UCLouvain
Institute for Information
and Communication Technologies,
Electronics and Applied Mathematics

Reducing manual annotation load for CNN-based image segmentation

Nana Yang

Thesis submitted in partial fulfillment
of the requirements for the degree of
Ph.D. in Engineering Science and Technology

Dissertation committee:

Prof. Christophe De Vleeschouwer (UCLouvain, advisor)
Prof. Benoit Macq (UCLouvain)
Prof. Christophe Craeye (UCLouvain, chair)
Dr. Adrien Deliège (Université de Liège, Belgium)
Prof. Saïd Mahmoudi (University of Mons, Belgium)

Version of October 2024 .

Abstract

Convolutional neural networks (CNNs) have demonstrated remarkable performance in semantic segmentation, typically relying on extensive labeled datasets. However, obtaining such data in practical applications is challenging and entails significant human and time costs. This thesis aims to explore methods for achieving semantic segmentation with limited manual annotation. We investigate three primary approaches to reducing the need for manual labelling: utilizing pseudo-labels, leveraging single-class samples, and employing graph-cut methods with prior information.

First, in scenarios where a large number of unlabeled samples are accessible, pseudo-labels can be generated to train a model with improved test accuracy. Specifically, pseudo-labels are automatically generated by a machine learning model trained on a small set of manually annotated samples. The pseudo-labeled data are then used to train a so-called data-reinforced model that is shown to achieve a better test accuracy than the one obtained by a model solely trained on the manually annotated data. Our work demonstrates that data-reinforcement is stimulated by exploiting soft pseudo-labels, when knowledge about label certainty is available, or by increasing the diversity of manual annotations through iterative active learning mechanisms, in conjunction with a sufficiently expressive model as pseudo-label generator.

Second, at the data collection stage, when single-class samples are available, our work demonstrates that they can be leveraged to learn a multi-class model. This approach utilizes a straightforward background/foreground segmentation solution to isolate the single class foreground objects, thereby effectively reducing the annotation effort required for training the multi-class model. For instance, annotating mixed pollen images is difficult, but pure pollen images, containing only one type of pollen, are easier to annotate. By training a universal pollen segmentation model on a small

set of semi-manually labeled pure pollen images, we can automate the segmentation of a larger pure pollen dataset. This dataset then supervises the training of a pollen species segmentation model, which is shown to accurately segment species within mixed pollen images despite being trained on pure images.

Third, when prior information about the topology of objects to be segmented is available, graph-cut methods can be employed. These methods naturally integrate the prior knowledge to provide an initial rough segmentation, which can then be manually refined, thereby reducing the overall time required for manual annotation. For example, in the medical imaging domain, annotation requires high expertise and is labor-intensive. We propose utilizing existing anatomical knowledge and traditional image segmentation techniques to generate initial segmentation results. These results are refined by non-experts to create an annotated dataset for CNN training.

Through these methods, this research demonstrates significant advancements in semantic segmentation with practical implications across various domains, reducing the dependency on extensive labeled datasets and its associated cost in expert time.

Acknowledgements

Since arriving in Belgium in 2020, my unforgettable four-year doctoral journey is now drawing to a close. As countless memories flood my mind, I feel a mix of emotions, but gratitude stands out the most.

First and foremost, I want to express my deepest thanks to my supervisor, Prof. Christophe De Vleeschouwer. He provided me with the opportunity to pursue my doctorate, offered unwavering support throughout my research, and guided me when I felt lost. Despite his numerous research commitments, he always took the time to patiently and meticulously guide me. Words cannot fully convey my gratitude. His passion, curiosity, rigor, and professionalism in academics, along with his trust in his students and the freedom he granted us, will profoundly influence my future studies, career, and life.

Secondly, I would like to thank my colleague, Victor Joos de ter Beerst. In the early days of my doctoral studies, he consistently contributed unique insights during our work discussions and helped me overcome various challenges. I am also grateful to Professor Anne-Laure Jacquemart for her support with the data in my research. When I was initially unfamiliar with pollen data, she patiently explained everything and assisted me in my studies. My gratitude also extends to my thesis jury for their thorough reading and invaluable feedback.

A big thank you to all my colleagues for their assistance and for fostering a humorous, free, and friendly working atmosphere.

I am also profoundly grateful to my family for their help, understanding, and support. Special thanks to my father and mother for their comfort, understanding, and tolerance, which provided me with the motivation to persevere.

Finally, I would like to thank UCLouvain for offering a free and friendly academic environment and for its partial financial support. I also extend

my thanks to the China Scholarship Council for its financial assistance.

List of publications

- Yang, Nana, Victor Joos, Anne-Laure Jacquemart, Christel Buyens, and Christophe De Vleeschouwer. "Using Pure Pollen Species When Training a CNN to Segment Pollen Mixtures." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, pp. 1695-1704. 2022.
- Yang, Nana, Charles Rongione, Anne-Laure Jacquemart, Xavier Draye, and Christophe De Vleeschouwer. "On the Importance of Diversity When Training Deep Learning Segmentation Models with Error-Prone Pseudo-Labels." Applied Sciences 14, no. 12 (2024): 5156.
- Yang, Nana, Robin Verschuren, and Christophe De Vleeschouwer. "Graph-cut-assisted CNN training for pulmonary embolism segmentation." ESANN 2024.
- Han, Sutao, Nana Yang, Lihua Wang, and Magd Abdel Wahab. "Modelling of microscopic crack initiation behaviour of fretting fatigue based on X-ray micro-computed tomography scan." International Journal of Fatigue 186 (2024): 108374.

Contents

Chapter 1

Introduction

1.1 Context

In the Information Age, images serve as a vital medium for conveying information . They communicate information efficiently and intuitively, playing an increasingly role across a multitude of applications. Digital images, defined as images represented by discrete numerical values, can be likened to a mosaic of tiny squares, where each square is referred to as a pixel.

Digital image processing encompasses a range of methods and technologies aimed at enhancing images, removing noise, segmenting regions, and extracting features, all through computational means. As computing power and mathematical techniques progress, the demands across fields such as agriculture, forestry, industry, and medicine, digital image processing has seen rapid and substantial development.

Among the various tasks in digital image processing, semantic segmentation stands out as one of the most crucial. Semantic segmentation involves assigning each pixel in an image to a specific semantic category, thus enabling pixel-level understanding and analysis of the image. This task is fundamental in the field of computer vision, and is useful in numerous applications. For instance, in autonomous driving, it assists vehicles in making accurate decisions by identifying traffic signs, pedestrians, roads, vehicles, and more. In medical imaging, semantic segmentation aids doctors in identifying and segmenting tumors, organs, lesions, and other critical areas.

Early semantic segmentation techniques were often dependent on manually designed features and rules, which may not perform well in complex scenes or datasets with significant variation. Today, most popular semantic segmentation methods are based on deep learning.

Deep learning is a subset of machine learning. Just as humans learn from experience, deep learning models acquire knowledge through large volumes of data (i.e., experience). Through this "learning" process, they can uncover intricate patterns and features within the data.

