processing many (empty) small image patches sliced by the standard SAHI method, all while maintaining necessary detail for precise small object detection.

On the other hand, ASSAHI may encounter difficulties in datasets where objects do not group or form excessively large groups. In the former situation, a singular small object could potentially be overlooked by the primary segmentation model, thus being excluded from subsequent higher-resolution processing and missed entirely in the final prediction. In the latter case of extensive object groups, such a large group may be accommodated into a single patch whose resolution might exceed the model's handling capabilities. Consequently, the patch would be downscaled during the data loading process, risking the loss of details necessary for distinguishing smaller objects.

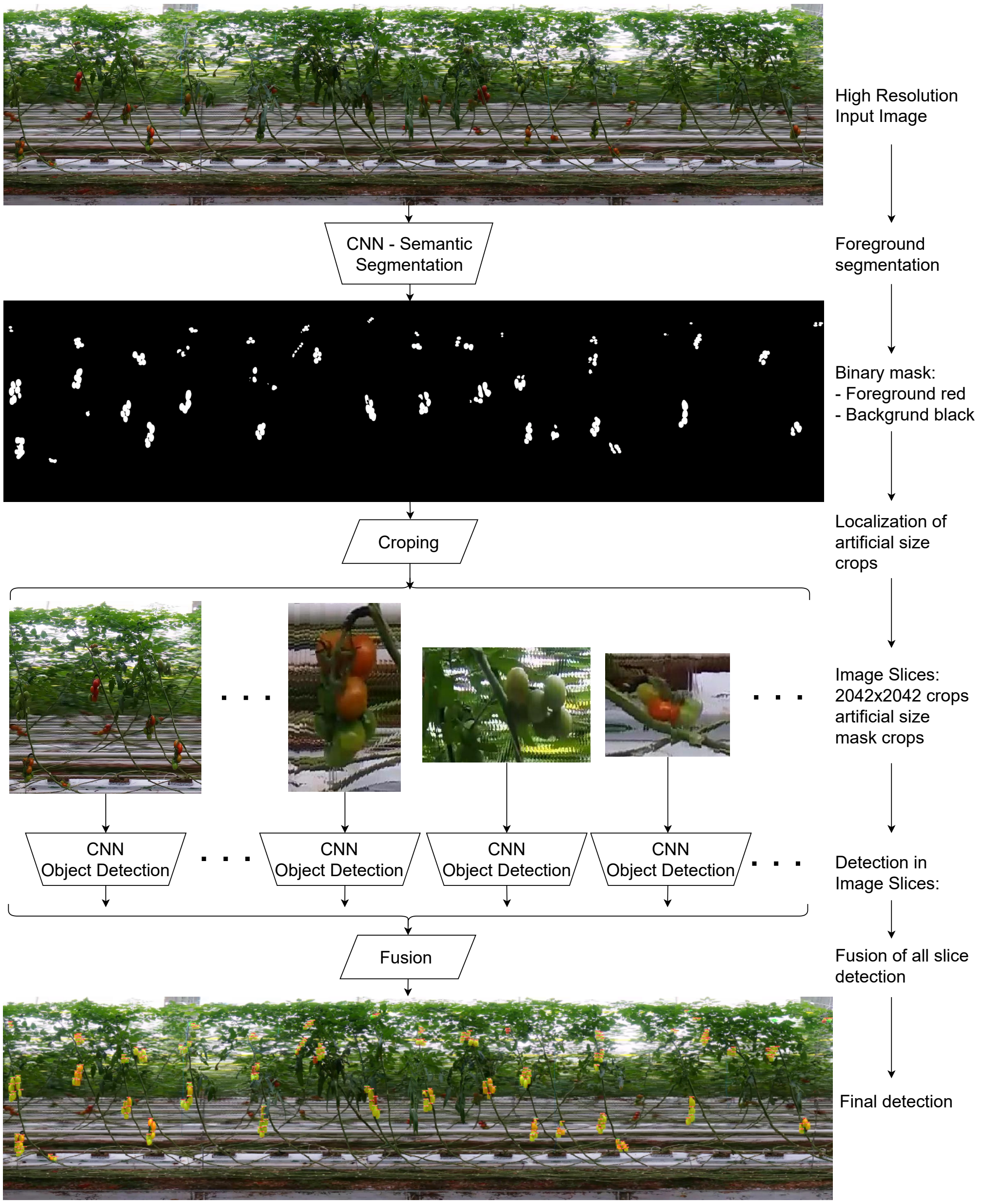

High Resolution Input Image
CNN - Semantic Segmentation
Foreground segmentation
Binary mask:
- Foreground red
- Backgrund black
Croping
Localization of artificial size crops
Image Slices: 2042x2042 crops artificial size mask crops
CNN Object Detection
CNN Object Detection
CNN Object Detection
CNN Object Detection
Detection in Image Slices:
Fusion
Fusion of all slice detection
Final detection

Fig. 4.2 Visualization of Artificial Size Slicing Aided Hyper Inference (ASSAHI) procedure proposed in this thesis.

A balanced combination of ASSAHI and SAHI techniques appears to be the best solution. Then ASSAHI assists with detailed small object detection while SAHI ensures comprehensive coverage of the entire input image with relatively large patches. Crop sizes in SAHI and ASSAHI parameter setups must always be adjusted to suit the specific dataset in use. The exploration of these techniques applied to the Tomato360 dataset is presented in the results part of this work, in section 5.4.

4.3.3 Image stitching

The merging of overlapping patches' predictions is relatively straightforward in the semantic segmentation task. Output masks can be aligned to their original positions within the larger image, and the values in overlapping areas might be simply averaged to generate the final output. This method generally yields satisfactory results, although it has been noted that output precision tends to diminish towards image borders [71, 42]. This can be mitigated by assigning greater weight to pixels near the center than to those near the border. To further increase the prediction accuracy, inference augmentation can be applied; for example, mirror all patches along all axes, and consider their average final output.

For object detection tasks, however, merging patch predictions is more complex. Objects on the border may be detected multiple times or even fragmented into two or more parts if they are positioned on a crop border. There are several potential strategies for merging or suppressing repeated predictions in object detection. Generally speaking, the process has two main parameters: the match metrics and the post-processing algorithm.

Match metrics identify potential detections for merging or suppression. The threshold value of match metrics is crucial; a higher value allows only highly similar predictions to be merged. The post-processing algorithm, on the other hand, dictates the sequence in which potential detections are processed and how the final detection instance is formulated.

There are two common match metrics:

1. Intersection over union (IOU) is a term used to describe the extent of overlap of two bounding boxes. The greater the region of overlap, the greater the IOU. The metric is defined as:

$$IOU = \frac{Area\ of\ intersection}{Area\ of\ union} \tag{4.1}$$

2. Intersection over a smaller area (IOS) is very similar to the IOU metric; only the area of the intersection is divided by the area of the smaller of the two boxes:

$$IOS = \frac{Area\ of\ intersection}{Area\ of\ smaller\ box} \tag{4.2}$$

In this thesis work, those variants of the post-processing algorithms were utilized:

1. Greedy non-maximum suppression (NMS),
2. Non-maximum merging (NMM),
3. Greedy non-maximum merging (GREEDYNMM).

The Greedy non-maximum suppression (NMS) algorithm is documented by the pseudocode 1. Only the greedy variant of non-maximum suppression is described because it is the standard in object detection. The algorithm chooses the predictions with the maximum confidence and suppresses all the other predictions overlapping with the selected predictions greater than a threshold of the match metrics.

Algorithm 1 Pseudocode of the greedy non-maximum suppression (NMS) algorithm, where P is the set of all predictions, S is the list of selected predictions with the highest confidence score, T is the list of any other prediction present in P being compared with S.

```
keep ← ∅                                         ▷ Initialize the final prediction list
while $P \neq \emptyset$ do                                   ▷ Repeat until $P$ is empty
    $S$ ← Select prediction with highest confidence score from $P$
    Remove $S$ from $P$
    Add $S$ to keep
    for each prediction $T$ in $P$ do
        Calculate overlap of $S$ with $T$ using IOU or IOS metrics
        if overlap exceeds the match metrics threshold then
            Remove $T$ from $P$
        end if
    end for
end while
return keep                         ▷ Return the list containing the filtered predictions
```

As can be observed from the algorithm 1, the whole filtering process depends on the match metrics threshold value. Therefore, a suitable threshold value selection is vital for the final performance. Although the NMS is a commonly adopted algorithm, it has several drawbacks. It is not ideal for object clusters because it leads to a misdetection if an object lies within the predefined overlap threshold.

From the nature of the problem of splitting the image, predicting the patches, and then stitching the resulting predictions; It is not unusual that the bounding boxes contain the cutoff objects. In this scenario, the NMS can only choose the best cut of the object, dumping the others. The Non-Maximum Merging (NMM) algorithm targets to solve this problem. It takes traditional NMS as the first step and matches all the detected boxes between themselves. Instead of keeping only the box with the higher confidence threshold and throwing away all the overlapping boxes, it merges them to form the new output box.

The greedy variant of non-maximum merging sorts the bboxes according to their confidence and removes the processed boxes from the list, similar to greedy non-maximum suppression, making the algorithm more efficient and effective

In a recent study [A1], the author of this thesis explores the influence of the parameters mentioned above on the final prediction precision. The implementation details of all described post-processing algorithms can be found in the open-source library SAHI: Slicing Aided Hyper Inference [2, 3], which is utilized in the experiments. It is a lightweight vision library for performing large-scale object detection and instance segmentation.

4.4 Validation metrics

There are several metrics to evaluate and understand the outputs correctly. The definitions of the ones commonly used in computer vision tasks and relevant to the experimental part of this work are in the sections below.

4.4.1 Metrics operating with the confusion matrix values

The confusion matrix (sometimes named as error matrix) is one of the standard methods to analyze the model performance in greater detail [65]. It is classically utilized in classification but can also be applied in object detection. Confusion matrix in the context of object detection compares the results of the classifier under test with trusted external judgments (dataset ground truth) using the terms true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). The terms positive and negative refer to the classifier's prediction, and the terms true and false refer to whether that prediction corresponds to the ground truth.

There are several metrics associated with the confusion matrix from which the most commonly used are explained below:

$$Precision = \frac{TP}{numDet} * 100\%, \tag{4.3}$$

$$Recall = \frac{TP}{numGT} * 100\%, \tag{4.4}$$

$$F1\text{-}score = \frac{2 * Precision * Recall}{Precision + Recall}, \tag{4.5}$$

$$False\ discovery\ rate = \frac{FP}{numDet} * 100\%, \tag{4.6}$$

$$False\ negative\ rate = \frac{FN}{numGT} * 100\%, \tag{4.7}$$

where numGT and numDet refer to the number of the ground true and detected objects, respectively.

The precision or positive predictive value gives the percentage of true positive samples from all the samples returned by the model. The recall value, sometimes named sensitivity or true positive rate, informs us about the percentage of the objects correctly retrieved by the model from all the objects present. The F1-score is the harmonic mean of precision and recall. The false discovery rate gives the fraction of incorrectly detected objects, while the false negative rate gives the fraction of objects left out by the model.

Aside from assessing the values stated above, the precision-recall curve is a useful diagnostic tool to judge model performance. The precision-Recall curve summarizes the trade-off between the true positive rate and the positive predictive value for a model using different probability thresholds. A high area under the curve represents both high recall and high precision, where high precision relates to a low false-positive rate, and high recall relates to a low false-negative rate. An average precision metric described in the following section calculates such area under a precision-recall curve.

4.4.2 Object detection evaluation according to COCO challenge

COCO challenge evaluation$^{1)}$ [51] is the one most used in current research papers. It calculates a 101-point interpolated mean average precision (mAP). Its primary metric is an mAP averaged over multiple intersection over union (IOU) values; specifically, it uses 10 IOU thresholds ranging from 0.5 to 0.95 with step 0.05. This metric emphasizes the precision of the bbox localization. Please refer to the original challenge evaluation for the details about these metrics.

In addition to the calculation of the mAP values, Derek's PR curve inspired by [36] provides a detailed breakdown of false positives. This plot is a series of precision-recall curves where each PR curve is guaranteed to be strictly higher than the previous as the evaluation setting becomes more permissive. The curves are as follows:

- C75: mAP at IOU=0.75 (AP at strict IOU), area under curve corresponds to mAP at IOU=0.75 metric.
- C50: mAP at IOU=0.50 (AP at PASCAL IOU), area under curve corresponds to mAP at IOU=0.50 metric.
- Loc: mAP at IOU=0.10 (localization errors ignored, but not duplicate detections). All remaining settings use IOU=0.1.
- Sim: mAP after supercategory false positives are removed. Specifically, any matches to objects with a different class label but that belong to the same supercategory do not count as either a FP (or a TP). Sim is computed by setting all objects in the same supercategory to have the same class label as the class in question and setting their ignore flag to 1. Note that when a single category is used, Sim result is identical to Loc.
- Oth: mAP after all class confusions are removed. Like the Sim, except now if a detection matches any other object, it is no longer a FP (or a TP). Oth is computed by setting all other objects to have the same class

---

1) The coco detection challenge evaluation is described here: `https://cocodataset.org/#detection-eval`

label as the class in question and setting their ignore flag to 1. Note that when a single category is used, Oth result is identical to Loc.

- BG: mAP after all background (and class confusion) FP samples are removed. BG is a step function for a single category that is 1 until max recall is reached then drops to 0 (the curve is smoother after averaging across categories).
- FN: PR after all remaining errors are removed (trivially mAP=1).

4.4.3 Evaluation metrics for segmentation

Different metrics are used to evaluate segmentation masks produced by a semantic segmentation model. The most commonly utilized ones are explained below. For more information about metrics in segmentation, please refer to [79, 80].

The easiest metric is an **Absolute Volume Difference** (AVD) [79], which counts the absolute number of voxels segmented differently by model than is stated in ground true segmentation. It provides a simple and direct way to measure the difference in segmented volumes, and while easy to understand and calculate, it may lack sensitivity to spatial relationships between objects, potentially leading to misleading interpretations.

Average distance, or **Average Hausdorff Distance** (AHD) [80] is the mean distance between the segmented object (S) and ground true segmentation (GT).

$$AHD(S, GT) = \max(d(S, GT), d(GT, S)) \tag{4.8}$$

where d(S,GT) is the directed Average Hausdorff distance that is given by

$$d(S, GT) = \frac{1}{N} \sum_{s \in S} \min_{gt \in GT} ||s - gt|| \tag{4.9}$$

where ||s−gt|| is some norm, e.g. Euclidean distance. AHD assesses the spatial relationship between segmented and ground true objects and is often used in medical imaging where precision matters. It is sensitive to spatial structures but may be sensitive to noise and outliers.

One of the most common metrics in medical imaging is the **Dice Coefficient** [21](4.10), which equals twice the number of elements common to both sets divided by the sum of the number of elements in each set. If $|X|$ and $|Y|$ are denoted as the cardinalities of the two sets, the Dice coefficient can be written as:

$$Dice = \frac{2|X \cap Y|}{|X| + |Y|} \tag{4.10}$$

This coefficient evaluates the overlap between predicted segmentation and ground truth, reflecting the quality of boundary segmentation. It is less sensitive to the absolute size of the segmented regions but may not be suitable in scenarios where false negatives and positives have different importance.