The introduction of the fully convolutional networks (FCNs) [LSD15] in 2014 signaled the beginning of semantic segmentation's journey into the deep learning era. FCNs achieve end-to-end pixel-level prediction by substituting the fully connected layers with convolutional layers, thereby accomplishing pixel-wise classification tasks.

From the mid-2010s to the present, deep learning methods have continued to evolve. In recent years, researchers have refined and optimized these methods, proposing various network architectures and technologies specifically for semantic segmentation, such as U-Net [RFB15a], Residual U-net [ZLW18], the DeepLab series [CPK+14, CPK+17, CPSA17, CZP+18], Attention U-Net [OSF+18], and Inception U-Net [ZWCK20]. These innovations have enhanced model architecture, feature fusion, and context understanding, leading to significant improvements in segmentation performance.

Despite the impressive performance of deep learning, it still faces numerous challenges in practical applications. This paper focuses specifically on the limitations associated with the collection of data labels. Most effective deep learning models rely on supervised learning, where the model learns from a set of labeled data. Labeling the data means the model is explicitly told which class is represented by each pixel during the learning process. Labeled data provides a supervisory signal for model training, guiding the model on the expected output given a specific input.

By training on labeled data, the deep learning model adjusts its internal parameters, enabling it to make accurate predictions. This process helps the model learning patterns and features within the data, hopefully generalizing this ability on unseen data. In essence, labeled data is crucial for model learning, determining both performance and generalization capabilities, and playing an indispensable role in training deep learning models.

However, using supervised learning to train deep learning models necessitates a substantial amount of labeled images. Obtaining high-quality pixel-level labeled data is often challenging, as the annotation process is

time-consuming and labor-intensive for human experts. This paper is dedicated to addressing the issue of reducing the annotation effort in deep learning training.

1.2 Research questions, objectives and significance

Research question In image classification, a picture has only one label, while in semantic segmentation, each pixel has a label. The labeling workload in semantic segmentation is considerable, and the difficulty and cost of obtaining labeled data are high. Insufficient labeled data may cause the model to fail in learning the true data distribution, instead memorizing noise in the training data or characteristics of specific samples. This can lead to overfitting, poor generalization ability, performance degradation, and training instability.

Research purpose This thesis aims to address the challenges of CNN training for image semantic segmentation when labeled data is insufficient.

Research Significance By addressing the challenges of CNN training with insufficient labeled data, this research can reduce dependence on labeled data, thereby lowering training costs. Additionally, it can enhance the generalization ability of semantic segmentation models on small datasets, providing efficient and low-cost solutions for practical applications. This is expected to promote the development of related fields and expand the applicability of deep learning across various application scenarios.

1.3 Contributions

On the Importance of Diversity When Training Deep Learning Segmentation Models with Error-prone Pseudo-labels Traditional machine learning methods for image segmentation are less expressive and thus less reliant on labeled data than deep learning models. To address the challenge of limited labeled data, we leverage both traditional machine learning and deep learning methods to propose data-reinforced models. In practice, an initial (conventional or CNN) model is trained from a small set of manually labeled data, and is used to generate pseudo-labels on unlabeled data. We demonstrate that by using those pseudo-labels to train a so-called data-reinforced model, with the help of large unlabeled datasets, it is possible to achieve higher segmentation accuracy even when using erroneous

pseudo-labels. Interestingly, our work reveals that increasing diversity in the generation of pseudo-labels can enhance the performance of data-reinforced models. This increase in diversity can be achieved by enriching manual annotations through interactive selection and accurate labeling of hard[1] samples or by considering soft pseudo-labels for each sample when prior information about their certainty is available. Specifically, our experiments demonstrate that, at constant pseudo-label generator accuracies, stronger data-reinforced models are obtained when using diverse annotations. Moreover, when using the manual annotation time to measure the quantity of annotation, they also reveal that less annotations are needed to achieve a given data-reinforced model accuracy when annotation diversity increases.

Training multiclass segmentation models from single class images This study has considered the use of **using pure pollen species when training a CNN to segment pollen mixtures**. Labeling mixed pollen images is a challenging task even for highly skilled biological experts, as pollens are often very similar and difficult to distinguish. However, labeling pure pollen images (i.e., images containing only one type of pollen) is relatively straightforward since pollen grain significantly differ from their background in microscopy images. To reduce labeling effort, we propose a method to segment mixed pollen images using a dataset of pure pollen images. First, we train a universal foreground/background pollen segmentation model on a small set of (semi-)manually labeled pure pollen images. This model is then used to automatically segment a larger pure pollen dataset, which in turn supervises the training of a multi-species segmentation model. Although trained on pure images, we demonstrate that the model can accurately segment pollen mixtures. Additionally, by learning a set of models, each of them recognizing only a subset of species, it becomes feasible to define a bin pollen class, making the set of models effective in handling unexpected species in a mixture.

Graph-cut-assisted CNN Training for Pulmonary Embolism Segmentation Medical image annotation requires highly skilled experts and is both time-consuming and labor-intensive. To address this, we propose to embed prior anatomical knowledge of experts in a graph-cut and traditional image segmentation framework to obtain initial segmentation results. These results are then refined by non-professionals to create an annotated dataset for training CNN models.

[1] In the sense that the sample is wrongly predicted by the current state of the pseudo-label generator.

1.4 Thesis outline

This thesis is structured as follows. In Chapter 2, we introduce the concepts of semantic segmentation, providing an overview of its historical development, application areas, challenges, solutions, and evaluation metrics. This section aims to furnish readers unfamiliar with semantic segmentation with foundational knowledge relevant to our research field. Chapters 3, 4, and 5 delve into detailed descriptions of the contributions outlined earlier. Each chapter elaborates on specific aspects crucial to our research. Chapter 6 offers future perspectives based on the contributions presented in this thesis.

Chapter 2

Semantic segmentation methods

This chapter begins by defining semantic segmentation, outlining its goals, and differentiating it from traditional image segmentation and instance segmentation. Section 2.2 introduces the evaluation criteria for semantic segmentation. Section 2.3 provides a historical overview of semantic segmentation development. Section 2.4 discusses the challenges associated with semantic segmentation, which are central to the research topic of this thesis. In Section 2.5, we summarize the solutions to these challenges.

2.1 What is semantic segmentation?

Traditional image segmentation divides an image into several regions based solely on visual similarity between pixels, without considering the semantic information within each region. As a result, each region may contain multiple different semantic categories despite having similar visual features. In contrast, instance segmentation focuses not only on the semantic information within the image but also on distinguishing the boundaries between different instances. This approach assigns pixels to semantic categories while also differentiating between distinct instances within the same category.

Semantic segmentation is a crucial task in computer vision. Unlike traditional image segmentation and instance segmentation, it focuses on the semantic information within an image and disregards differences between

individual instances. Semantic segmentation classifies objects at the pixel level, requiring each pixel to be labeled according to its category. Unlike image classification, which assigns one label per image, semantic segmentation assigns one label per pixel.

The task of semantic segmentation can be defined as follows: given an input image I, the goal is to generate an output image O of the same size, where each pixel O_{ij} represents the semantic category of the corresponding pixel I_{ij}. By assigning a semantic category to each pixel, the image is divided into different semantic regions, enhancing the understanding of the image content. Mathematically, semantic segmentation can be represented as a mapping function f from the input image I to the output segmented image O, expressed as: $f : I \rightarrow O$, where $O_{ij} = f(I_{ij})$ denotes the semantic category assigned to the pixel I_{ij}.

Semantic segmentation has numerous applications, including crop monitoring [PTB19, MRCW+19, YLL20], medical image analysis [JGR+18, KGLK21, KNN21, QYA+23], autonomous driving [FHSR+20, SGAR+18, HUAM+19, SDG23], and industrial automation [ULPG22, Qia19, EAN20, SZC+23]:

- **Crop Monitoring:** High-resolution remote sensing images from drones or satellites monitor crop growth and changes, evaluate crop health, and guide agricultural production.

- **Medical Image Analysis:** Semantic segmentation is widely used in medical imaging, with significant clinical importance. It can automatically identify and segment tumor tissues in CT and MRI images, aiding in precise tumor localization and diagnosis. It also helps identify and segment various organs, such as the heart, lungs, and kidneys, supporting clinical diagnosis and surgical planning.

- **Autonomous Driving:** In autonomous driving, semantic segmentation systems identify pedestrians, vehicles, traffic signals, road edges, and more, adjusting the vehicle's speed and route. This ensures accurate lane keeping and effective path planning, especially on complex urban roads and highways.

- **Industrial Automation:** Semantic segmentation technology detects surface defects on products during manufacturing, such as cracks, flaws, and foreign matter, facilitating automated quality control.

2.2 Metrics of Semantic Segmentation

Commonly used evaluation metrics in semantic segmentation tasks include:

- **IOU.** IOU (Intersection over Union) is a metric used to measure the similarity between predicted results and true labels in semantic segmentation tasks. It calculates the ratio of the intersection to the union of the predicted and true sets, serving as an indicator of their overlap. In binary semantic segmentation, the IOU for foreground is IOU = TP / (TP + FP + FN), where TP (True Positive) denotes the amount of pixels whose class is correctly predicted, FP (False Positive) denotes incorrectly segmented parts predicted as a certain label, and FN (False Negative) denotes parts incorrectly predicted as background which are not actually background. In multi-class semantic segmentation tasks, the IoU is calculated for each category to measure how well the predicted segmentation matches the ground truth for each specific class. To provide an overall evaluation metric that considers all categories, the Mean Intersection over Union (mIoU) is used. The mIoU is the average of the IoUs for all the categories.

- **Pixel Accuracy.** In binary sementic segmentation, overall pixel accuracy refers to the percentage of correctly predicted pixels out of all pixels, which is calculated as Acc = (TP + TN) / (TP + TN + FP + FN), where TN (True Negative) denotes correctly predicted background parts in the prediction. It is a straightforward metric but may not accurately evaluate model quality when the category distribution is uneven. Therefore, it is often used in conjunction with other evaluation metrics. If it is a multi-class segmentation task, the mean pixel accuracy (mPA) is used as an evaluation indicator. This is done by calculating how many pixels of each class are correctly classified and then averaging them.

- **Precision.** Precision calculates the proportion of correctly predicted pixels among the pixels predicted by the model as positive examples, that is, precision = TP / (TP + FP). In semantic segmentation, it indicates the model's ability to accurately identify positive samples.

- **Sensitivity.** Sensitivity, also known as recall or true positive rate, measures the proportion of correctly classified positive examples among all actual positive examples, which is TP / (TP + FN).

2.3 Evolution of Semantic Segmentation

2.3.1 Traditional methods

Traditional semantic segmentation methods primarily rely on manually designed features crafted by domain experts and traditional machine learning algorithms. In these methods, experts extract various features from images, encompassing low-level features (such as color, texture, and shape features extracted directly from pixels), local features (including key point features [ECC02] and local binary patterns (LBP) [Pie10, LS08] derived from local image regions), and global features (such as color histograms, gradient orientation histograms (HOG) [Tom12, NB07], and Fourier transforms [Sne95, SG12] that characterize the overall image properties). These designed features serve as inputs to traditional machine learning algorithms like k-means clustering [DMC15, WLC07] and random forests [SCZ08, LLV16] for achieving semantic segmentation. Traditional semantic segmentation methods can be categorized as follows.

Threshold based segmentation method

Threshold-based methods [RRDR09, NMI10] are mostly used for grayscale images, including global thresholding, local thresholding, adaptive thresholding, and multi-thresholding. In the global thresholding method, a single threshold separates the foreground from the background in binary classification, that is, when the image pixel value is greater than the threshold, it is considered to be the foreground, otherwise it is the background. However, it is not easy to find a suitable threshold. Otsu's algorithm is the most popular method for finding an appropriate threshold [YM12]. The local thresholding method determines the threshold based on the local characteristics of the image. [LSP+18] mentions a method of calculating the threshold within the neighborhood of each pixel. The adaptive thresholding method combines the statistical characteristics of the image and adaptively selects the threshold. Otsu's algorithm is also an adaptive thresholding method. When the image becomes complex, a single threshold cannot achieve the segmentation goal, and [RM03] the multi-thresholding method is widely used for image segmentation with higher requirements.

Edge detection based methods

Edge detection algorithms segment images into distinct regions, effectively delineating different objects within an image. These algorithms operate under the assumption that grayscale values change abruptly at edges, distinguishing them from smoother changes in surrounding areas. Edge detection algorithms are categorized into first-order operators and second-order operators based on the mathematical operations used. First-order operators utilize first-order differentials to calculate edge strength. They determine the local gradient direction and locate edges by identifying the maximum gradient modulus in that direction. Examples include the Canny operator [DWY13], Sobel operator [KVB88], and Prewitt operator [ZYCL19]. Second-order operators rely on second-order differentials to locate edges by identifying zero-crossings in the second derivative of the image, such as the Laplacian of Gaussian (LOG) operator [UM90] and Laplacian operator [CSS10]. Edge detection algorithms are highly sensitive to noise, necessitating post-processing techniques to achieve accurate semantic segmentation.

Region based segmentation methods

Region-based segmentation algorithms partition images into distinct regions based on specific shared attributes, such as color or intensity. Two common approaches are region growing [PT01] and split-and-merge [LS01]. Region growing begins with multiple seed pixels and iteratively incorporates connected pixels that are similar to the seeds, expanding the region until distinct segmented regions are formed. In contrast, region splitting and merging starts with the entire image and progressively divides it into numerous sub-regions. It then merges smaller regions with similar attributes. This method is effective when there is high contrast between objects and backgrounds. However, it struggles to achieve accurate segmentation results when grayscale values are similar between objects and backgrounds.