In summary, these metrics offer distinct perspectives on the segmentation quality, with each having its unique strengths and weaknesses. The choice of metric should align with the specific goals and constraints of the task, and using a combination of these metrics may provide a more comprehensive understanding of the model's performance.

5 Results

This section offers a comprehensive and detailed overview of the experiments conducted throughout this research and the corresponding results. The first part describes the creation of the Tomato360 dataset from the tomato greenhouse environment. This part provides an in-depth view of the various stages involved in dataset creation, including field data collection, data annotation, providing basic dataset statistics, and revealing dataset challenges.

The next two subsections study the role of attention mechanisms integrated into deep convolutional neural networks (CNN), with a particular emphasis on small object detection. The first section covers the abdominal organs and tumor segmentation domain, where attention gates are used to enhance the Unet architecture's accuracy, discerning tiny yet significant tumors. Continuing, the discussion delves into spatial attention for small object detection, particularly focusing on tomatoes within high-resolution images. This section includes an ablation study that evaluates architectural choices' effects on performance using the Tomato360 dataset.

The following part continues on the topic of fruit detection within the high-resolution Tomato360 dataset, employing novel techniques: Artificial Size Slicing Aided Fine-tuning (ASSAFT) and Hyper Inference (ASSAHI).

Lastly, the results section concludes with an analysis of practical applications within a tomato greenhouse environment, including whitefly counting, tomato counting, and estimating tomato crops. This segment connects all previously discussed methodologies and techniques and demonstrates their direct applications in real-world scenarios.

5.1 Creation of the Tomato360 Dataset

This section delineates the process of constructing a real-world, custom dataset named Tomato360 that forms one of the cornerstones of this dissertation. It signifies the practical application of the studied deep learning models and acts as a tangible example of transferring modern artificial intelligence technologies into functional, real-world applications.

Its primary intent is to capture high-resolution images of tomato rows in a greenhouse allowing precise counting of tomato fruits and enabling accurate current and future harvest predictions. Such predictions hold significant implications for the commercial aspects of greenhouse tomato cultivation, as they influence crucial factors such as delivery contracts and supply chain logistics. The models trained on the Tomato360 dataset then serve as a pivotal tool in understanding and improving the dynamics of this critical process.

Aside from this, the Tomato360 dataset serves a dual purpose within the context of this dissertation. Firstly, it provides a means to demonstrate the transfer of AI technologies from theory to practical implementation, offering insights into the challenges and potential solutions that such a process entails. Secondly, it offers a valuable opportunity to evaluate the performance and applicability of deep learning models outside the confines of standard, large-scale public datasets commonly used in the research community. The variability and unique characteristics of real-world data often present challenges not encapsulated in these publicly available datasets. As such, working with such custom data offers insights into the robustness and adaptability of the models, shedding light on their capabilities to generalize and perform in diverse scenarios. Therefore, the value of Tomato360 dataset creation extends beyond its immediate practical application in a Tomato greenhouse, providing a fertile ground for academic investigation and analysis.

The creation of the Tomato360 dataset is a product of a collaborative endeavor supported by the Technology Agency of the Czech Republic. This project brings together the efforts of the academic community and industry partners, specifically NWT, a technology-focused company, and Bezdinek, a farm with tomato greenhouses. This partnership aims to leverage the power of modern AI technologies, transforming them from abstract concepts into tangible tools that have a practical impact on real-world agricultural challenges. The Tomato360 dataset embodies the spirit of this initiative, serving as a pivotal resource in the project's quest to bring advanced deep learning techniques to the heart of precision agriculture.

5.1.1 Field Data Collection

As was mentioned in the literature review, section 3.5, the classical approach for fruit detection in already published research uses object detection algorithms applied on a video stream or simple images of plants. The practical usage of this approach in tomato greenhouses faces several limitations. The alleys between each tomato row are narrow, while the vertical range of tomato fruits on the plant is wide. Therefore the vertical field of view (FOV) of the used camera needs to be more than 90 degrees to capture the full plant height. Another limitation is based on the requirement for maximal counting accuracy. When applying image detection in single video frames or overlapping photos, some indexing or tracking algorithm must be used to prevent multiple counting of single tomato fruit. Due to the noise created during the capturing process - mainly vibrations of the trolley on rails, this quest is quite complicated and brings additional errors into final tomato counts.

To eliminate those limitations, a novel approach is proposed that uses 360 degrees camera (Ricoh Theta Z1 with high resolution 4K UHD (3840x2160) video) for image acquisition to cover the necessary wide field of view. Here comes the dataset name - Tomato360. The proposed counting solution works only with the source image from the camera. This approach has a significant consequence in that the process of data acquisition and data processing can be separated. Therefore, the automation of the data acquisition can be done by simple autonomous or semi-autonomous devices or by mounting the camera devices on available technical equipment in greenhouses such as, for instance, commonly used trolleys.

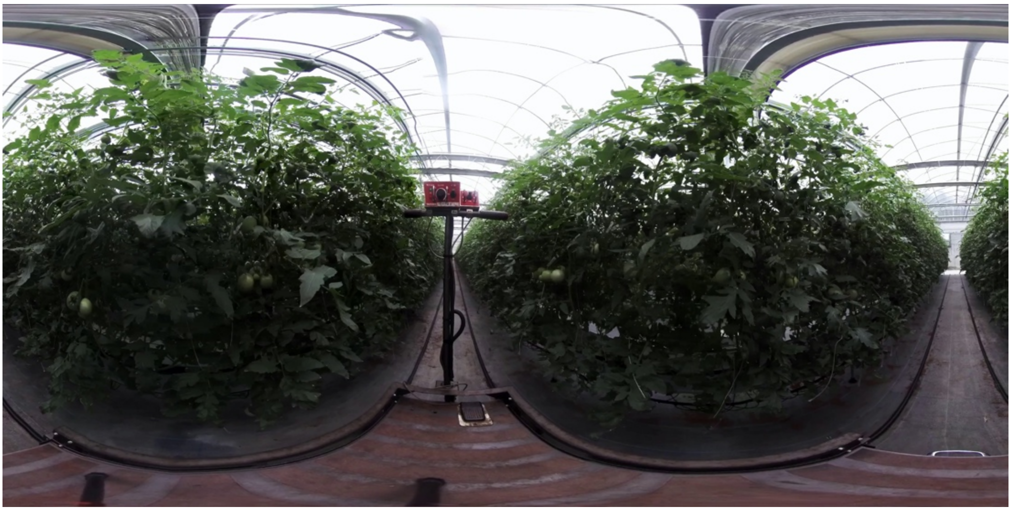


Fig. 5.1 A frame from a 360-degree camera before any preprocessing.

The source videos for experiments described below were acquired during several visits of the author and her colleagues from the project in the hydroponic greenhouse of Farma Bezdínek (Dolní Lutyně, Czech Republic) in 2020-2022. The videos were taken during the harvest period of the fully developed crops - cherry (10 - 12 g / fruit) and cocktail (35 - 45 g / fruit) tomatoes. The camera was mounted on a greenhouse trolley (Berkvens - Control Lift), allowing a maximum 5 km/ha speed.

5.1.2 Video converting

A 360-degree camera is utilized to produce a dual-view video stream, each offering a 180-degree perspective captured by fisheye lenses. The preliminary preprocessing step involves merging these two vantage points into a unified panorama, represented in equirectangular projection (see Fig. 5.1). This projection encompasses an expansive vertical view that approaches 180 degrees of the tomato row. Despite the potential distortion at the image's top and bottom regions, the central area remains conducive to efficient tomato detection. An added advantage of this method is simultaneously capturing both sides of the greenhouse aisle, recording two rows of tomato plants in a single frame.

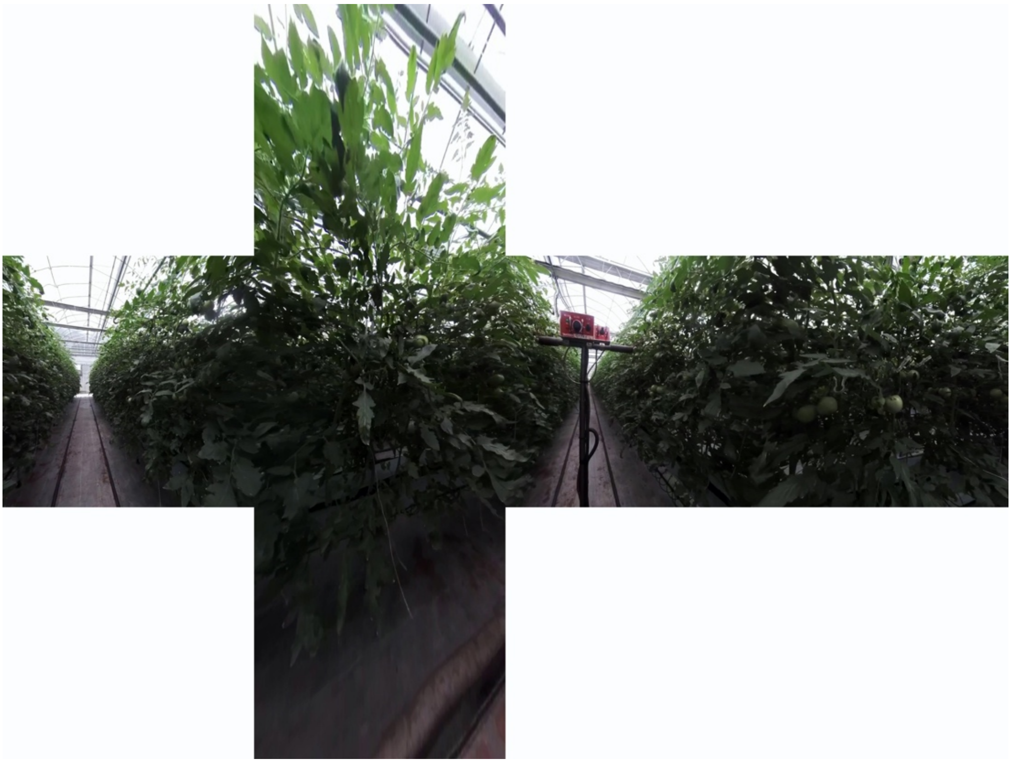


Fig. 5.2 Projection of a 360-degree camera frame into two planes, only the vertical part is utilized in the final image reconstruction.

Moving forward to image production, the process begins with decomposing the video into individual frames (see Fig. 5.2). The focus is then narrowed to the portion of the 360-degree equirectangular frame that directly captures the tomato row. A narrow vertical section is extracted from each frame, positioned immediately in front of the camera. These trimmed segments are then seamlessly connected to generate a comprehensive image of the entire row. Any deformations appearing at the top and bottom regions of the picture arising from the 180-degree view are rectified mathematically.

This method allows the transition from video or isolated video frames to an inclusive, wide-format picture, which can also be referred to as a panorama for the purposes of this document. This way, each fruit appears only once in the resulting image, eliminating any overlapping boundaries that might complicate

Tab. 5.1 Distribution of tomato ripeness categories in the Tomato360 dataset.

Category	Total Number	Mean per Image	Range per Image
Tomato	13815	234	71-591
Green	9932	168	47-323
Orange	1997	34	4-184
Red	1886	32	2-159

the counting process. This approach facilitates more effective analysis and application of computer vision techniques.

5.1.3 Data Annotation

To select a proper annotation tool, the following requirements for the annotation tool were defined based on the obtained data from the 360-degree video and its processing:

- the ability to process high image resolution of the composed images (image size over 350MB and 100 000 pixels wide),
- the tool's stability for dealing with the high frequency of tomato fruit occurrence in images,
- the possibility to annotate tomato fruits with polygons (to be used both for object detection and instance segmentation models),
- dividing the annotation task between multiple persons (working on different platforms).

Considering all those factors, the multi-platform desktop annotation tool Labelme [73] was selected as the annotation software. After completing the initial annotation process (by multiple persons from Tomas Bata University in Zlin), the segmentation masks were cross-validated by the leading person of the group of annotators - the author of this thesis. In the end, the annotations were exported to COCO format.

5.1.4 Dataset statistics

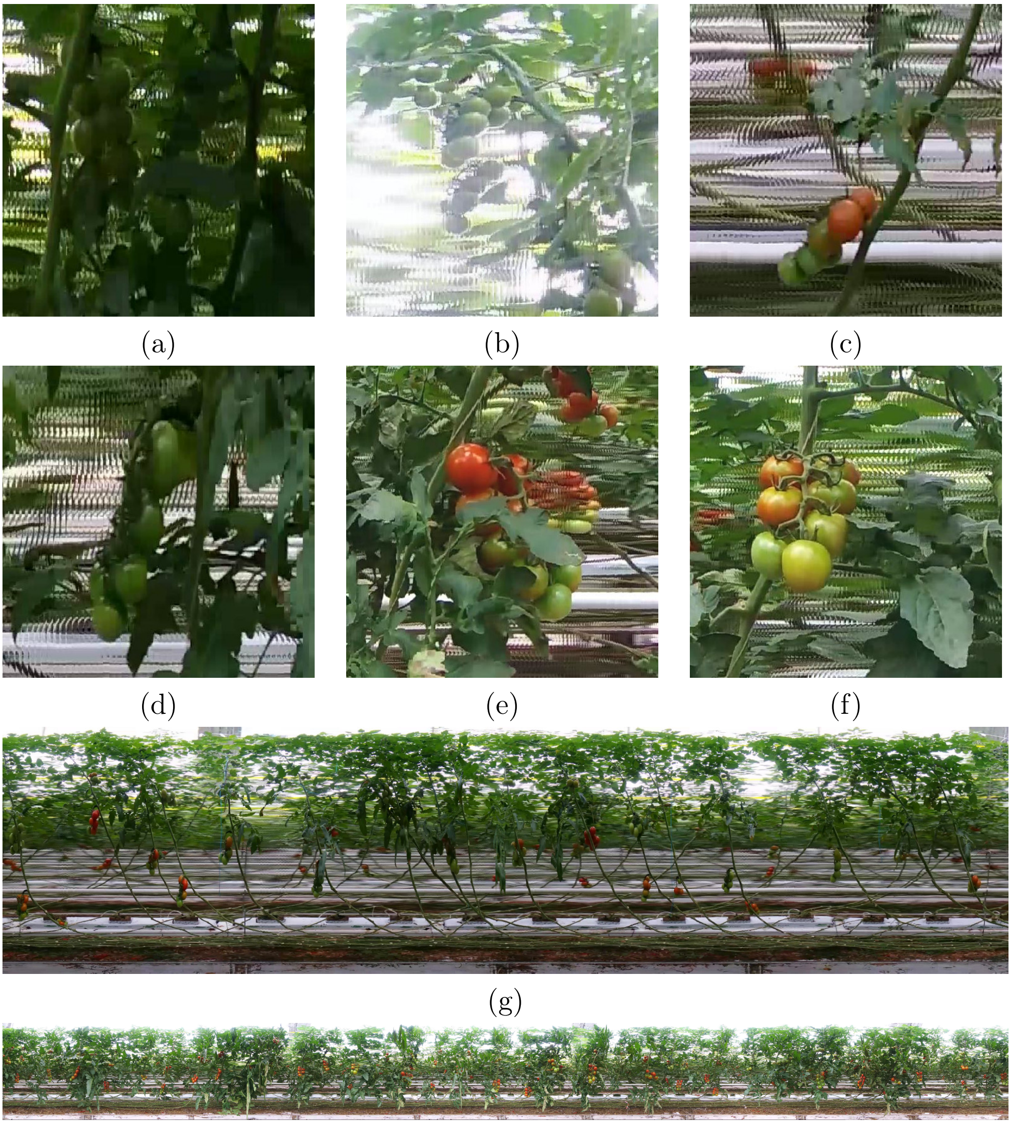
The compiled dataset comprises 58 images, each containing annotated tomatoes classified according to their stage of ripeness: green (unripened), orange (partially ripened), or red (fully ripened). In total, 1385 tomatoes were annotated. The dimensions of the images vary significantly, with widths spanning from 7500 to 20,000 pixels and heights falling between 1920 and 2048 pixels. On average, each image is marked with 234 objects, with a range between 71 and 591 objects.

When breaking down the total number of annotated tomatoes per category, there are 9932 green, 1997 orange, and 1886 red tomatoes. The distribution of the categories across images also varies substantially. For the green category, an average of approximately 168 tomatoes is seen per image, ranging from 47 to 323. For the orange category, the average drops to around 34 tomatoes per image, with a minimum of 4 and a maximum of 184. Similarly, the red category has an average of nearly 32 tomatoes per image, ranging from as few as 2 to as many as 159. Table 5.1 shows the described Tomato360 dataset statistic.

5.1.5 Dataset challenges

The Tomato360 dataset introduces several distinct challenges that make it unique. Fig. 5.3 showcases the difficulties described in the following text. Most striking is the high resolution of the images (Fig. 5.3 (g-f)), with widths reaching up to 20,000 pixels. Moreover, this high resolution comes with its own set of challenges, such as the noise introduced by the frame sampling from the video source. This noise visible especially in Fig. 5.3 (c, d, f), is mostly low-frequency and adds another layer of complexity to the dataset.

Besides, environmental factors specific to greenhouses present additional difficulties. One such issue is the high vertical range in the images, causing variable brightness levels. For instance, the top of an image could be overexposed by the light coming from a glass roof (Fig. 5.3 (b)), while the bottom portion appears



(a) (b) (c) (d) (e) (f) (g) (h)

Fig. 5.3 Example images documenting difficulties of Tomato360 dataset: (a) dark, (b) bright, (c) back row tomatoes, (d) fruit overlapping each other, (e) leaf occluding tomatoes, (f) indeterminacy of ripeness, (g-f) overview of whole wide images.

comparatively dark (Fig. 5.3 (a)). Since the accurate detection of the tomatoes at the bottom is crucial, the camera settings were adjusted to capture these parts more precisely, but this approach inherently impacts the overall image quality.

From an object detection standpoint, the Tomato360 dataset poses further complications. Tomatoes, which typically grow in clusters, often overlap and obscure each other (Fig. 5.3 (d)). Even more common is occlusion by leaves (Fig. 5.3 (e)). Additionally, unripe green tomatoes can be easily confused with the surrounding leaves, especially in variable image brightness. The greenhouse's row setup further exacerbates these issues. Tomatoes from the back row might sometimes appear in the images but should not be included in the count as they belong to a different row. The situation is nicely visible in Fig. 5.3 (c), where the bottom tomatoes are from the front row, but the top tomatoes come from a background row, image shown in Fig. 5.3 (g) captures a tomato row with cut bottom leaves, exposing the background row substantially. The front/back row exchange led to frequent discrepancies also among human annotators; such errors were the most common ones rectified during the annotation's quality control phase.

Lastly, there's a subjective element when assigning ripeness categories to the tomatoes (as an example, see Fig. 5.3 (e-f)). Judgments about whether a tomato is half-ripened, unripened, or fully ripened can significantly vary from one individual to another, adding another layer of complexity to the Tomato360 dataset.

5.2 Attention Gates for Tumor Segmentation

The paper [A2] represents the author's contribution to advancing the field of organ and tumor segmentation from computed tomography (CT) scans. The research presents a novel methodology, integrating attention mechanisms and deep supervision to enhance the precision of tumor segmentation. An exhaustive comparison of different CNN architectures for various organ-tumor segmentation tasks forms the core of this study. Besides, it visualizes the feature maps from trained CNN architectures to provide some insight into what is the focus of attention in the different parts of the model. In the following sections, an emphasis is placed on the attention mechanism applied in this work since it is relevant to the aim of this thesis. A brief description of the methodology is provided for the concept, and only the most salient results are discussed. For a comprehensive understanding of the research, it is recommended to refer to the original publication [A2].

5.2.1 Methodology

This section offers a short description of the proposed methodology. For detailed information, please seek the original paper [A2]. A publicly accessible implementation of the methodology using PyTorch is available at `github.com/tureckova/Abdomen-CT-Image-Segmentation`.

Because of memory restrictions, the model was trained on 3D image patches. Two different approaches were considered. **Full-resolution**, where the original resolutions of images are used for the training, and relatively small 3D patches are chosen randomly during training. And **low-resolution**, where the patient image is downsampled by a factor of two until the median shape of the resampled data has less than four times the voxels that can be processed as an input patch.

The popular architectural design of the fully convolutional encoder-decoder structure with skip connections was studied. This model is referred to as VNet. In addition to the original encoder-decoder network structure, attention gates [62] were added in the top two model levels and deep supervision [43]. Both extensions are described in detail in the original paper [A2]. A block diagram of the segmentation model with attention gates and deep supervision is in Fig. 5.4.

To minimize the problem of overfitting, a large variety of data augmentation techniques are applied, namely: random rotations, random scaling, random elastic deformations, gamma correction augmentation, and mirroring. All the augmentation techniques were applied on the fly during training. All models were trained

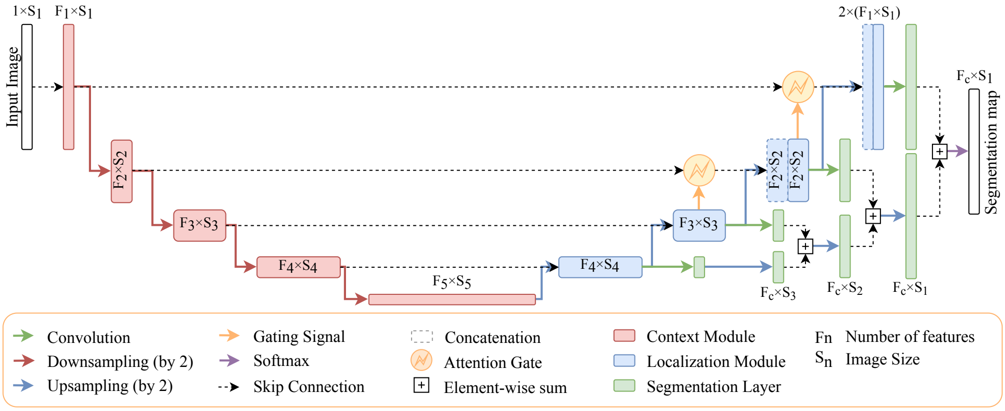


Fig. 5.4 A block diagram of an encoder-decoder segmentation model with attention gates and deep supervision.

with five-fold cross-validation using a combination of Dice and cross-entropy loss function (5.1). The cross-entropy loss speeds up the learning at the beginning of the training, while the Dice loss function helps to deal with the label unbalance, which is typical for medical image data.

$$L_{total} = L_{dice} + L_{crossEntropy} \tag{5.1}$$

According to the training, the inference of the final segmentation mask is also made patch-wise. The patches are overlapped by half the size of the patch, and voxels close to the center are weighted higher than those close to the border when aggregating predictions across patches. To further increase the stability, test time data augmentation by mirroring all patches along all axes was utilized.

5.2.2 Experimental Evaluation and Discussion

To demonstrate mainly the validity of incorporating an attention mechanism into network architecture, only part of the results that highlight the difference gained by the network architecture changes, namely attention gates and deep supervision were extracted from the original paper. The methodology is evaluated on the challenging abdominal CT segmentation problem - detection of cancerous

Tab. 5.2 Comparison of the proposed VNet-AG-DSV to the state-of-the-art network with similar parameters presented by [42]. All the models were trained on the same dataset, released by Medical Decathlon Challenge (MDC), and validated in five-fold cross-validation. A better score from the comparison of the two models is highlighted in bold.

	MDC Task03-Liver		MDC Task07-Pancreas	
Model	Liver	Tumor	Liver	Tumor
VNet [42] - Low Res.	**94.69**	47.01	79.45	49.65
VNet [42] - Full Res.	94.11	**61.74**	77.69	42.69
VNet [42] - Best model	95.43	61.82	79.30	52.12
VNet-AG-DSV - Low Res.	94.54	**54.72**	**79.58**	**52.43**
VNet-AG-DSV - Full Res.	**95.95**	57.65	**80.09**	**50.14**
VNet-AG-DSV - Assembly	**95,74**	**64,70**	**81,22**	**52,99**

tissue inside two different organs: pancreas and liver, both datasets published in Medical Decathlon Challenge 2018 [78], .

For each dataset, two model variants were trained to show the impact of the different model architecture choices. Moreover, assembly results from the respective full and low-resolution models were provided. The soft-max output maps from the full and the low-resolution model variant were averaged, and only then the final segmentation map is created.

The proposed network architecture is benchmarked against the winning submission of the Medical Decathlon Challenge (MDC), namely nnUnet [42]. Table 5.2 shows the mean Dice scores from five-fold cross-validation for the low and the full-resolution variants of models as well as the best model presented in either work. The winning results from nnUnet consist of the combined prediction from three different models (2D Unet, 3D Unet, and 3D Unet cascade) assembled together. Therefore, the results from the 3D Unet model, whose model architecture is close to our network, are compared to highlight the difference gained by the network architecture changes, namely attention gates and deep supervision.

The full- and low-resolution models with attention gates (VNet-AG-DSV) achieve higher Dice scores for both labels on the pancreas dataset. Of particular inter-

Tab. 5.3 Performance comparison.

	VNet [42]	VNet-AG-DSV
num. parameters [M]	29.6873	29.7383
train iteration* [ms]	297.2699	338.3336
eval iteration* [ms]	268.6558	299.3836

* measured as mean from 100 runs on GeForce GTX 1080 Ti

est is that the tumor Dice scores are substantially increased by three and seven points in low and full-resolution, respectively. In the case of the liver dataset, there is a significant improvement in the low-resolution case. Attention gates improve the liver-tumor Dice score by seven points while the liver segmentation precision is comparable. Any noticeable decrease in Dice score happens only in the liver-tumor class in the full-resolution case. Finally, if the best models presented in original paper [42] were compared with the best solution proposed in this work - i.e., assembly of full- and low-resolution models in this work and assembly from 2D Unet, 3D Unet, and 3D Unet cascade in case of nnUnet paper [42], our model with attention gates and deep supervision (VNet-AG-DSV) wins on both datasets, adding nearly three score points on the liver-tumor class and two points in pancreas label.

The performance of the model with and without the attention gates is quantitatively compared in Table 5.3. Both the number of parameters and the training and evaluation time increased just slightly, while the performance improvement was considerable. It should be mentioned that the decrease in the number of parameters in the work of [42] was compensated by training the network with larger patch size: 128 × 128 × 128 versus 96 × 128 × 128 for the Liver dataset and 96 × 160 × 128 versus 64 × 128 × 128 for the Pancreas dataset.

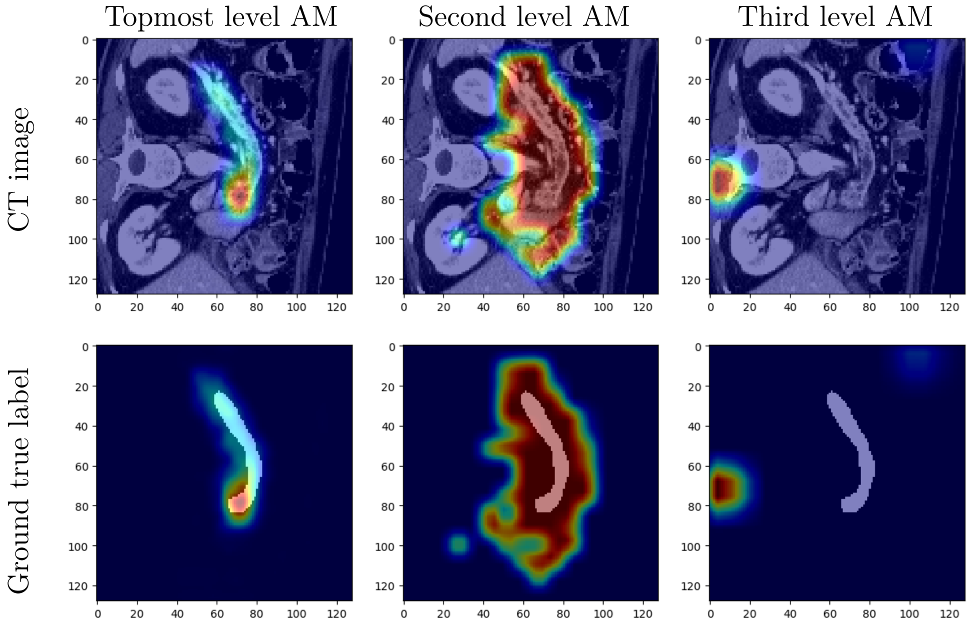


Fig. 5.5 Examples of attention maps (AM) obtained from attention gates in the three topmost levels of the low-resolution VNet (from left to right: full spatial resolution, downsampling of two and four).

5.2.3 Visualization of the Attentional Maps

The network design allows us to visualize meaningful activation maps and thus enables an exciting insight into the functionality of the convolutional network. The low-resolution VNet was chosen to study the attention coefficients generated at different levels of a network trained on the Medical Decathlon Pancreas dataset. Fig. 5.5 shows the attention coefficients obtained from three top network levels (working with full spatial resolution and downsampled two and four times). The attention gates provide a rough outline of the organs in the top two network levels but not in the lower spatial resolution cases. For this reason, in realized experiments, the AG was implemented only in two topmost levels to save the computation memory and handle larger image patches.