Superpixel segmentation is a specialized region-based method that partitions images into irregular regions known as superpixels [RM03]. These regions are composed of adjacent pixels sharing similar characteristics such as color, texture, and brightness. Superpixel segmentation transforms the pixel-level representation of an image into a compact region-level graph, abstracting pixel information and aggregating conceptually related pixels. This process replaces numerous pixels with a smaller number of superpix-

els, significantly reducing the complexity of subsequent image processing tasks. Superpixel segmentation typically serves as a preprocessing step for segmentation algorithms. Two widely used algorithms include Simple Linear Iterative Clustering (SLIC) [CO16] and Quickshift [VS08]. SLIC adapts the K-means clustering algorithm by constructing a distance metric in the color space and utilizing a five-dimensional feature vector of pixel coordinates to generate uniform and compact superpixels. However, this method is sensitive to parameter settings. In contrast, Quickshift leverages pixel similarity and density information for segmentation without requiring a predefined number of segments. It constructs a density estimation map based on pixel similarities and updates pixel positions to move towards higher-density regions, thereby achieving image segmentation. Region-based segmentation harnesses pixel similarity and connectivity to simplify image representation, reduce data complexity, and enhance computational efficiency.

Energy minimization based segmentation methods

The fundamental concept of energy minimization [Nik11] involves adjusting the state of a system to achieve its most stable or optimal state by minimizing its energy. In image segmentation, this principle is applied systematically to evaluate and optimize segmentation quality, posing it as an energy minimization problem. The energy minimization framework formalizes image segmentation as an optimization task aimed at finding an energy function that reaches its minimum value. This approach is akin to maximizing a posteriori estimation in Markov random fields (MRF) [BKR11]. Energy minimization algorithms typically involve designing and optimizing an energy function, also known as an objective function. When formulating the energy function, two main components are considered: the data term and the regularization term. The data term ensures that the segmentation aligns closely with the original image data, while the regularization term promotes label consistency among neighboring pixels, thereby enhancing segmentation coherence and boundary smoothness. For a simple binary image segmentation problem, the energy function can be defined as follows:

$$E(O) = \sum_{p}(-logP(O_p|I_p)) + \lambda \sum_{(p,q)\in N} exp(-\beta\|I_p - I_q\|^2)\delta(O_p \neq O_q) \tag{2.1}$$

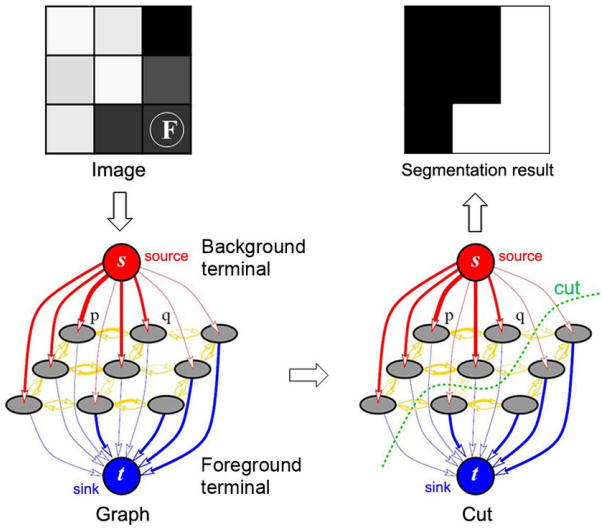


Figure 2.1: Example graph cut segmentation for a simple 3 × 3 image. The pixel labeled "F" is a hard constraint. The n-links are yellow edges while the t-links are red and blue edges.The weight (cost) of each edge is reflected by its thickness. From [XYZ+17]

Where, $-logP(O_p|I_p)$ denotes the negative logarithmic probability of assigning pixel p to label O_p, $exp(-\beta\|I_p - I_q\|^2)\delta(O_p \neq O_q)$ represents the cost associated with assigning different labels to adjacent pixels p and q, and weight $exp(-\beta\|I_p - I_q\|^2)$ signifies the similarity between pixels (higher weights indicate greater dissimilarity between pixel colors).

The optimization of the energy function involves minimizing the objective function. Since most energy functions are non-convex and exhibit multiple minima, the minimization process is challenging. The design of both the optimization method and the objective function are crucial. Commonly used optimization methods include simulated annealing [BT93], gradient descent [Rud16], Newton's method [Kel03], dynamic programming [BD15], variational methods [Rek12], and the graph cut method [FD05], with the graph cut method being particularly prevalent.

Graph cut integrates the image segmentation problem with the minimum cut problem of a graph. In Figure 2.1, F denotes foreground pixels, and the image is represented as an undirected graph $G(V, E)$, where V is the vertex set and E is the edge set. Vertices and edges are categorized

into two types. The first type of vertices corresponds to each pixel in the graph. Edges between adjacent pixels are termed n-links. The second type includes vertices s and t, collectively known as terminal vertices. In binary image segmentation, s and t represent background and foreground, respectively. Each pixel vertex is connected to these terminal vertices, forming the second type of edge, termed t-links. The thickness of edges in the graph corresponds to their respective weights, as illustrated by the smallest cut found in the image. A cut represents a set of edges. Disconnecting these edges prevents a path from s to t, ensuring each pixel connects exclusively to either s or t. T-links facilitate foreground-background classification and separation, while n-links delineate spatial boundaries. The minimum cut occurs when the sum of all cut edge weights is minimized, signifying completion of image segmentation. Two graph cut algorithms, Alpha-beta swap and Alpha-expansion [BDVJ+16, VKR08, WHMH14, YBS10], utilize these principles to accomplish image segmentation.

Machine learning based segmentation methods

To determine if an egg has gone bad, we hold it in both hands in a cylindrical shape and inspect it under sunlight or light. A fresh egg appears reddish, translucent, with a clear yolk. Conversely, a bad egg appears dim, opaque, and may have spots. This method works because our life experiences have familiarized us with similar situations. Similarly, in mathematics, solving numerous problems builds a repository of experiences. This allows us to draw inferences and effectively predict solutions when encountering similar problems in the future.

Machine learning mirrors our learning process. It enhances algorithm performance by iteratively learning from data and applying acquired patterns or rules to produce desired outcomes for new data. Machine learning is categorized into supervised, unsupervised, and semi-supervised learning based on data labeling. Fundamental components include data, models, and algorithms. Data forms the foundation of machine learning, encompassing training, validation, and test datasets. Training data educates the model, validation data refines hyperparameters and evaluates performance, while test data assesses the model's generalization capability. The model constitutes the heart of machine learning, deriving specific outputs from learned data patterns. Learning occurs through algorithms, where selecting an appropriate algorithm is critical to finding the optimal model. Algorithms, such as gradient descent, stochastic gradient descent, and mini-

batch gradient descent, iteratively adjust model parameters to achieve optimal results. In segmentation tasks, common machine learning classifiers include clustering [CA79, WFMF02, NMI10, DMC15, LLJ+19], Markov random fields [KP06, Li09, KZ+12, CZYW17], random forests [GEW06, PBK16, KN19], random ferns [BDVJ+16, WH21], gradient boosted trees [SJPH15, OSYO17].

Clustering based image segmentation. Clustering is an unsupervised learning technique used for classification without prior knowledge of pixel categories. It iteratively groups pixels into clusters based on the distances between their feature descriptions (such as color, texture, brightness, etc.). This process aims to maximize the similarity within each cluster while minimizing differences between clusters. Subsequently, the pixels are assigned different categories based on the clustering results to achieve segmentation. Post-processing steps, such as noise removal and merging adjacent regions, are often applied to refine the segmentation outcome. Popular clustering algorithms include K-means, spectral clustering [VL07], and DBSCAN [KRA+14]. K-means, in particular, is widely applied in image segmentation. For instance, [ZP22] utilized particle swarm and K-means clustering in the YCbCr color space to segment agricultural products. In another study [RLL+21], the K-means method automatically clustered pixels, followed by post-processing using mathematical morphology to segment the heart in chest CT images.

Markov Random Field based image segmentation. A random field consists of multiple positions, each assigned a value based on a specific distribution, thereby forming the entire field. Markov random field (MRF) is a specialized type of random field that posits each position's value depends solely on its adjacent positions. When represented as an undirected graph, nodes in the graph correspond to random variables, and edges between nodes signify relationships between their associated random variables. An image can be interpreted as a random field, where each pixel's position represents a node. Each node holds a random variable denoting the pixel's category (e.g., foreground or background). Adjacent pixels often exhibit correlated categories, implying neighboring pixels are likely to belong to the same category. In MRF, it is assumed that a pixel's category is influenced only by its directly adjacent nodes. [YWZ+21] provides a mathematical framework for Markov-based semantic segmentation and applies it to semantically segment high-resolution remote sensing images.

Random forest based image segmentation. Random forest is an ensemble of decision trees, consisting of multiple decision trees. Semantic segmentation based on random forest is defined as follows. Given an image I, whose size is $M \times N$, each pixel $I_{i,j}$ contains a feature vector $X_{i,j}$, which is the feature of the pixel, assuming that D is the dimension of the feature. The training process of random forest involves the construction of T decision trees. First, N samples (N is the size of the original dataset) are extracted from the original dataset using the bootstrap sampling method to form the training dataset S_t of decision tree t. Each sample consists of a feature vector X and a corresponding semantic label Y. Next, a decision tree is constructed. For each node of the tree, d features are randomly selected from all features, and the best features and partition points are selected by the Gini coefficient or information gain. After training, when predicting on the test set, the feature vector of each pixel is input into the trained random forest to obtain the corresponding semantic label, and then the voting method or averaging method is used to obtain the prediction result. [KHS10, MWH16, LLV16, SCZ08] Semantic segmentation tasks in specific fields are implemented based on random forest.

Random Fern based image segmentation. Random Fern is a variant of Random Forest, first proposed by [OCLF09], which has demonstrated strong performance in various semantic segmentation tasks [BDVJ+16, JSX+17, WH21]. Unlike Random Forest, which learns directly from posterior probability, Random Fern learns through the category-conditional probability density distribution. Additionally, in Random Forest, the features compared by each node for each input sample are randomly selected, resulting in different feature orders between samples. Conversely, in the Random Fern algorithm, the order of compared features remains the same across samples. In terms of training time, Random Forest's complexity grows exponentially with the depth of the tree, whereas Random Fern's training time increases linearly with the tree's depth, as illustrated in the accompanying Figure 2.2.

The principle of the Random Fern algorithm is as follows: In semantic segmentation, the category of a pixel can be determined by using features to calculate the category that corresponds to the maximum probability in the probability distribution of each category, as shown in the following formula.

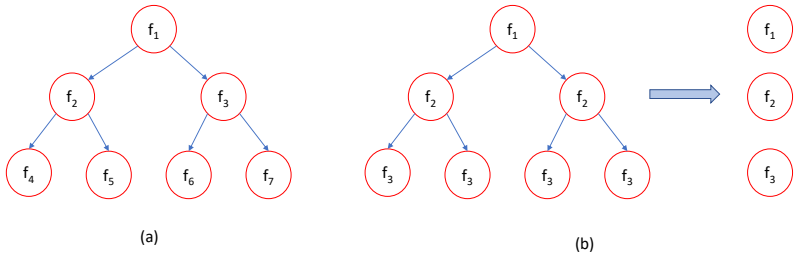


Figure 2.2: Random fern vs random forest. (a)Tree. Random Forests consist of multiple hierarchical trees. Due to the binary structure of these trees and the need to evaluate numerous potential splits during training, the number of nodes and the computational effort required increase exponentially with the depth of the tree. (b)Fern. Multiple ferns form Random Ferns. Fern performs the same test on all nodes at the same level, resulting in the evaluation of the same feature being independent of the path to a specific node, which is equivalent to no longer having a hierarchical structure.

$$C = \arg\max_k P(C_k \mid f_1, f_2, \ldots, f_N) \tag{2.2}$$

Where $\{f_1, f_2, ..., f_N\}$ are the binary features extracted from the pixel-centered patch, $C_k, k \in \{1, ..., H\}$ is a set of predefined classes. Under Bayes' rule, the above formula can be approximated as:

$$C \approx \arg\max_k P(f_1, f_2, \ldots, f_N \mid C_k) P(C_k) \tag{2.3}$$

However, for a complex input sample, calculating $P(f_1, f_2, \ldots, f_N \mid C_k)$ is very difficult.

Although Naive Bayes assumes that all features are completely independent of each other, as shown in the formula:

$$P(f_1, f_2, \ldots, f_N \mid C_k) P(C_k) = \prod_{j=1}^{N} P(f_j \mid C_k) \tag{2.4}$$

However, this independence assumption is often challenging to maintain in most cases. In practice, many features are frequently correlated. Therefore, let's further assume that the N features can be divided into L independent smaller feature sets $F = \{F_1, F_2, ..., F_L\}$ of size S. The small

feature set $F_l = \{f_{l_1}, f_{l_2}, ..., f_{l_S}\}$is called a fern. Then, the above formula can be written as:

$$P(f_1, f_2, \ldots, f_N \mid C_k)P(C_k) = \prod_{l=1}^{L} P(F_l \mid C_k) \tag{2.5}$$

To prevent computational overflow, which can occur when L is large, we use the monotone logarithm function. The score $s_{C_k}$of class C_k for a pixel is then defined as:

$$s_{C_k} = log(\prod_{l=1}^{L} P(F_l \mid C_k)) = \sum_{l=1}^{L} log(P(F_l|C_k)) \tag{2.6}$$

A straightforward approach to classification is to use the MAP estimator $\hat{C}$ for each pixel:

$$\hat{C} = \arg\max_k P(C_k \mid F_1, F_2, \ldots, F_L) \approx \arg\max_k s_{C_k} \tag{2.7}$$