The attention coefficients obtained from two randomly chosen validation images from each dataset are visualized in Fig. 5.6. All visualized attention maps cor-

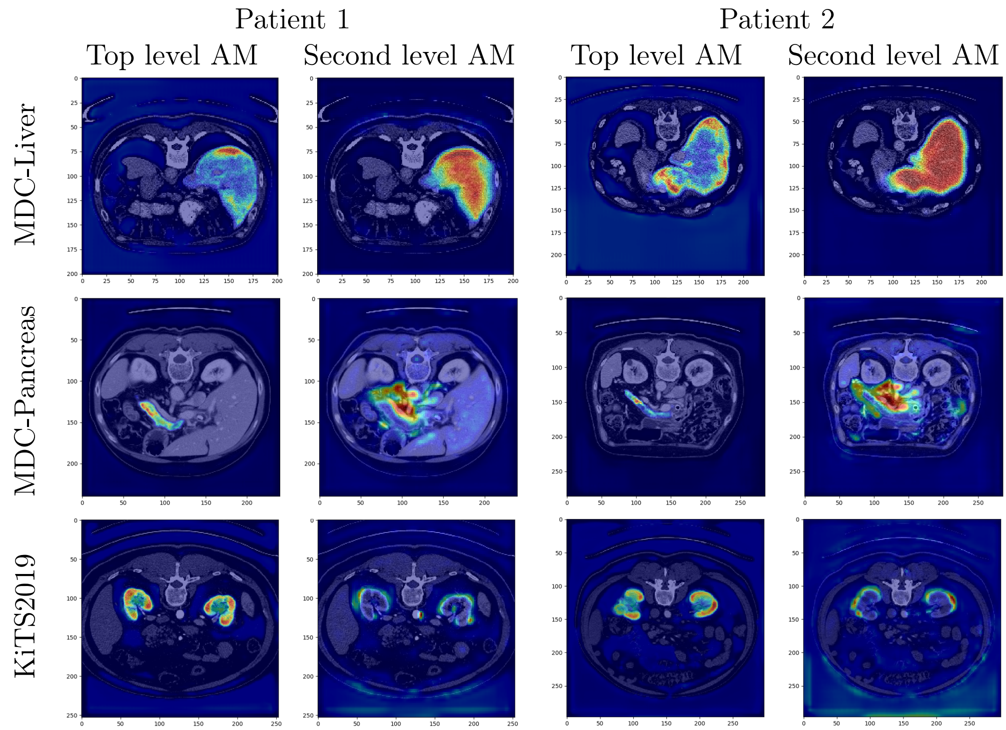


Fig. 5.6 Visualization of attention maps (AM) in low-resolution for VNet and two randomly chosen patient images from the validation set of each dataset. For each patient, the left picture shows the attention from the topmost layer (with the highest spatial resolution), and the right picture shows the attention from the second topmost layer.

relate with the organ of interest, which indicates that the attention mechanism is focusing on the areas of interest, i.e., it emphasizes the salient image regions and significant features relevant to organ segmentation. In the case of liver segmentation, the attention map correlates accurately with the organ on the second level while in the top-level, the attention seems to focus on the organ borders. In kidney and pancreas datasets, exactly the opposite behavior can be observed. The attention map from the top-level covers the organ, and the second-level attention map focuses on the borders and the close organ surroundings. This difference is possibly associated with the different target sizes as the liver is taking a substantially larger part of the image than the kidney or pancreas.

5.3 Spatial Attention for Tomato detection

This section brings the analysis of spatial attention mechanisms [100] incorporated into three common CNN models for object detection: Faster-RCNN [67], Tood [24] and RetinaNet [52]. It aims to explore the effectiveness of these attention mechanisms on the custom-made dataset Tomato360 which encompasses high-resolution images with relatively small tomato fruits to be detected. The assumption is that the attention mechanism might be able to incorporate effective contextual information from the image and level up the model's performance. The attention mechanism is incorporated into the feature extraction part of the network, extending a classical residual block common to all three object detection networks. For more details about the spatial attention mechanism, please refer to section 3.4.2.

The composed panorama images from the Tomato360 dataset have too high resolution to be processed at one pass on available hardware. Concurrently, the image can not be simply downsampled because the tomato fruit objects are tiny and would not be distinguishable in lower resolution. To overcome this issue, the images are sliced into several overlapping smaller patches, which are evaluated separately by the model; for more information, refer to methodology, section 4.3.2. The influence of the attention mechanism was evaluated on those patches. The final detection results in the original image after stitching are discussed in the next section 5.4. The following sections first describe the methodology aimed to bring a fair comparison of the different architectures and then bring an extended results comparison with discussion.

5.3.1 Methodology

Here, a brief outline of the methodology employed is presented. The implementation is written in Python, using PyTorch as a backend framework for building deep learning models. MMDetection [9], an open-source object detection toolbox, was employed for a consistent and fair comparison across all experiments.

Following paragraphs shortly describe the practicalities of data preprocessing, model training, inference setup, and evaluation.

First, the original wide images were randomly split into a train, validation, and test part, containing 38, 9, and 12 images, respectively. Two image-slicing pipelines were tested, first using fixed-size patches (marked as **SAHI**) and second utilizing artificial size slicing (marked as **ASSAHI**) as described in the methodology, section 4.3.2. The first conventional slicing (**SAHI**) follows the conclusion of paper [A1] by the author of this thesis and all the images were cut into patches of size $2042x2042$ pixels. During the cutting, the patches overlap by 0.2 times the patch size. This results in 447, 59, and 136 number of images in a train, validation, and test set respectively. The artificial size slicing cuts the input image according to object position and the final image counts produced in this setup were the following: 563, 280, and 562 for train, validation, and test set. After loading, the image was further resized to size $1024x1024$, normalized, and randomly flipped over the vertical axis; only then it is passed into the model.

Three common object detection models were studied: Faster-RCNN [67], Tood [24] and RetinaNet [52]. The MMDetection implementation of those models was utilized in this examination. Two variants of spatial attention mechanisms were tested. First, attention incorporating query content and relative position since this variant was recommended for the best cost/performance ratio in the original paper [100]. Secondly, the attention incorporating all four attention terms available was tested since it achieved the best performance in the original paper [100]. Aside from this, the original architectures were extended by deformable convolutions.

All the models use ResNet-50 [34] as a backbone and are initialized by pre-trained weights available in TorchVision, a library available in PyTorch. Standard MMDetection *schedule_ 1x* was applied for the training of all Faster-RCNN and Tood models. It utilizes 12 epochs, while epoch is meant as one run through all the training data. Stochastic Gradient Descent was used as an optimizer. The learning rate varies for different models but is always based on the original implementation taking into account the changing batch size. The batch size of

4 was preferred, but the GPU memory available was not sufficient for models implementing all attention terms; in those cases, the batch size of 2 was used instead. After each training epoch, a validation part of the dataset was evaluated. The learning rate was decreased by 10 on 8 and 11 epochs. The RetinaNet model was trained with a longer MMDetection *schedule_ 2x* with 24 epochs; since it was not able to train properly in 12 epochs due to its specific loss function, other parameters of the training stayed the same.

The inference was made on the same image patches as the training and the data preprocessing followed the pipeline applied during training except for the random flip which was omitted in the testing phase.

5.3.2 Experimental Evaluation and Discussion

This ablation study aims to show the impact of attention mechanism incorporated into three common object detection models: Faster-RCNN [67], Tood [24], and RetinaNet [52]. The mean average precision averaged over a series of different IOU (intersection over union) ratios from the COCO Detection challenge was utilized as a metric encompassing the overall model performance. As a second metric, a mean average precision using 0.5 IOU was displayed, too. The results were evaluated on image slices, to emphasize the model architecture choice influence that might be obscured during the final stitching and post-processing phase.

The Table 5.4 presents the results. In contrast to the result given in the original spatial attention paper [100], where authors achieved an increment of 5 points on a COCO test-dev, no substantial improvement caused by any of the attention mechanisms is present on the custom Tomato360 dataset. The only noticeable boost can be noted in the case of all four attention terms applied in the Tood model (in the table marked as 'Tood DCN att 1111'), but the training instability of this model variant and the fact that on the dataset with artificial-sized slices this model fails discourages any practical usability of this model variant. On the other hand, a steady increase can be associated with the use of deformable convolutions across all tested models and both dataset slicing variants.

Tab. 5.4 Architecture comparison (test on slices) - **tomato one class**

	SAHI		ASSAHI	
model	mAP	mAP50	mAP	mAP50
Faster-RCNN	0.441	0.818	0.491	0.869
Faster-RCNN DCN	0.445	0.819	0.499	0.873
Faster-RCNN att 0010	0.443	0.817	0.496	0.871
Faster-RCNN DCN att 0010	0.445	0.818	0.469	0.848
Faster-RCNN att 1111	0.441	0.820	0.494	0.872
Faster-RCNN DCN att 1111	0.441	0.819	0.493	0.872
Tood	0.429	0.828	0.509	0.887
Tood DCN	0.431	0.832	0.511	0.889
Tood DCN att 0010	0.431	0.831	0.501	0.881
Tood DCN att 1111	0.440	0.840	0.295	0.633
retinanet	0.305	0.664	0.383	0.749
retinanet DCN	0.312	0.667	0.446	0.812
retinanet att 0010	0.305	0.670	0.380	0.747
retinanet DCN att 0010	0.308	0.669	0.431	0.803
retinanet att 1111	0.307	0.670	0.390	0.755
retinanet DCN att 1111	0.306	0.656	0.389	0.753

Table 5.5 presents the computational complexity and model size, quantified as the number of FLoating point OPerations (FLOP) and the number of model parameters, respectively, for each evaluated architecture. For the Faster-RCNN model, the incorporation of Deformable Convolutional Networks (DCN) slightly decreases the FLOPs while marginally increasing the number of parameters. The addition of an attention mechanism increases both metrics. This pattern is also evident for the Tood and RetinaNet model, where the use of DCN and attention mechanisms elevates the complexity and model size. Overall, it is evident that advanced features like DCN and attention mechanisms add computational cost and complexity to the models, so it is crucial to evaluate their trade-offs carefully.

This is especially true in the Tomato360 dataset results, where the observed performance improvements of the attention mechanism are not particularly better. Therefore, the benefits may not justify the added complexities. Consequentially, the Faster-RCNN and Tood architectures, both equipped with deformable convolutions, were selected as the representative models for the subsequent investigations.

Tab. 5.5 FLOP in trillions (tera, 10^{12}) and number of parameters in millions (mega 10^{6}) for each of tested architecture, tested on the input image of size 1024x1024pixels.

model	FLOP [T]	Params [M]
Faster-RCNN	0.210	41.348
Faster-RCNN DCN	0.182	41.929
Faster-RCNN att 0010	0.211	44.312
Faster-RCNN DCN att 0010	0.211	44.893
Faster-RCNN att 1111	0.232	46.665
Faster-RCNN DCN att 1111	0.232	47.256
Tood	0.201	32.023
Tood DCN	0.172	32.599
Tood DCN att 0010	0.174	36.144
Tood DCN att 1111	0.194	38.507
retinanet	0.209	36.371
retinanet DCN	0.180	36.952
retinanet att 0010	0.209	39.875
retinanet DCN att 0010	0.209	40.456
retinanet att 1111	0.229	41.656
retinanet DCN att 1111	0.229	42.237

5.4 ASSAFT and ASSAHI in Tomato Detection

This section focuses on the evaluation of the newly proposed methodology named Artificial Size Slicing Aided Fine-tuning (ASSAFT) and Artificial Size Slicing Hyper Inference (ASSAHI), which extend upon the Slicing Aided Fine Tuning (SAFT) and Slicing Aided Hyper Inference (SAHI) concepts presented in [2]. Both novel principles are described in the methodology part of this work, namely sections 4.1.4 and 4.3.2. The methodology is evaluated on the custom Tomato360 dataset introduced in this work. This dataset is aptly suited for the successful implementation of ASSAFT and ASSAHI. First, the input data has a super high image resolution, effectively disabling any possibility of direct processing. Second, the tomatoes are growing in trusses, forming a distinct group of objects positioned relatively sparsely in the input image. A larger context is necessary to accurately locate these tomato trusses, particularly in the Tomato360 dataset, where the model is expected to differentiate between front and back-row tomatoes. Tomatoes in the back row come from a different row and should therefore not be included in the fruit count. For a comprehensive understanding of the properties of the Tomato360 dataset, please refer to section 5.1, or Fig. 5.3.

The needed larger context becomes available at two stages during the processing pipeline: initially when relatively large image patches are sampled and predicted by the object detection model, and again when the segmentation network, trained on these large patches, helps localize the tomato trusses. On the other hand, to be able to distinguish individual tomato fruits precisely, the model can benefit from the high details available in full resolution. The ASSAHI allows sampling such high-resolution image patches centered around each tomato truss localized by a semantic segmentation network. This results in more accurate detection, even when objects overlap or are hidden, for instance, by leaves.

This section presents results demonstrating the beneficial impact of the newly introduced Artificial Size Slicing Aided Fine-tuning (ASSAFT) and Artificial Size Slicing Hyper Inference (ASSAHI) techniques. These results are contrasted with those acquired from the Slicing Aided Fine Tuning (SAFT) and Slicing Aided Hyper Inference (SAHI) methods introduced in [2].

5.4.1 Methodology

Presented here is a brief outline of the employed methodology. The implementation is written in Python, using PyTorch as a backend framework for building deep learning models. MMDetection [9], an open-source object detection toolbox, was employed for a consistent and fair comparison across all experiments.

Following paragraphs shortly describe the practicalities of data preprocessing, model training, inference setup, and evaluation.

First, the original wide images were randomly split into a train, validation, and test part, containing 38, 9, and 12 images, respectively. Two image-slicing pipelines were tested, first using fixed-size patches (marked as **SAFT**) and second utilizing artificial size slicing (marked as **ASSAFT**) as described in the methodology, section 4.3.2. The first conventional slicing (**SAFT**) follows the conclusion of paper [A1] by the author of this thesis, and all the images were cut into patches of size $2042x2042$ pixels. During the cutting, the patches overlap by 0.2 times the patch size. This results in 447, 59, and 136 images in a train, validation, and test set, respectively. The artificial size slicing (**ASSAFT**) cuts the input image according to object position and the final image counts produced in this setup were the following: 563, 280, and 562 for train, validation, and test set. After loading, the image was further resized to size $1024x1024$, normalized, and randomly flipped over the vertical axis; only then it is passed into the model.

Two object detection models were studied: Faster-RCNN [67], Tood [24], both extended by deformable convolution. Those architecture choices were chosen in the ablation study presented in 5.3. The MMDetection [9] implementation of those models was utilized in this examination.

Both the models use ResNet-50 [34] as a backbone and are initialized by pretrained weights available in TorchVision, a library available in PyTorch. Standard MMDetection *schedule_1x* with 12 epochs and longer MDetection *schedule_2x* with 24 epochs was applied for the training of all Faster-RCNN and Tood models. The epoch is meant as one run through all the training data. Stochastic Gradient Descent was used as an optimizer. The learning rate varies for different models but is always based on the original implementation taking into account the changing batch size. The batch size of 4 was applied during training. After each training epoch, a validation part of the dataset was evaluated. The learning rate was decreased by 10 on the 8 and 11 epochs or on the 16 and 22 epochs for the longer training schema.

In order to determine the placement of artificially-sized patches in a test set of images, a mask identifying the positions of object groups within the original image is needed. To accomplish this, a basic FCN-Unet architecture [71] was trained using the MMSegmentation toolbox [17]. The process involved slicing images from the Tomato360 dataset into patches of size $2042x2042$ pixels and subsequently creating input segmentation maps from the object's instance masks, yielding 447, 59, and 136 images for the training, validation, and test sets, respectively. During the data loading phase, input images were normalized and downsampled to a resolution of $512x512$ pixels. The network training utilized the Cross-Entropy Loss criterion over 160000 epochs with the Stochastic Gradient Descent optimizer and an initial learning rate of 0.01, which polynomially decreased to 0.0001 during the training. Test set predictions were made patch-wise, consistent with the $2042x2042$ pixel patch size used in training.