The training process for random ferns involves determining the conditional class probabilities $P(F_l|C_k)$ for each fern F_l and class C_k by counting the occurrences of fern values for each class within the training subset of the ground truth data. The parameters for each fern can be easily estimated independently:

$$P(F_l|C_k) = \frac{N_{l,C_k}}{N_{C_k}} \tag{2.8}$$

Where N_{l,C_k} represents the count of occurrences of F_l for pixels belonging to class C_k, while N_{C_k} denotes the total number of pixels of class C_k used for training. However, during this phase, not all 2^S possible fern realizations for a given fern F_l might be encountered, even though these realizations could still occur when evaluating ferns on another image. Consequently, if $P(F_l|C_k) = 0$, it nullifies the product $\prod_{l=1}^{L} P(F_l \mid C_k)$ and leads to an error when computing log0, resulting in poor performance for this scheme. To solve this problem, add a smooth during calculation, as follows:

$$P(F_l|C_k) = \frac{N_{l,C_k} + smooth}{N_{C_k} + 2^S * smooth} \tag{2.9}$$

2.3.2 Deep learning based segmentation methods

Unlike traditional machine learning, deep learning does not require manually designing feature extractors to represent raw data. Instead, it can automatically discover the features needed to complete specific tasks directly from raw data. The term "deep learning" originated from [Dec86]. However, due to early hardware limitations, deep learning did not initially attract widespread attention from researchers. In 2000, deep learning was introduced into artificial neural networks (NNs) for machine learning. Neural networks with two or more hidden layers are typically referred to as deep neural networks. This multi-layered structure can capture complex nonlinear relationships and provide a higher level of abstraction for underlying features. In 2012, Hinton achieved remarkable results in the ImageNet image classification competition using a convolutional neural network (CNN) learning model. This success marked the beginning of the deep learning era, drawing significant attention from researchers and establishing deep learning as a major field of study.

CNNs

Convolutional neural networks (CNNs) form the cornerstone of deep learning applications in computer vision. In 1989, Yann LeCun introduced the LeNet-5 [LBBH98], a pioneering CNN model. LeNet-5 consists of an input layer, convolutional layers, pooling layers, fully connected layers, and an output layer, as illustrated Figure 2.3. This model achieved significant success in the task of handwritten digit recognition, marking the first substantial demonstration of CNNs' potential in image processing. The foundational structure of LeNet-5 continues to underpin modern CNN architectures.

Input layer. The input layer receives the original image data. Unlike images perceived by the human eye, computer images are represented as matrices composed of pixel values.

Convolutional layer. The convolutional layer is the core of a CNN, responsible for extracting local features from the input data through convolution operations. Each convolutional layer contains multiple convolution kernels that slide across and scan the entire image, producing different feature maps, as illustrated in Figure 2.4.

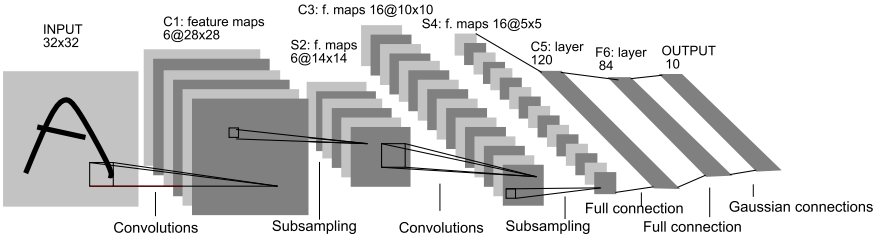


Figure 2.3: Architecture of LeNet-5, a convolutional neural network. Each plane is a feature map. From[LBBH98]

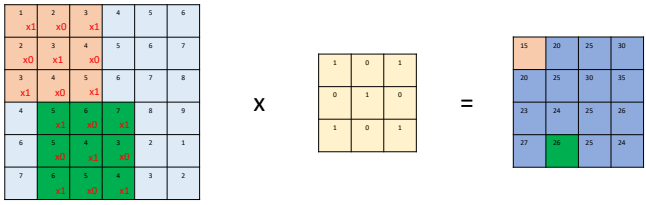


Figure 2.4: Convolution.

Pooling layer. The pooling layer, also known as the downsampling layer, reduces the size of the feature map while preserving essential features. This operation decreases the dimensionality of the data and reduces computational complexity.

Fully connected layer. The fully connected layer is typically located at the end of the network, where each neuron is connected to all neurons in the preceding layer. This layer integrates the features extracted by the preceding convolutional and pooling layers to perform the final classification.

Output layer. The output layer produces the final prediction result. For classification tasks, it typically employs a sigmoid or softmax function to generate the probabilities for each category.

CNN-based semantic segmentation

CNNs have been proven effective in image classification and object detection, and they are widely used. However, the fully connected layers in

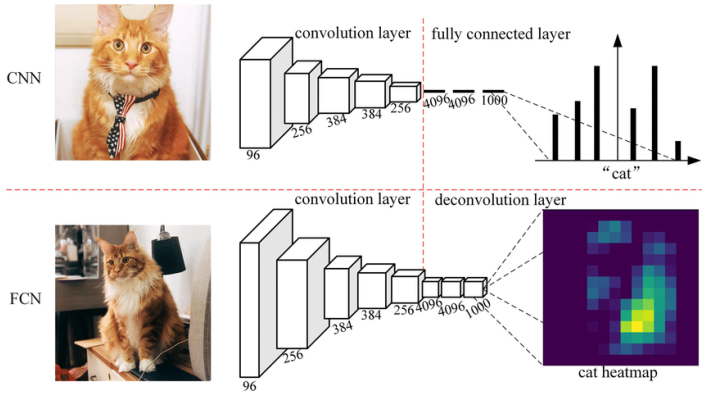


Figure 2.5: The difference between the CNN and FCN(the transforming of fully connected layers into convolutional layers by an FCN enables a classification net to output a heatmap). From[YMW+20].

these CNNs are one-dimensional, which limits them to identifying the category of the entire image. Image segmentation, on the other hand, requires pixel-level classification. To adapt CNNs for semantic segmentation, modifications are usually made to achieve pixel-level classification. The fully convolutional network (FCN) forms the basis of CNN-based semantic segmentation algorithms.

As illustrated in Figure 2.5, the FCN achieves pixel-level label prediction by replacing traditional fully connected layers with convolutional layers, thus retaining spatial information. The FCN restores the feature map to the original image size using deconvolution operations. It combines shallow and deep features through skip connections, allowing information to jump between different layers, as shown in Figure 2.6. This connection operation is typically achieved through a 'add' operation, where the input and output are added together.

By using an end-to-end, pixel-to-pixel training approach, FCNs significantly improve the computational efficiency and prediction performance of semantic segmentation. FCNs represent a milestone in the application of deep learning to semantic segmentation, laying the foundation for the development of subsequent semantic segmentation algorithms.

Although fully convolutional networks (FCNs) are powerful, they have some shortcomings. FCNs extract image features through multiple downsampling steps, which significantly reduce the resolution of the output fea-

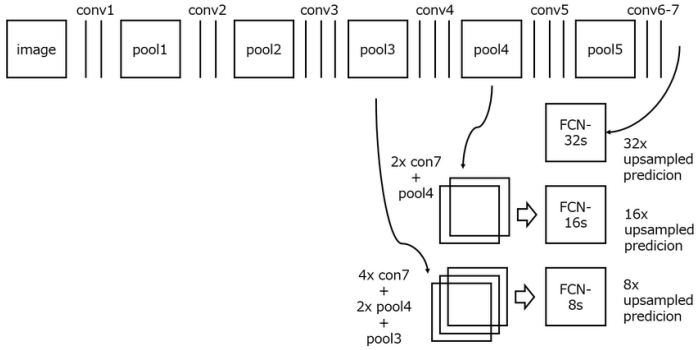


Figure 2.6: Skip connections combine information from the lower and higher layers. FCN-32s: upsamples stride 32 predictions back to pixels in a single step, FCN-16s: combines predictions from both the final layer and the pool4 layer, at stride 16, FCN-8s: additional predictions from pool3 layer. From [LIIU16]

ture map compared to the input image. While skip connections are used to fuse shallow and deep features in an attempt to restore resolution, this incomplete restoration often results in insufficient fineness in segmentation results, particularly at boundary details.

To address this, the U-Net network model was proposed, as illustrated in Figure 2.7. U-Net employs a completely symmetrical encoder-decoder structure. The encoder processes the image to extract multi-level features through gradual downsampling. The decoder then restores the resolution of these high-level semantic features to achieve pixel-level classification. This restoration process ensures that the features are more accurate.

U-Net also utilizes a large number of skip connections between the symmetrical layers of the encoder and decoder. These connections directly transfer high-resolution feature maps from the encoder to the corresponding layers in the decoder, allowing the decoder to leverage spatial detail features from the encoder and achieve multi-level feature fusion. This approach not only helps restore detail features but also reduces information loss during upsampling. By combining local and global features, U-Net significantly improves segmentation accuracy.

Subsequent researchers have developed many variants based on U-Net to address specific semantic segmentation tasks. For example, [ÇAL+16]

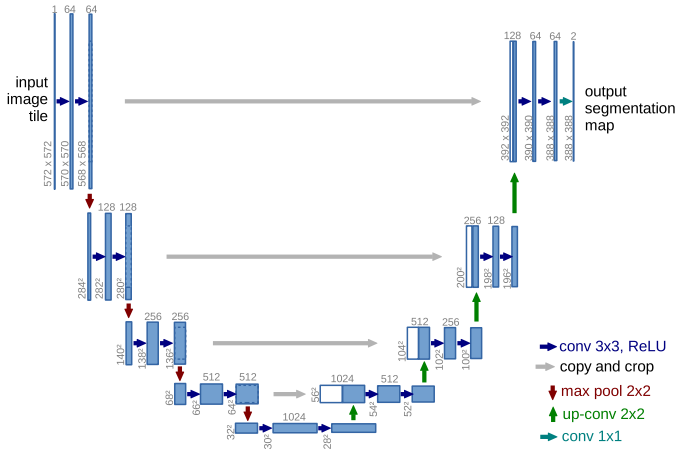


Figure 2.7: U-Net architecture. From [RFB15a]

presented a variant that retains the core structure of U-Net but replaces 2D operations with 3D operations, such as 3D convolution, 3D pooling, and 3D upconvolutions, to segment 3D images [LBD+20, BLJ+19]. In [LBJ+18] a deep CNN network is proposed to segment 2D CT image slices of the bladder using manual delineation on adjacent slices as prior knowledge. Inspired by residual learning [HZRS16] and U-Net, [ZLW18] proposed the Residual U-Net, which incorporates ResNet as the encoder and adds skip connections in the encoder. These connections, combined with the skip connections in U-Net, enhance information propagation during both forward and backward calculations. Additionally, [AHY+18] combined the advantages of U-Net, residual networks, and RCNN to propose RU-Net and R2U-Net. Attention U-Net was another notable variant that integrated the attention mechanism [VSP+17] into the U-Net framework. This enhancement leverages U-Net's powerful ability to restore details while incorporating the global context perception capability of the attention mechanism. By adaptively adjusting the weights of different positions in the feature map, the network focuses on important areas and disregards irrelevant backgrounds, further improving segmentation accuracy, especially in complex scenes and tasks with multi-scale features. There are numerous other U-Net variations, including Inception U-Net, U-Net++ [ZRSTL18], V-Net [MNA16], TransUnet [CLY+21], and Swin-Unet [CWC+22a], which incorporates transformers [VSP+17].

In addition to U-Net and its variants, numerous other modifications have been made to fully convolutional networks (FCNs) within CNN-based semantic segmentation frameworks. For example, the DeepLab series [CPK+14, CPK+17, CPSA17, CZP+18] introduces several significant innovations. DeepLabv1 [CPK+14], the first model in the series, incorporates dilated convolutions into FCNs to expand the receptive field without increasing computational complexity. Building on this, DeepLabv2 [CPK+17] proposes atrous spatial pyramid pooling (ASPP) [CCZ+20] [GVZI+21], applying dilated convolutions to extract features from multiple scales at different sampling rates. DeepLabv3 [CPSA17] further enhances ASPP by including global average pooling and introducing additional sampling rates. Finally, DeepLabv3+ [CZP+18] adopts an encoder-decoder structure based on DeepLabv3 to improve the boundary details of the segmentation results. Deep learning predictions are not completely trustworthy, so [LLKDV22, SDVD+23] provide an uncertainty assessment of the model.

Vision transformer architecture (ViT)

As research on Transformers [VSP+17] has advanced in natural language processing, their application has expanded into the field of computer vision, where they have gained significant popularity. Vision Transformer (ViT) [Dos20] was introduced for image classification, demonstrating its effectiveness in computer vision tasks. Researchers have since applied Transformers to semantic segmentation, leading to models like SegFormer [XWY+21], Segment Anything [KMR+23] and SAM2[RGH+24]. While Transformers have made notable progress in the vision domain and are increasingly used in image segmentation, their computational cost remains relatively high. In addition, ViTs usually exhibit weaker inductive biases [SKZ+21], and they have to learn more from training data, resulting in heavy reliance on massive datasets for large-scale training. Compared with ViTs, CNNs are easier to optimize. In the field of semantic segmentation, CNNs remains a largely adopted and reliable choice. Our work focuses on this architecture. Validating the lessons drawn from our experiments on other architectures is left for future work.

2.4 The challenge of insufficient labeled data

The CNN-based deep learning method currently leads the task of semantic segmentation in computer vision. A semantic segmentation model with good generalization ability requires a large amount of labeled data for training. Labeled data forms the basis of supervised learning, providing the model with the correspondence between feature information and category labels. This enables the model to learn and identify the essential characteristics of objects, allowing it to make accurate predictions when encountering new data. The quality and quantity of labeled data directly affect the model's performance and generalization ability, making it crucial for effective training.

Without labeled data, the model lacks fixed information and knowledge about the world. For example, an autonomous driving model relies on labeling to recognize roads, vehicles, and pedestrians. Data labeling helps us understand the model's decision-making basis and prediction logic, enhancing trust in AI systems and promoting their widespread application across various fields.