The inference was made employing Slicing Aided Hyper Inference (SAHI) and Artificial Slicing Aided Hyper Inference (ASSAHI). The data loading pipeline was kept the same as was during the training phase, with the exclusion of the random flip, which was not included in the testing stage. Test set masks essential for artificial size slicing were generated using the trained FCN-Unet segmentation network. Predictions were also made using ground truth labels to generate a segmentation mask, eliminating an error included by the segmentation network and fully showcasing the capabilities of the ASSAHI technique.

5.4.2 Experimental Evaluation and Discussion

In this evaluation, the influence of newly proposed methodologies named Artificial Size Slicing Aided Fine-tuning (ASSAFT) and Artificial Size Slicing Hyper Inference (ASSAHI) is explored in the context of the Tomato360 dataset. The results are compared with the Slicing Aided Fine Tuning (SAFT) and Slicing Aided Hyper Inference (SAHI) concepts presented in [2].

Tab. 5.6 Foreground segmentation mean accuracy (mACC) and mean Intersection over union (mIOU results) for semantic segmentation model trained to obtain object groups positions needed for test image Artificial Size Slicing Aided Hyper Inference (ASSAHI).

model	mACC	mIOU	foreground ACC	foreground IOU
FCN-Unet	79.69	77.28	59.52	55.58

The first table, Tab. 5.6, displays the results of the semantic segmentation model, FCN-Unet [71], specifically tailored to obtain object groups' positions needed for ASSAHI. The model shows a mean accuracy (mACC) of 79.69%, a mean Intersection over Union (mIOU) of 77.28%, a foreground accuracy of 59.52%, and a foreground IOU of 55.58%. Although the results would not be very impressive in a standard semantic segmentation task, these results indicate a sufficient ability to correctly segment and identify foreground objects from the background.

The second table, Tab. 5.7, provides an extensive evaluation of the proposed ASSAFT and ASSAHI methodologies on the Tomato360 dataset. The metrics include mAP, mAP50, Precision, Recall, and F1-score. The results of two models (Faster-RCNN and Tood) trained using classic and artificial size slicing fine-tuning methodologies (SAFT versus ASSAFT) were compared. The test set results are produced by classical or artificial size-aided hyper inference (SAHI versus ASSAHI). It can be seen that the application of ASSAFT and ASSAHI brings a substantial increase in all presented metrics in comparison to SAFT and SAHI methodology. On the other hand, simple usage of ASSAHI with a model fine-tuned by SAFT does not bring better model performance. Presumably, the model was not prepared for the change in image resolution and therefore is not able to profit from it. This property might be very specific in the Tomato360 dataset, where the size of the object is not very variable, encompassing the great majority of small objects, less medium size objects, and zero large objects, as are the categories defined by COCO detection challenge[1]. Therefore the model was

---

[1] The COCO detection challenge (`https://cocodataset.org/`) divides the object into three categories according to the bbox size: small $< 32x32$ pixels, medium $> 32x32$ pixels and $< 96x96$ pixels and large $> 96x96$ pixels.

Tab. 5.7 Evaluation of proposed ASSAFT and ASSAHI methodologies on a Tomato360 dataset from Faster-RCNN DCN and Tood DCN models. Each row presents results from two different fine-tuning methods, SAFT and ASSAFT, combined with different inference techniques. SAFT and ASSAFT results are shown when combined with SAHI and ASSAHI inference methods respectively. For ASSAHI, the segmentation input is from either an FCN-Unet or a ground truth mask (gtmask).

model & method		mAP	mAP50	Prec.	Recall	F1-score
Faster-RCNN DCN						
SAFT	SAHI	0.398	0.674	0.67	0.75	0.71
	ASSAHI FCN-Unet	0.397	0.669	0.67	0.75	0.71
ASSAFT	ASSAHI FCN-Unet	0.431	0.713	0.71	0.80	0.75
	ASSAHI gtmask	0.455	0.753	0.78	0.85	0.82
Tood DCN						
SAFT	SAHI	0.367	0.645	0.65	0.72	0.68
	ASSAHI FCN-Unet	0.365	0.640	0.64	0.72	0.68
ASSAFT	ASSAHI FCN-Unet	0.432	0.703	0.70	0.77	0.74
	ASSAHI gtmask	0.461	0.747	0.75	0.82	0.78

SAFT - Slicing Aided Fine Tuning
ASSAFT - Artificial Size Slicing Aided Fine Tuning
SAHI - Slicing Aided Hyper Inference
ASSAHI - Artificial Size Slicing Aided Hyper Inference

not prepared to detect large objects present in high-resolution slices produced by ASSAHI. The ASSAFT eliminates this problem.

On top of that, the ASSAHI methodology is combined either with FCN-Unet segmentation mask prediction or with masks obtained from ground true annotation (gtmask) to showcase the full potential of ASSAHI, i.e., how much the results can be improved by utilizing a better semantic segmentation model. The difference between ASSAHI with FCN-Unet and gtmasks is noticeable and allows space for further improvement. The gap between FCN-Unet and gtmasks variants keeps even with the usage of longer training schema with 24 epochs, as presented in Tab. 5.8. From the data in the table, it can be concluded that utilizing a longer training schema is overly beneficial in all experiments, while the Tood architecture benefits from it more than the Faster-RCNN architecture.

Tab. 5.8 Comparison of performance metrics between Faster-RCNN DCN and Tood DCN model, using the ASSAHI method with FCN-Unet or ground truth mask (gtmask) as segmentation input. For each model, results are shown for two different training schemas (12 and 24 epochs).

model	epochs	mAP	mAP50	Precision	Recall	F1-score
Faster-RCNN DCN						
ASSAHI FCN-Unet	12	0.431	0.713	0.71	0.80	0.75
	24	0.453	0.732	0.73	0.80	0.77
ASSAHI gtmask	12	0.455	0.753	0.78	0.85	0.82
	24	0.455	0.753	0.78	0.85	0.82
Tood DCN						
ASSAHI FCN-Unet	12	0.432	0.703	0.70	0.77	0.74
	24	0.448	0.728	0.73	0.80	0.76
ASSAHI gtmask	12	0.461	0.747	0.75	0.82	0.78
	24	0.475	0.766	0.77	0.83	0.80

The Tood beats the Faster-RCNN in mAP metrics, obviously producing more precise bboxes. While the other metrics which utilize the 0.5 Intersection over Union (IOU), show the comparable performance of both model architectures.

The computation resources needed for both pipelines should be taken into account too. The downside of ASSAFT and ASSAHI methodology is mainly in the claims for resources, especially in the training phase. The usage of ASSAFT methodology results in approximately 5x more training data samples, which adequately lengthens the training process. Moreover, the semantic segmentation model must be trained, too. On the other hand, during the inference phase, the extra computing needs brought by ASSAHI in comparison to SAHI are not that substantial. The final prediction time of the whole test set by ASSAHI versus SAHI technique takes 60.9 seconds versus 128.2 seconds; the calculation time was measured on GeForce GTX 1080 Ti GPU. This difference is not substantial, especially if the extensive data preprocessing needed in Tomato360 image data production is taken into account. Therefore the counting does not opt for real-time processing, and consequently, one minute difference in processing 12 images would not make any substantial difference in a practical implementation of this method.

While evaluating the ASSAFT and ASSAHI methodologies, it is important also to consider the computational resources required. These methods, despite their advantages, do present a substantial demand for resources, particularly during the training phase. The application of the ASSAFT method increases the number of training data samples by approximately five times, leading to a significant extension of the training process. Additionally, the semantic segmentation model must be trained, too.

During the inference phase, however, the additional computational needs of ASSAHI, when compared to SAHI, are relatively minimal. To illustrate, the total prediction time for a complete test set using ASSAHI is 60.9 seconds, while SAHI takes 128.2 seconds. These measurements were taken on a GeForce GTX 1080 Ti GPU. This difference in processing time is not particularly significant, particularly when one considers the extensive data preprocessing required for the production of Tomato360 image data. Given that real-time processing is either way not possible in this task, a one-minute difference in processing 12 images is unlikely to have a major impact on the practical application of this method. The increase in training time is paid for by the increased object detection precision.

In conclusion, these evaluations underscore the effectiveness of the proposed ASSAFT and ASSAHI methodologies for object detection on the Tomato360 dataset. The results highlight the importance of appropriate segmentation mask input for the correct ASSAHI function and show the benefits of increased training time.

5.5 Practical Applications in Tomato Greenhouse

This section moves beyond theoretical concepts to focus on tangible, real-world applications within a tomato farming environment. The first application addresses the critical task of detecting whiteflies on yellow sticky tags, which are commonly employed for pest monitoring in greenhouses. The effectiveness of the proposed solution in this context provides a meaningful evaluation of the technology's practical use. The second application delves into tomato detection and counting, utilizing the Tomato360 dataset and newly introduced techniques ASSAFT and ASSAHI for a successful tomato yield prediction. Together, these two core areas represent significant strides in addressing the unique challenges of tomato farming, bridging the gap between theoretical research and on-the-ground implementation.

5.5.1 Whitefly Detection

The primary objectives of this project focus on the automation and evaluation of a system designed for scouting yellow sticky traps (YST) for whiteflies in a tomato greenhouse. The main goal is to automate this process, replacing labor-intensive manual scouting with a more efficient and automated system. The comprehensive assessment of the system's performance is performed in work [A6] coauthored by the author of this thesis. This analysis is crucial to understanding its potential for practical use and real-world applicability. Here the results showcase a successful practical application of Slicing Aided Fine Tuning (SAFT) and Slicing Aided Hyper Inference (SAHI).

The pictures of yellow sticky traps (YST) were taken inside the tomato production greenhouse. Used pictures were collected using a mobile device to be easily replicable and suitable for future fast processing in a real environment. The whiteflies (Aleyrodidae) were manually marked by bounding boxes and validated by a professional phytopathologist. The created dataset was then used to train a deep convolutional network model for object detection. The state-of-the-art Tood architecture was utilized. The model training and inference were enhanced by a slicing that utilizes cutting of the original image to maintain full image resolution in the training (SAFT) and prediction process (SAHI). The utilized patch size was 512x512 pixels; during the cutting, the patches overlap by 0.2 times the patch size. More details can be found in the original paper [A6].

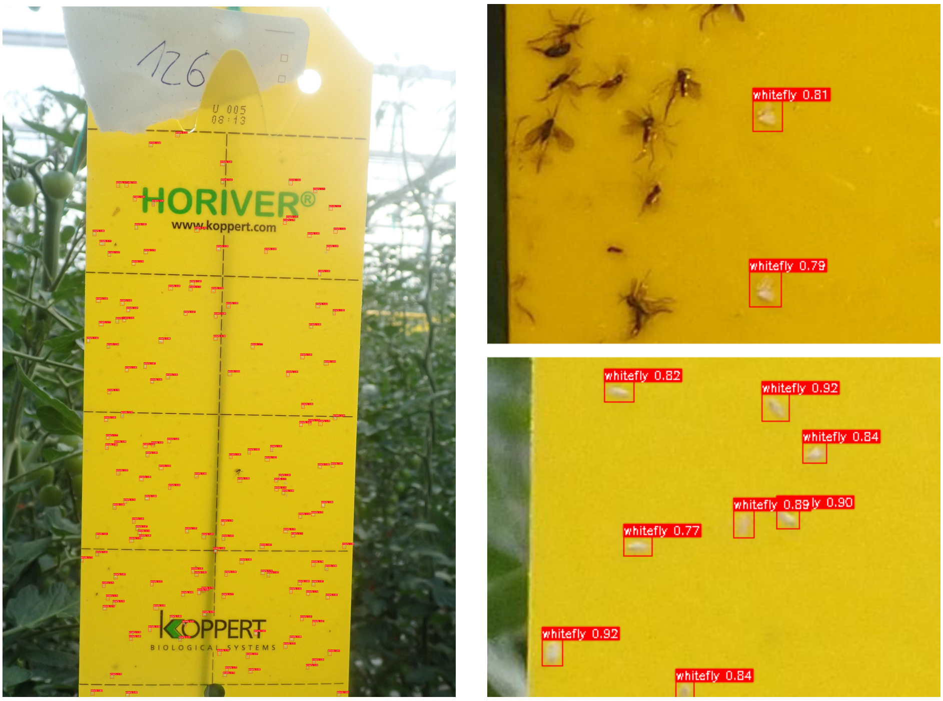


Fig. 5.7 On the left side is the example of the full input image with bounding box detections (red boxes) marking present whiteflies. On the right side, there are the zoomed cutouts with bounding box detections marking class labels and the model's detection confidence.

The final model achieves the F1-score of 0.82. The F1-score is defined as the harmonic mean of precision and recall. The key to a correct model function is to minimize the glue reflection on the YST since the reflected spots might be incorrectly classified as whitefly. If this precaution is complied with, the model gives stable output. The counting results from the greenhouse employees with the numbers from the phytopathologist (on a smaller dataset) were also compared to observe the human error rate. Human labor achieves the F1-score of 0.81.

This project has achieved its primary objectives, resulting in several significant contributions. By implementing a deep convolutional network model for object detection in scouting yellow sticky traps (YST) for whiteflies, a previously labor-intensive task is successfully automated. This automated system replaces repetitive human work and operates 24/7 without any degradation in precision, demonstrating the potential for wide-scale applicability in tomato greenhouses. Notably, the model's precision is competitive with human work, as evidenced by the comparable F1-scores achieved. Moreover, the model is capable of providing accurate counts even during high infestation periods, offering a dependable tool for tracking and managing whitefly populations.

Hence, the study not only offers promising insights into the practical application of Slicing Aided Fine Tuning (SAFT) and Slicing Aided Hyper Inference (SAHI) but also presents a robust solution for real-world challenges in pest management. These achievements underscore the relevance and potential of using Tood architecture for practical applications, as detailed in the coauthored work [A6].

5.5.2 Tomato Detection and Counting

This section introduces the final solution proposed for tomato fruit detection in wide high-resolution images captured within a tomato greenhouse environment. The solution involves a multi-faceted approach that merges various cutting-edge techniques. At its core, it leverages the Tood object detection model [24], which is enhanced with deconvolution extensions for an extra layer of detail and precise object localization.

Crucially, the training phase of this model incorporates a novel method Artificial Size Slicing Fine Tuning (ASSAFT), which has been specifically developed by the author of this thesis to maximize the model's effectiveness. Further, the model's predictions are generated through an equally innovative technique: Artificial Size Slicing Hyper Inference (ASSAHI).

This comprehensive framework, encompassing the processing pipeline, model training, and architecture, has undergone a rigorous ablation study to test other possible solutions. For a more detailed examination of these elements, please refer to sections 5.3 and 5.4, where each choice was evaluated and discussed in depth.

Here, both qualitative and quantitative analyses are carried out to evaluate the solution's capability for practical applications. The primary metrics used for the quantitative analysis are precision, recall, and the F1-score, which is defined as a harmonic mean of precision and recall. All these metrics were calculated with an Intersection over Union (IOU) threshold of 0.5 to determine true positive detections.

To evaluate the system's performance on the entire dataset, the data were divided into five folds, with each fold having a unique model trained on it. Each model then made predictions only on its testing portion of the dataset, and the final results were aggregated from all these five models' test predictions. The resulting scores were solid, with a **precision** of **0.85**, a **recall** of **0.93**, and an **F1-score** of **0.89**.

The results were also examined qualitatively, revealing an increased error rate in the top sections of the plants/images. The tomato fruits present in the top parts of the plant are usually very small and immature/green. Detection of such small, green objects among leaves is even more complicated by the fact that the top part of the images tends to be overly bright and suffers from image reconstruction artifacts. Additionally, the quality of the annotations in these upper sections was inconsistent; some human annotators overlooked these small, top-level detections, leading to some detections being incorrectly marked as false positives during the evaluation process.

However, detecting fruit in the top section of the plant is less crucial in practical terms, as greenhouse managers primarily aim to predict the crop yield for the forthcoming one or two weeks. To assess accuracy in different sections of the images, detections can be filtered based on their height percentile. As illustrated in Fig. 5.8, detection accuracy increases notably in the top 20 percentile.

Regardless, the exclusion of the top 20 percentile of detections has an insignificant impact on the overall precision, recall, and F1-score, only confirming the unsubstantial role of those image parts. Nevertheless, this analysis reveals some confusion originating from these upper image sections. As a result, a further

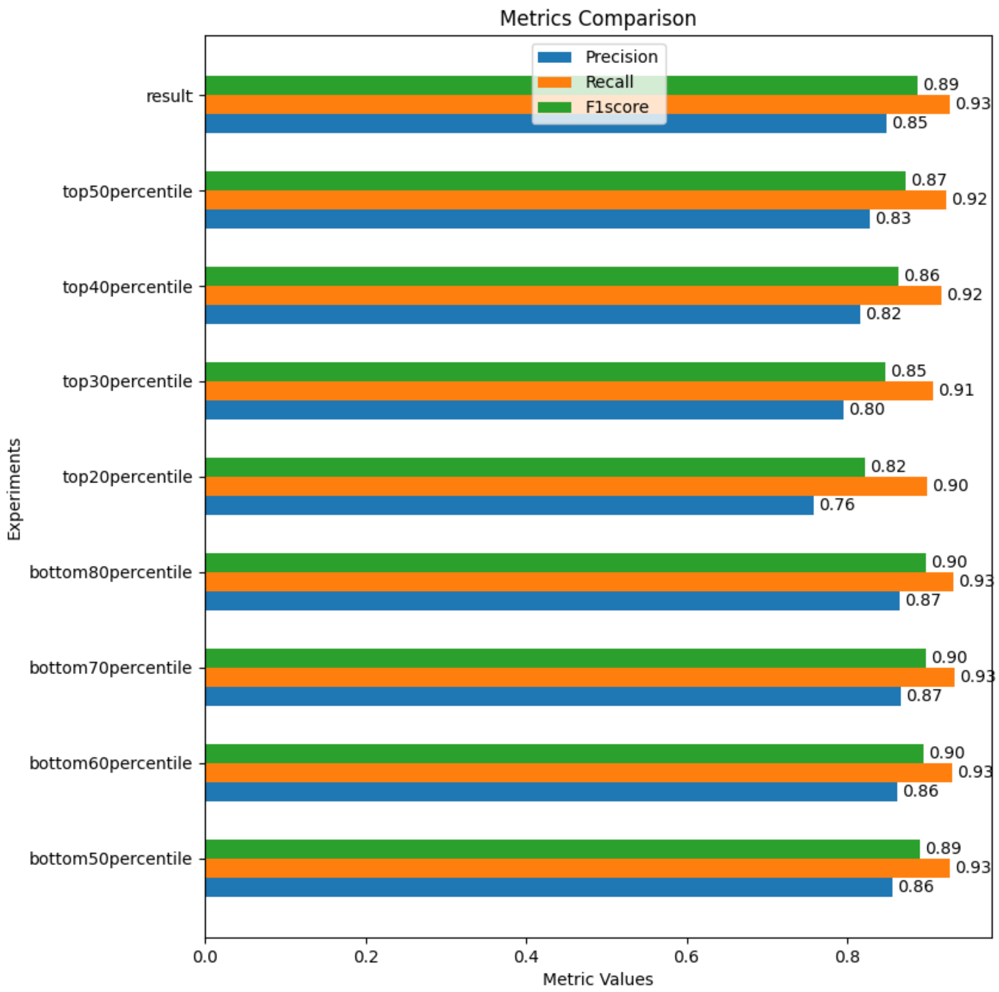


Fig. 5.8 Precision, recall, and F1-score comparison for different percentiles of tomato vertical position.

review of the input data could be beneficial, aiming to standardize annotations in the top sections of the images and possibly improve the training process.

5.5.3 Estimating Yield of Tomato Crops

The Tomato360 dataset was specifically designed to aid in predicting future yields of tomato crops. To demonstrate the real-world utility of the methodology

Tab. 5.9 Comparison of crop estimates made by an agronomist and by the proposed model.

harvest	current day		in 7 days	
row number	27	29	27	29
actual crop yield [kg]	30.40	38.20	92.20	99.60
agronomist's estimate [kg]	50.00	50.00	110.00	110.00
agronomist's error [%]	-64.50	-30.89	19.35	10.44
model's estimate [kg]	29.07	30.45	90.19	98.20
model's error [%]	4.38	20.29	2.18	1.41

proposed in this thesis, two rows of Belioso tomato species were captured in October 2022 at Bezdinek greenhouse. An agronomist was asked to provide an estimate of the crop yield for the day of imaging as well as for one week thereafter. The actual crop yield in kilograms was then recorded both on the day of the imaging and seven days later.

To convert the number of fruits into kilograms, an average tomato weight of 38.5g was used. The model's predictions for the current day's yield were calculated by multiplying the number of fully ripened tomatoes by this average weight. The predictions for the following week incorporated semi-ripened and immature tomatoes as well. The final results, as shown in Table 5.9, demonstrate that the model consistently provided more accurate predictions than the agronomist.

Historical comparisons of estimated yields versus actual harvests in the Bezdinek greenhouse have shown that the precision of the agronomist's estimates can vary considerably. Overestimations or underestimations by more than 20% have been particularly economically damaging. Thus, the model's F1-score of 0.89, coupled with the specific example of real applicability presented here, indicates that the detection model can serve as a reliable basis for predicting tomato crop yields in a real-world setting.

6 Discussion

The effectiveness of deep learning methodologies is fundamentally dependent on a robust deep learning pipeline. Given that deep learning models offer data-centric solutions, how data is processed critically determines the ultimate success of the model's implementation. This concept is echoed in a study on medical image segmentation by [42] and further examined in a research paper authored by this thesis's author [A2]. This study delves into the extensive incorporation of attention mechanisms into the basic model architecture and suggests a tangible benefit to including attention mechanisms in the detection of abdominal tumors, evidenced by a consistent increase in the Dice coefficient.

The research then extends to investigate the impact of attention mechanisms on the custom-made, real-world Tomato360 dataset. In this study, a comprehensive examination of various architectural decisions was conducted, including deconvolution techniques and different spatial attention mechanisms in object detection. These variations were integrated into three well-established object detection models and tested on the newly created Tomato360 dataset, produced in collaboration with NWT, a tech-focused company, and Bezdinek, a tomato farm. Surprisingly, these architectural choices did not significantly affect the success of the final object detection.

In contrast, the influence of the deep learning pipeline was substantiated further through the implementation of the newly proposed methods: Artificial Size Slicing Aided Fine Tuning (ASSAFT) and Artificial Size Slicing Aided Hyper Inference (ASSAHI). These methods are specifically designed to process high-resolution images containing small objects targeted for detection. Tests of ASSAFT and ASSAHI on the Tomato360 dataset demonstrated a considerable enhancement in the success of object detection, further affirming the pivotal role of data handling in the practical deployment of deep learning techniques.

Importantly, the use of a custom, real-world dataset demonstrates the process of transitioning deep learning techniques into practical applications. This work documents the intricate and non-linear process of creating a real-world dataset, providing solutions to specific challenges encountered in new contexts. These challenges, typically not found in the standard large-scale datasets used by the computer vision community, necessitated a somewhat different approach. This dissertation offers a comprehensive overview of such a procedure, providing insight into the complexities that arise during the practical implementation of deep learning techniques to solve real-world problems.

The applicability of the proposed solution was confirmed in two showcase studies realized at Bezdinek greenhouse. The first focuses on whitefly detection at Yellow sticky tags, commonly used for pest monitoring. The realized system achieves a comparable F1-score as a human operator. Moreover, the model is capable of providing accurate counts even during high infestation periods, offering a dependable tool for tracking and managing whitefly populations.

The second application deals with large-scale tomato detection and counting. The results demonstrate that the model consistently provided similar or more accurate predictions than the agronomist. Historical comparisons of estimated yields versus actual harvests in the Bezdinek greenhouse have shown that the precision of the agronomist's estimates can vary considerably. Especially economically damaging are estimates with errors higher than +-20%. Thus, the model's F1-score of 0.89 indicates that the detection model can serve as a reliable basis for predicting tomato crop yields in a real-world setting.

6.1 Fulfillment of the Doctoral Thesis Aims

This section summarizes the efforts undertaken to achieve the objectives of this dissertation, which were initially defined as follows:

1. **Appraise the current state of the research area:** Specifically, deep learning methods applied in computer vision with a particular focus on small object detection and segmentation in high-resolution images.

The area of deep learning methods applied in computer vision is a rapidly evolving research field marked by frequent incremental advancements rather than groundbreaking discoveries. Despite this, diligent efforts were made to stay abreast of the latest studies published in reputable journals and conferences, with only the most relevant ones to the dissertation topic being selected. A thorough overview of these selected studies can be found in the Literature Review, sections 3.1 - 3.6.

2. **Develop and curate a custom dataset in a tomato greenhouse:** The creation of a custom, real-world dataset aims to demonstrate the transfer of AI technologies from theory to practical implementation. This involves acquiring, collecting, and labeling high-resolution images that capture the challenges specific to this domain.

   In partnership with NWT, a tech-oriented company, and Bezdinek, a tomato farming enterprise, the Tomato360 dataset was created. The dataset is introduced in the author's paper [A1]. The process of data acquisition, image production, and labeling is detailed in the result section 5.1. This section also includes basic dataset statistics and identifies major challenges.

3. **Investigate and compare the effectiveness of attention mechanisms:** Explore possibilities of incorporating attention mechanisms into different convolutional neural network (CNN) architectures. Compare their performance in terms of accuracy and computational efficiency.

   Sections 5.2 and 5.3 present two case studies of attention mechanism integration into deep CNNs. The first study, published in the impacted journal [A2], documents a successful implementation of attention gates in medical image segmentation. In the second case, an ablation study of spatial attention incorporated into different object detection models was conducted and tested on the Tomato360 dataset.

4. **Develop an enhanced deep learning pipeline:** Design and develop a novel processing pipeline tailored to handle the challenging task of small object detection in high-resolution images.

The effectiveness of deep learning methodologies depends significantly on a robust deep learning pipeline. All the fundamental components of such a pipeline are outlined in the Methodology part, sections 4.1-4.4. The practical knowledge needed to establish such a comprehensive methodology overview was gathered over the whole doctoral studies of the author, and its correctness is confirmed in a successful computer vision application in different application fields published by the author of this thesis: [A1, A2, A4, A5, A6, A7, A8, A9]. The novel techniques, Artificial Size Slicing Aided Fine Tuning (ASSAFT) and Artificial Size Slicing Aided Hyper Inference (ASSAHI), are introduced in sections 4.1.4 and 4.3.2, respectively. Those techniques are specifically tailored to handle high-resolution images with small objects within to be detected.

5. **Evaluate the proposed pipeline on the custom dataset:** Apply and test the developed processing pipeline on the custom dataset from the tomato greenhouse. Measure its performance against existing standard techniques used for small object detection. Assess and compare the proposed pipeline's accuracy, robustness, and efficiency.

   The newly proposed methods of Artificial Size Slicing Aided Fine Tuning (ASSAFT) and Artificial Size Slicing Aided Hyper Inference (ASSAHI) are successfully applied to a custom-made Tomato360 dataset. The methodologies were tested using two different object detection model architectures. The effects of each methodology component were analyzed in the results, in section 5.4, along with the demands on time and computational resources.

6. **Analyze the impact and practicality of the proposed methods:** Conduct a comprehensive analysis to understand the impact of incorporating attention mechanisms and the newly developed processing pipeline on small object detection in high-resolution images. Evaluate their practicality in real-world scenarios, considering factors such as computational requirements, scalability, and generalizability.

   This dissertation thoroughly examines the implications and feasibility of incorporating attention mechanisms and the newly developed processing pipeline for small object detection in high-resolution images. Despite promising outcomes from the application of attention mechanisms in medical image data (Section 5.2), a similar mechanism did not yield significant improvements for the Tomato360 dataset as discussed in section 5.3. In contrast, the novel image-slicing techniques - ASSAFT and ASSAHI, showcased substantial improvements in the final tomato detection performance (See the section 5.5.2). This reinforces the importance of a methodically constructed deep learning pipeline as a critical determinant of successful real-world applications of deep learning models. Moreover, the final proposed solution proved to be a reliable basis for predicting tomato crop yields in a real-world setting, as is documented in section 5.5.3.

7 Impact of Work on Science and Practice

Deep learning techniques, and especially deep convolutional neural networks, occupy the field of computer vision nowadays, outperforming other techniques substantially. Despite the success of deep CNN techniques, there are difficulties inherent to their applicability. First, large datasets are needed for the successful training of deep CNN models, which requires a considerable amount of resources. Aside from problems due to the cost of acquisition, labeling, and data anonymization techniques, the methodology of processing and dealing with the data strongly influence the final method's success rate. This work seeks to establish an overview of the current standard techniques and best practices to set up the logical, consistent pipeline applicable to different computer vision tasks.

The practical knowledge needed to establish such a comprehensive methodology overview was gathered over the whole doctoral studies, and its correctness is confirmed in successful computer vision applications in different fields published by the author of this thesis: [A1, A2, A4, A5, A6, A7, A8, A9]. This thesis, moreover, documents the application of deep convolutional neural networks in a practical, real-world application from a commercial farming environment. From the first problem definition to a final solution.

The assignment of tomato fruit counting appears repeatedly throughout this work and creates the connection between theoretical research and practical application. In this practical example, this work documents the complex and non-linear process of creating a real-world dataset, shedding light on the unique challenges that arise in specific application contexts and proposing solutions to address them. Those challenges are not included in common large-scale datasets utilized by the computer vision community and therefore needed a slightly different approach. By providing a detailed account of how these challenges were identified and addressed, this dissertation underscores the need for flexibility and innovation in the application of deep learning techniques to real-world problems. Furthermore, it highlights the potential value that custom datasets can bring in furthering our understanding of how deep learning techniques behave in varying contexts and how they can be adapted and optimized for it.

A substantial effort is made to document the decision process of development and employ extensive analysis to empower the decision with comprehensive information. While the importance of architecture changes and extensions proved not to be very significant in the problem of tomato fruit detection, the significance of a well-structured deep learning pipeline was reinforced through the execution of the newly proposed methodologies: Artificial Size Slicing Aided Fine Tuning (ASSAFT) and Artificial Size Slicing Aided Hyper Inference (ASSAHI). These methods provide an innovative approach to data processing, specifically aimed at small object detection groups in high-resolution images. The results from the application of ASSAFT and ASSAHI on the Tomato360 dataset yielded notable enhancements in the success rates of object detection. These outcomes further underline the crucial role that data management plays in effectively implementing deep learning techniques into practice.

Finally, this doctoral thesis significantly contributes to both the scientific community and practical applications by highlighting the importance of a robust deep learning pipeline, introducing innovative methodologies for enhancing object detection in high-resolution images, and demonstrating the process and value of creating and using custom, real-world datasets. It is a stepping stone that bridges the gap between theory and practice in the field of deep learning, shedding light on the path for future research and applications.

7.1 Limitations and Future Directions

This dissertation offers significant advancements in the field of object detection in high-resolution images, particularly for the application of crop yield estimation in tomato greenhouses. However, there remain areas where improvements and further research would yield beneficial results.

The Tomato360 dataset, while it facilitated the creation of a successful large-scale tomato detection and counting system, still presents room for enhancements. Image data production from the 360 ° video could be improved to reduce low-level noise and brightness issues that may hamper the precision of object detection models. Furthermore, the dataset's usability in practice could be increased by including various tomato species, thus allowing the methodology to be robust across different tomato varieties.

Evaluating yield predictions during a longer period of time could provide valuable insights into the real-world applicability, stability, and efficiency of created models. Additionally, integrating a module for fruit ripeness determination would improve the usefulness of the system, as it would provide more nuanced information than the current one-class fruit detection system.

The proposed ASSAFT and ASSAHI methods have shown their value, yet there is room for further investigation. Evaluating different semantic segmentation models responsible for patch location could help optimize the system, as can be seen from the evaluation using ground true data. Additionally, testing the methodology on a variety of datasets, both within and outside the field of agriculture, could assess the general applicability of these methods.

Lastly, future research should aim to standardize the rules for patch production from image masks, making the methodology generally applicable across diverse datasets and problem contexts. This would not only extend the scope of these techniques but also provide a standard approach that could be replicated across different problem domains in high-resolution image analysis.

8 Conclusion

This dissertation presents a thorough exploration of small object detection within high-resolution images, focusing on applications across various domains. An in-depth study on attention mechanisms was conducted, where the successful implementation of a U-Net model with attention gates led to improved detection of abdominal organs and tumors in CT images. This achievement shed new light on the importance of attention in complex medical imaging. However, the research also discovered that similar attention mechanisms did not prove to be beneficial in the specific case of tomato detection, providing practical insights into the domain-specific nature of these techniques.

In the agricultural context, the work introduced a comprehensive framework for tomato fruit detection, demonstrating a multi-faceted approach that synergizes various cutting-edge techniques. Leveraging the Tood object detection model, novel methods: Artificial Size Slicing Fine Tuning (ASSFT), and Artificial Size Slicing Hyper Inference (ASSAHI), were developed, resulting in a solid F1-score of 0.89. These innovative techniques allowed for accurate yield predictions in a real-world setting, outperforming common agronomist estimates and providing an economically advantageous solution.

In conclusion, the research detailed in this dissertation contributes substantially to both the field of computer vision and practical applications within the medical and agricultural sectors. By advancing the understanding of attention mechanisms, innovating in small object detection, and demonstrating real-world applicability in applications from Tomato greenhouse, this work establishes a robust and reliable approach to high-resolution image analysis. The insights and methodologies developed throughout this research provide a robust foundation for future exploration, setting the stage for further refinement and expansion into diverse applications and challenges within object detection and beyond.

REFERENCES

[1] Deep Image: Scaling up Image Recognition. *CoRR*. 2015, abs/1501.02876. Dostupné z: `<http://arxiv.org/abs/1501.02876>`. Withdrawn.

[2] Akyon, F. C., Cengiz, C., Altinuc, S. O., Cavusoglu, D., Sahin, K. and Eryuksel, O. SAHI: A lightweight vision library for performing large scale object detection and instance segmentation, November 2021.

[3] Akyon, F. C., Altinuc, S. O. and Temizel, A. Slicing Aided Hyper Inference and Fine-tuning for Small Object Detection, 2022.

[4] Badrinarayanan, V., Kendall, A. and Cipolla, R. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*. 2017, 39, 12, pp. 2481–2495.

[5] Bilic, P. et al. The Liver Tumor Segmentation Benchmark (LiTS). *CoRR*. 2019, abs/1901.04056. Dostupné z: `<http://arxiv.org/abs/1901.04056>`.

[6] Birkfellner, W. *Applied medical image processing: a basic course*. CRC Press, 2016.

[7] Bodla, N., Singh, B., Chellappa, R. and Davis, L. S. Soft-NMS — Improving Object Detection with One Line of Code. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 5562–5570, 2017. doi: 10.1109/ICCV.2017.593.

[8] Chen, I.-T. and Lin, H.-Y. Detection, Counting and Maturity Assessment of Cherry Tomatoes using Multi-spectral Images and Machine Learning Techniques. In *VISIGRAPP (5: VISAPP)*, pp. 759–766, 2020.

[9] Chen, K. et al. MMDetection: Open MMLab Detection Toolbox and Benchmark. *arXiv preprint arXiv:1906.07155*. 2019.

[10] Chen, L.-C., Yang, Y., Wang, J., Xu, W. and Yuille, A. L. Attention to scale: Scale-aware semantic image segmentation. In *Proceedings of the*

IEEE conference on computer vision and pattern recognition, pp. 3640–3649, 2016.

[11] Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K. and Yuille, A. L. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*. 2017, 40, 4, pp. 834–848.

[12] Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K. and Yuille, A. L. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*. 2017, 40, 4, pp. 834–848.

[13] Chen, L.-C., Papandreou, G., Schroff, F. and Adam, H. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*. 2017.

[14] Chen, L.-C., Zhu, Y., Papandreou, G., Schroff, F. and Adam, H. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 801–818, 2018.

[15] Chu, J., Zhang, Y., Li, S., Leng, L. and Miao, J. Syncretic-NMS: A Merging Non-Maximum Suppression Algorithm for Instance Segmentation. *IEEE Access*. 2020, 8, pp. 114705–114714. doi: 10.1109/ACCESS.2020.3003917.

[16] Ciresan, D. C., Meier, U., Masci, J., Gambardella, L. M. and Schmidhuber, J. Flexible, high performance convolutional neural networks for image classification. In *Twenty-Second International Joint Conference on Artificial Intelligence*, 2011.

[17] Contributors, M. MMSegmentation: OpenMMLab Semantic Segmentation Toolbox and Benchmark. `https://github.com/open-mmlab/mmsegmentation`, 2020.

[18] Dai, Z., Yang, Z., Yang, Y., Carbonell, J., Le, Q. V. and Salakhutdinov, R. Transformer-xl: Attentive language models beyond a fixed-length context. *arXiv preprint arXiv:1901.02860.* 2019.

[19] Dalal, N. and Triggs, B. Histograms of oriented gradients for human detection. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, 1, pp. 886–893. Ieee, 2005.

[20] Dechter, R. Learning While Searching in Constraint-Satisfaction-Problems. pp. 178–185, 01 1986.

[21] Dice, L. R. Measures of the amount of ecologic association between species. *Ecology.* 1945, 26, 3, pp. 297–302.

[22] Ding, J. et al. Object detection in aerial images: A large-scale benchmark and challenges. *IEEE transactions on pattern analysis and machine intelligence.* 2021, 44, 11, pp. 7778–7796.

[23] Dominik, S. and Jacek, N. *Computer vision in robotics and industrial applications.* 3. World Scientific, 2014.

[24] Feng, C., Zhong, Y., Gao, Y., Scott, M. R. and Huang, W. Tood: Task-aligned one-stage object detection. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 3490–3499. IEEE Computer Society, 2021.

[25] Forsyth, D. A. and Ponce, J. *Computer vision: a modern approach.* Prentice Hall Professional Technical Reference, 2002.

[26] Fu, C., Liu, W., Ranga, A., Tyagi, A. and Berg, A. C. DSSD : Deconvolutional Single Shot Detector. *CoRR.* 2017, abs/1701.06659. Dostupné z: `<http://arxiv.org/abs/1701.06659>`.

[27] Fuglie, K. The growing role of the private sector in agricultural research and development world-wide. *Global Food Security.* 2016, 10, pp. 29–38. ISSN 2211-9124. doi: https://doi.org/10.1016/j.gfs.2016.07.005.

[28] Girshick, R. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pp. 1440–1448, 2015.

[29] Girshick, R., Donahue, J., Darrell, T. and Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 580–587, 2014.

[30] Girshick, R., Donahue, J., Darrell, T. and Malik, J. Region-based convolutional networks for accurate object detection and segmentation. *IEEE transactions on pattern analysis and machine intelligence.* 2015, 38, 1, pp. 142–158.

[31] Gongal, A., Amatya, S., Karkee, M., Zhang, Q. and Lewis, K. Sensors and systems for fruit detection and localization: A review. *Computers and Electronics in Agriculture.* 2015, 116, pp. 8–19.

[32] Goodfellow, I., Bengio, Y. and Courville, A. *Deep Learning.* MIT Press, 2016. Dostupné z: `<http://www.deeplearningbook.org>`.

[33] Grewal, M., Srivastava, M. M., Kumar, P. and Varadarajan, S. Radnet: Radiologist level accuracy using deep learning for hemorrhage detection in ct scans. In *2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018)*, pp. 281–284. IEEE, 2018.

[34] He, K., Zhang, X., Ren, S. and Sun, J. Deep Residual Learning for Image Recognition. *CoRR.* 2015, abs/1512.03385. Dostupné z: `<http://arxiv.org/abs/1512.03385>`.

[35] He, Z., Xie, L., Chen, X., Zhang, Y., Wang, Y. and Tian, Q. Data Augmentation Revisited: Rethinking the Distribution Gap between Clean and Augmented Data, 2019.

[36] Hoiem, D., Chodpathumwan, Y. and Dai, Q. Diagnosing error in object detectors. In *European conference on computer vision*, pp. 340–353. Springer, 2012.

[37] Hu, P., Wu, F., Peng, J., Bao, Y., Chen, F. and Kong, D. Automatic abdominal multi-organ segmentation using deep convolutional neural network and time-implicit level sets. *International journal of computer assisted radiology and surgery.* 2017, 12, 3, pp. 399–411.

[38] Huang, J. et al. Speed/Accuracy Trade-Offs for Modern Convolutional Object Detectors. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3296–3297, July 2017. doi: 10.1109/CVPR.2017.351.

[39] Huang, Z., Wang, X., Huang, L., Huang, C., Wei, Y. and Liu, W. CCNet: Criss-Cross Attention for Semantic Segmentation. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019.

[40] Hubel, D. H. and Wiesel, T. N. Receptive fields and functional architecture of monkey striate cortex. *The Journal of physiology*. 1968, 195, 1, pp. 215–243.

[41] Illingworth, J. and Kittler, J. A survey of the Hough transform. *Computer vision, graphics, and image processing*. 1988, 44, 1, pp. 87–116.

[42] Isensee, F., Jaeger, P. F., Kohl, S. A., Petersen, J. and Maier-Hein, K. H. nnU-Net: a self-configuring method for deep learning-based biomedical image segmentation. *Nature methods*. 2021, 18, 2, pp. 203–211.

[43] Kayalibay, B., Jensen, G. and Smagt, P. CNN-based Segmentation of Medical Imaging Data. *CoRR*. 2017, abs/1701.03056. Dostupné z: `<http://arxiv.org/abs/1701.03056>`.

[44] Kearney, V., Chan, J. W., Wang, T., Perry, A., Yom, S. S. and Solberg, T. D. Attention-enabled 3D boosted convolutional neural networks for semantic CT segmentation using deep supervision. *Physics in Medicine & Biology*. jul 2019, 64, 13, pp. 135001. doi: 10.1088/1361-6560/ab2818.

[45] Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*. 2014.

[46] Kirillov, A., He, K., Girshick, R., Rother, C. and Dollár, P. Panoptic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9404–9413, 2019.

[47] LeCun, Y., Bengio, Y. and Hinton, G. Deep learning. *nature*. 2015, 521, 7553, pp. 436–444.

[48] Lee, H., Hwang, S. J. and Shin, J. Rethinking Data Augmentation: Self-Supervision and Self-Distillation, 2019.

[49] Li, C., Yang, T., Zhu, S., Chen, C. and Guan, S. Density map guided object detection in aerial images. In *proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pp. 190–191, 2020.

[50] Lin, G., Adiga, U., Olson, K., Guzowski, J. F., Barnes, C. A. and Roysam, B. A hybrid 3D watershed algorithm incorporating gradient cues and object models for automatic segmentation of nuclei in confocal image stacks. *Cytometry Part A: the journal of the International Society for Analytical Cytology.* 2003, 56, 1, pp. 23–36.

[51] Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P. and Zitnick, C. L. Microsoft coco: Common objects in context. In *European conference on computer vision*, pp. 740–755. Springer, 2014.

[52] Lin, T., Goyal, P., Girshick, R. B., He, K. and Dollár, P. Focal Loss for Dense Object Detection. *CoRR.* 2017, abs/1708.02002. Dostupné z: `<http://arxiv.org/abs/1708.02002>`.

[53] Liu, G., Nouaze, J. C., Touko Mbouembe, P. L. and Kim, J. H. YOLO-Tomato: A Robust Algorithm for Tomato Detection Based on YOLOv3. *Sensors.* 2020, 20, 7. ISSN 1424-8220. doi: 10.3390/s20072145.

[54] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S. E., Fu, C. and Berg, A. C. SSD: Single Shot MultiBox Detector. *CoRR.* 2015, abs/1512.02325. Dostupné z: `<http://arxiv.org/abs/1512.02325>`.

[55] Long, J., Shelhamer, E. and Darrell, T. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3431–3440, 2015.

[56] Lowe, D. G. Distinctive image features from scale-invariant keypoints. *International journal of computer vision.* 2004, 60, pp. 91–110.

[57] Mittal, S. and Vaishay, S. A Survey of Techniques for Optimizing Deep Learning on GPUs. *Journal of Systems Architecture*. 2019, pp. 101635.

[58] Mnih, V., Heess, N., Graves, A. and others. Recurrent models of visual attention. In *Advances in neural information processing systems*, pp. 2204–2212, 2014.

[59] Mu, G., Lin, Z., Han, M., Yao, G. and Gao1, Y. Segmentation of kidney tumor by multi-resolution VB-nets. Technical report, Shanghai United Imaging Intelligence Inc., Shanghai, China, 2019.

[60] Mu, Y., Chen, T.-S., Ninomiya, S. and Guo, W. Intact Detection of Highly Occluded Immature Tomatoes on Plants Using Deep Learning Techniques. *Sensors*. 2020, 20, 10. ISSN 1424-8220. doi: 10.3390/s20102984.

[61] Mureşan, H. and Oltean, M. Fruit recognition from images using deep learning. *arXiv preprint arXiv:1712.00580*. 2017.

[62] Oktay, O. et al. Attention U-Net: Learning Where to Look for the Pancreas. *CoRR*. 2018, abs/1804.03999. Dostupné z: `<http://arxiv.org/abs/1804.03999>`.

[63] Parico, A. I. B. and Ahamed, T. Real Time Pear Fruit Detection and Counting Using YOLOv4 Models and Deep SORT. *Sensors*. 2021, 21, 14. ISSN 1424-8220. doi: 10.3390/s21144803.

[64] Plaut, D. C. and others. Experiments on Learning by Back Propagation. 1986.

[65] Powers, D. Evaluation: From Precision, Recall and F-Factor to ROC, Informedness, Markedness Correlation. *Mach. Learn. Technol.* 01 2008, 2.

[66] Redmon, J., Divvala, S., Girshick, R. and Farhadi, A. You only look once: Unified, real-time object detection. 2016, pp. 779–788.

[67] Ren, S., He, K., Girshick, R. B. and Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *CoRR*. 2015, abs/1506.01497. Dostupné z: `<http://arxiv.org/abs/1506.01497>`.

[68] Ren, S., He, K., Girshick, R. and Sun, J. Faster R-CNN: towards real-time object detection with region proposal networks. *IEEE transactions on pattern analysis and machine intelligence*. 2016, 39, 6, pp. 1137–1149.

[69] Robbins, H. and Monro, S. A stochastic approximation method. *The annals of mathematical statistics*. 1951, pp. 400–407.

[70] Rodríguez-Sánchez, A., Oberleiter, S., Xiong, H. and Piater, J. Learning V4 Curvature Cell Populations from Sparse Endstopped Cells. In *International Conference on Artificial Neural Networks*, pp. 463–471. Springer, 2016.

[71] Ronneberger, O., Fischer, P. and Brox, T. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241. Springer, 2015.

[72] Rosenfeld, A. and Thurston, M. Edge and curve detection for visual scene analysis. *IEEE Transactions on computers*. 1971, 100, 5, pp. 562–569.

[73] Russell, B. C., Torralba, A., Murphy, K. P. and Freeman, W. T. LabelMe: a database and web-based tool for image annotation. *International journal of computer vision*. 2008, 77, 1-3, pp. 157–173.

[74] Shamshiri, R. Measuring optimality degrees of microclimate parameters in protected cultivation of tomato under tropical climate condition. *Measurement*. 2017, 106, pp. 236–244. ISSN 0263-2241. doi: https://doi.org/10.1016/j.measurement.2017.02.028.

[75] Shaw, P., Uszkoreit, J. and Vaswani, A. Self-attention with relative position representations. *arXiv preprint arXiv:1803.02155*. 2018.

[76] Short, T., Draper, C. and Donnell, M. Web-based decision support system for hydroponic vegetable production. In *International Conference on Sustainable Greenhouse Systems-Greensys2004 691*, pp. 867–870, 2004.

[77] Simonyan, K. and Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*. 2014.

[78] Simpson, A. L. et al. A large annotated medical image dataset for the development and evaluation of segmentation algorithms. *CoRR*. 2019, abs/1902.09063. Dostupné z: `<http://arxiv.org/abs/1902.09063>`.

[79] Taha, A. A. and Hanbury, A. Metrics for evaluating 3D medical image segmentation: analysis, selection, and tool. *BMC medical imaging*. 2015, 15, 1, pp. 29.

[80] Taha, A. A. and Hanbury, A. Metrics for evaluating 3D medical image segmentation: analysis, selection, and tool. *BMC medical imaging*. 2015, 15, 1, pp. 1–28.

[81] Tieleman, T. and Hinton, G. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning*. 2012, 4, 2, pp. 26–31.

[82] Torralba, A., Efros, A. A. and others. Unbiased look at dataset bias. In *CVPR*, 1, pp. 7. Citeseer, 2011.

[83] Uzkent, B., Yeh, C. and Ermon, S. Efficient object detection in large images using deep reinforcement learning. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pp. 1824–1833, 2020.

[84] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł. and Polosukhin, I. Attention is all you need. *Advances in neural information processing systems*. 2017, 30.

[85] Viola, P. and Jones, M. Rapid object detection using a boosted cascade of simple features. In *Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001*, 1, pp. I–I. Ieee, 2001.

[86] Wan, S. and Goudos, S. Faster R-CNN for multi-class fruit detection using a robotic vision system. *Computer Networks*. 2020, 168, pp. 107036. ISSN 1389-1286. doi: https://doi.org/10.1016/j.comnet.2019.107036.

[87] Wang, F., Jiang, M., Qian, C., Yang, S., Li, C., Zhang, H., Wang, X. and Tang, X. Residual Attention Network for Image Classification. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[88] Wei, X., Jia, K., Lan, J., Li, Y., Zeng, Y. and Wang, C. Automatic method of fruit object extraction under complex agricultural background for vision system of fruit picking robot. *Optik*. 2014, 125, 19, pp. 5684–5689.

[89] Wei, Z. and Duan, C. AMRNet: Chips Augmentation in Areial Images Object Detection. *CoRR*. 2020, abs/2009.07168. Dostupné z: `<https://arxiv.org/abs/2009.07168>`.

[90] Wilson, A. C., Roelofs, R., Stern, M., Srebro, N. and Recht, B. The marginal value of adaptive gradient methods in machine learning. In *Advances in Neural Information Processing Systems*, pp. 4148–4158, 2017.

[91] Xiao, T., Xu, Y., Yang, K., Zhang, J., Peng, Y. and Zhang, Z. The application of two-level attention models in deep convolutional neural network for fine-grained image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 842–850, 2015.

[92] Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhudinov, R., Zemel, R. and Bengio, Y. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, pp. 2048–2057, 2015.

[93] Xu, Z.-F., Jia, R.-S., Liu, Y.-B., Zhao, C.-Y. and Sun, H.-M. Fast Method of Detecting Tomatoes in a Complex Scene for Picking Robots. *IEEE Access*. 2020, 8, pp. 55289–55299. doi: 10.1109/ACCESS.2020.2981823.

[94] Yang, G. et al. Automatic Segmentation of Kidney and Renal Tumor in CT Images Based on 3D Fully Convolutional Neural Network with Pyramid Pooling Module. In *2018 24th International Conference on Pattern Recognition (ICPR)*, pp. 3790–3795. IEEE, 2018.

[95] Ypsilantis, P.-P. and Montana, G. Learning what to look in chest X-rays with a recurrent visual attention model. *arXiv preprint arXiv:1701.06452*. 2017.

[96] Zhang, W., Itoh, K., Tanida, J. and Ichioka, Y. Parallel distributed processing model with local space-invariant interconnections and its optical architecture. *Applied optics*. 1990, 29, 32, pp. 4790–4797.

[97] Zhang, Y., Zhou, D., Chen, S., Gao, S. and Ma, Y. Single-image crowd counting via multi-column convolutional neural network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 589–597, 2016.

[98] Zhao, Y., Gong, L., Huang, Y. and Liu, C. A review of key techniques of vision-based control for harvesting robot. *Computers and Electronics in Agriculture*. 2016, 127, pp. 311–323. ISSN 0168-1699. doi: https://doi.org/10.1016/j.compag.2016.06.022.

[99] Zhu, P., Wen, L., Du, D., Bian, X., Fan, H., Hu, Q. and Ling, H. Detection and Tracking Meet Drones Challenge. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2021, pp. 1–1. doi: 10.1109/TPAMI.2021.3119563.

[100] Zhu, X., Cheng, D., Zhang, Z., Lin, S. and Dai, J. An empirical study of spatial attention mechanisms in deep networks. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 6688–6697, 2019.

[101] Zhu, X., Hu, H., Lin, S. and Dai, J. Deformable convnets v2: More deformable, better results. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9308–9316, 2019.

PUBLICATIONS OF THE AUTHOR

Journal Publications with Impact Factor

[A1] Turečková, A., Tureček, T., Janků, P., Vařacha, P., Šenkeřík, R., Jašek, R., Psota, V., Štěpánek, V., Komínková Oplatková, Z., Slicing aided large scale tomato fruit detection and counting in 360-degree video data from a greenhouse. In *Measurement*, vol. 204, pp 111977, 2022, Elsevier. ISSN 0263-2241. DOI: `10.1016/j.measurement.2022.111977`

[A2] Tureckova, A., Turecek, T., Kominkova Oplatkova, Z., Rodríguez-Sánchez, A. J. Improving CT Image Tumor Segmentation Through Deep Supervision and Attentional Gates. In *Frontiers in Robotics and AI*, vol. 7, pp 106, 2020, Frontiers. ISSN 2296-9144. DOI: `10.3389/frobt.2020.00106`

[A3] Li, H., Zhang, H., Xu, Y., Tureckova, A.„ Zahradník, P., Chang, H., Neuzil, P. Versatile digital polymerase chain reaction chip design, fabrication, and image processing, In *Sensors and Actuators B: Chemical*, vol. 283, pp. 677-684, 2019. Elsevier B.V. ISSN 0925-4005. DOI: `10.1016/j.snb.2018.12.072`.

Journal Publications Indexed in Scopus

[A4] Tureckova, A., Holik, T. Kominkova Oplatkova, Z. Dog Face Detection Using YOLO Network. In *MENDEL*, vol. 26, num. 2, pp 17-22, 2020. DOI: `10.13164/mendel.2020.2.017`

Conference Proceedings

[A5] Tureckova, A., Turecek, T., Kominkova Oplatkova, Z. ICIP 2022 Challenge: PEDCMI, TOOD Enhanced by Slicing-Aided Fine-Tuning and Inference, In *2022 IEEE International Conference on Image Processing (ICIP)*, pp. 4292–4295, Bordeaux, France, 2022. DOI: `10.1109/ICIP46576.2022.9897826`.

[A6] Tureček, T., Vařacha, P., Turečková, A., Psota, V., Janků, P., Štěpánek, V., Viktorin, A., Šenkeřík, R., Jašek, R., Chramcov, B., Grivas, I., Komínková Oplatková, Z., Scouting of Whiteflies in Tomato Greenhouse Environment Using Deep Learning. In *Agriculture Digitalization and Organic Production*, pp. 323-335, Singapore, 2022. Springer Singapore. ISBN 978-981-16-3349-2.

[A7] Tureckova, A., Turecek, T., Kominkova Oplatkova, Z., Rodríguez-Sánchez, A. J. KiTS challenge: VNet with attention gates and deep supervision, In *KiTS 2019 challenge*, preprint, 2020. URL `http://results.kits-challenge.org/miccai2019/manuscripts/tureckova_2.pdf`.

[A8] Tureckova, A., and Rodríguez-Sánchez, A. J. ISLES Challenge: U-Shaped Convolution Neural Network with Dilated Convolution for 3D Stroke Lesion Segmentation, In *Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries*, pp. 319-327, Cham, 2019. Springer International Publishing. ISBN 978-3-030-11723-8. DOI: `10.1007/978-3-030-11723-8_32`.

[A9] Vlachynska, A., , Kominkova Oplatkova, Z. , Turecek, T. Dogface Detection and Localization of Dogface's Landmarks, In *Artificial Intelligence and Algorithms in Intelligent Systems*, pp. 465-476, Cham, 2019. Springer International Publishing. ISBN 978-3-319-91189-2. DOI: `10.1007/978-3-319-91189-2_46`.

[A10] Vlachynska, A., Kominkova Oplatkova, Z., Sramka, M. The coordinate system of the eye in cataract surgery: Performance comparison of the circle Hough transform and Daugman's algorithm, In *AIP Conference Proceedings*, vol. 1863. 2017. AIP Publishing. DOI: `10.1063/1.4992259`.

[A11] Vlachynska, A., Cerveny, J., Cmiel, V., Turecek, T. Automatic Image-Based Method for Quantitative Analysis of Photosynthetic Cell Cultures, In *Hybrid Artificial Intelligent Systems*, pp. 402-413, Cham, 2016. Springer International Publishing. ISBN 978-3-319-32034-2. DOI: `10.1007/978-3-319-32034-2_34`.

[A12] Vlachynska, A., Sramka, M. Artificial Neural Networks Application in Intraocular Lens Power Calculation, In *Conference: 9th EUROSIM Congress on Modelling and SimulationAt: Oulu, Finland*, 2016. DOI: `10.1109/EUROSIM.2016.45`.

CURRICULUM VITAE

Personal Information

Name: Alžběta Turečková, maiden name Vlachynská
E-mail: tureckova@utb.cz
Birth: 27th May 1991
Nationality: Czech

Personal Skills and Competences

Mother tongue: Czech
Other languages: English (C1)
Computer skills and competences: Python, C++, Matlab, PyTorch, OpenCV
Social skills and competencies : Friendly and communicative, she is capable of working independently and also thrives in team settings, including those with an international composition. Her adaptability has been honed through multiple study internships and participation in project teams under the Technology Agency of the Czech Republic (TACR). These experiences have fortified her interpersonal skills and ability to adjust to diverse work environments.

Education and Training

2015 - now	**Engineering Informatics** (Doctoral studies) Tomas Bata University in Zlin, Faculty of Applied Informatics Aim of dissertation: Soft Computing Methods in Computer Vision • Researcher (application of research, grants, projects) • Artificial Intelligence Laboratory member (ailab.fai.cz) • Lecturer
2013 - 2015	**Biomedical Engineering and Bioinformatics** (Master's degree) Brno University of Technology, FEEC Thesis: Photosynthetic cell suspension cultures quantitative image data processing Overall study results: passed with honor.
2010 - 2013	**Biomedical Technology and Bioinformatics** (Bachelor's degree) Brno University of Technology, FEEC Thesis: Searching adenine and guanine-rich regions Overall study results: passed with honor.

Study Internships

04 - 08/2018	**University of Innsbruck, Austria** The deep learning models for image classification and segmentation The internship focused on understanding and modifying CNN-based image classification and segmentation models. A medical data segmentation model was created for the ISLES 2019 challenge, published and presented at MICCAI 2019.
05 - 07/2017	**University of Oviedo, Oviedo, the Kingdom of Spain** Image-based analysis of root development in Arabidopsis Developed a program for automatic measurement of root length and root hair count in Arabidopsis.
01 - 04/2017	**Northwestern Polytechnical University, Xi'an, P. R. China** Automatic image-based analysis of qPCR on the chip Worked on quantifying DNA detection on a chip based on the detection of positive/negative sub-reaction cells in fluorescent microscope images. The research findings were later published in a journal.

Work Experience

10/2019-now	**Tomas Bata University in Zlin, Faculty of Applied Informatics** Lecturer of courses Software Project Development Tools, Computer Science Basic, and Neural Networks. Consultant and tutor of bachelor and diploma theses. Junior researcher in A. I. Lab (ailab.fai.cz). Team member of projects from the Technology Agency of the Czech Republic. Her main research interest and professional focus lie on artificial intelligence methods applied in computer vision, deep learning, convolutional neural networks, and medical image processing.
2015-2017	**GEMINI oční klinika a.s.** Biomedical engineer (Biomedical engineering, Medical devices) Manager of internal IS. She did requirements analysis of medical workflow and their implementation into the clinical system. Alpha/beta - testing of the clinical system. Definition and management of clinical reports.

Publication Activity

Orcid ID: 0000-0002-5566-7393

Nr. publications: 12 (3 IF journals, 1 Scopus journal, 8 conference proceedings)

Nr. citations: 84 (excluding self-citations), **H-index:** 5

Alžběta Turečková

Deep Learning Methods Applied in Computer Vision

Využití metod hlubokého učení v počítačovém vidění

Doctoral Thesis

Published by Tomas Bata University in Zlin

Nad Stráněmi 4511,760 05 Zlín, Czech Republic

First edition

Typography: Alžběta Turečková

This publication did not pass through a language nor an editorial revision.

The year of publication: 2023