Despite its vital role, data labeling faces several challenges. It is a time-consuming and labor-intensive process, particularly with large-scale data. The quality of labeling directly impacts the model's accuracy. Insufficient labeled data prevents the model from learning effective patterns during training, leading to underfitting. This results in the model performing well on training data but poorly on test data or in real-world applications. Additionally, with less data, the model is more susceptible to noise and outliers, showing high variance and sensitivity to training data. This variability leads to instability, as the model's performance can vary significantly across different training sets.

Insufficient labeled data remains a significant challenge that needs to be addressed in many deep learning tasks.

2.5 Current strategies for addressing insufficient labeled data

To tackle the issue of insufficient labeled data, researchers have explored various approaches such as data augmentation [SK19, ZZK+20, CLZJ20], transfer learning [WKW16, NSZ20], semi-supervised learning [VEH20, ZG22, DSYA24], self-supervised learning [HMKS19, MM20, GCZ+24], gen-

erative models [NMT+18, Lam21, CLZZ23], and active learning [DG22, MLG+18, XFC+20].

2.5.1 Data augmentation

Data augmentation involves generating new images by making minor modifications to existing ones. For image segmentation tasks, overcoming challenges such as viewpoint changes, lighting variations, occlusions, background clutter, and scale differences is crucial. By integrating these transformations into the dataset, data augmentation enables models to effectively handle such challenges and diversifies the training samples, thereby enhancing the performance of deep learning models. Common techniques used in data augmentation include:

- **Rotation** Randomly rotate the image by a specified angle (e.g., ±10 degrees, ±30 degrees) to improve the model's capability to generalize across various viewing angles.
- **Cropping** Randomly cropping parts of the image aids the model in achieving accurate segmentation, particularly when dealing with partial occlusion or visibility.
- **Flip** Horizontal or vertical flipping of an image enhances data diversity, particularly beneficial when objects exhibit symmetry.
- **Scaling** Randomly scaling images, such as zooming in or out, helps the model adapt to objects of varying sizes.
- **Translation** Randomly translate the image horizontally and vertically to improve the model's resilience to changes in object position.
- **Noise** Introducing random noise to the image, such as Gaussian or salt and pepper noise, can bolster the model's robustness against noise and subtle variations.
- **Blurring** Applying blurring techniques like Gaussian blur can enhance the model's performance across various focal lengths and clarity conditions.
- **Contrast Adjustment** Modifying the image's contrast helps the model accommodate different lighting conditions during image capture.

Data augmentation is crucial in deep learning. By employing diverse augmentation techniques during training, such as flipping, rotation, scaling, noise addition, blurring, and contrast adjustment, the dataset size is increased and its diversity enhanced. This enables the model to learn more robust features, enhancing its adaptability and resilience in real-world scenarios. Additionally, data augmentation helps mitigate overfitting risks and reduces annotation expenses.

2.5.2 Transfer learning

Transfer learning is a machine learning technique used to address the issue of insufficient labeled data by leveraging knowledge transfer [TS10] across different domains. Similar to how humans apply knowledge learned in one subject to another subject for improved learning, machines can enhance their learning capabilities in a specific domain by transferring information from related fields. Transfer learning aims to boost learning performance and reduce the dependency on labeled data in the target task by applying knowledge and experience gained from one source task or domain to another as shown in Figure 2.8. For instance, a model pre-trained on a large dataset like ImageNet [DDS+09] can be adapted for fine-grained image classification tasks, such as identifying different flower species. By fine-tuning the pre-trained model, accurate classification results can be achieved even with a smaller dataset.

Transfer learning begins by selecting a source task related to the target task. The source task should have sufficient data volume and complexity to enable the model to learn meaningful features. The model is then pre-trained on this source task. To leverage the learned feature representation effectively, the initial layers of the trained model are typically frozen, while the latter layers are fine-tuned using a smaller learning rate on the target task's data. This approach is generally effective because many image datasets share common low-level spatial features that are best learned from extensive data.

With the proliferation of big data repositories and the scarcity of labeled data for specific tasks, training a model from scratch can lead to overfitting. Transfer learning offers an appealing solution by utilizing existing datasets that are related, though not entirely identical, to the target domain. Transfer learning has found applications across various domains. However, it's crucial to note that successful transfer relies on a meaningful connection between the source and target tasks. If the tasks lack common ground,

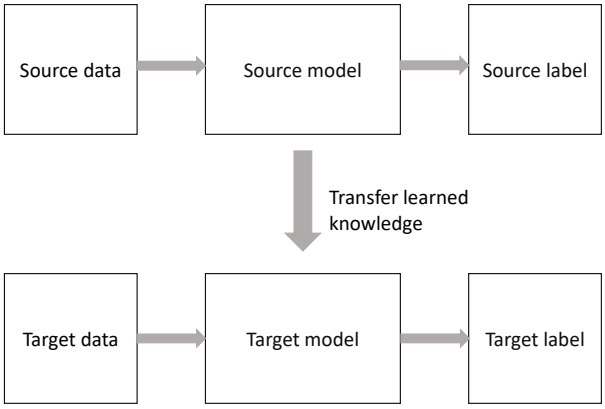


Figure 2.8: Transfer learning.

knowledge transfer may fail. Even when tasks are related, dataset variations can pose challenges. Addressing domain shifts between different data domains is currently a prominent research focus.

2.5.3 Semi-supervised learning

In many practical applications, obtaining labeled data can be very expensive or time-consuming, whereas unlabeled data is usually easier to acquire. Semi-supervised learning leverages a large amount of unlabeled data to enhance model performance without significantly increasing labeling costs. This machine learning method addresses the problem of insufficient labeled data by operating between the extremes of unsupervised learning (with no labeled training data) and supervised learning (with fully labeled training data). In semi-supervised learning, the training dataset includes both labeled and unlabeled samples. By utilizing both types of data, semi-supervised learning aims to build a classifier that outperforms one trained solely on labeled data.

The core idea of semi-supervised learning is to utilize the structural information in unlabeled data to enhance the model's generalization ability. While unlabeled data lacks explicit labels, it provides additional samples that help the model better understand the data's distribution and structure.

Leveraging this information allows the model to grasp the underlying patterns and rules within the data, thus improving its performance on unseen data. Additionally, a model trained on a small amount of labeled data can generate pseudo-labels for the unlabeled data. These pseudo-labels, although not entirely accurate, can be used for further training. This approach helps regularize the model, preventing it from overfitting to the limited labeled data and enabling it to generalize better to new, unseen data.

Semi-supervised learning methods can be categorized into self-training [AFP+22, XLHL20, WSCM20], co-training [BM98, PEPD20, XYY+20], and consistency regularization [XLBY22, KMK+22, FKDS23], based on how they utilize unlabeled data.

The core idea of self-training is to use unlabeled datasets to expand the labeled datasets. The training process involves the following steps:

- Train a teacher model using the initial labeled data.
- Use this model to make predictions on the unlabeled data, selecting samples with high prediction confidence and adding their predicted labels to the labeled dataset.
- Retrain the model with the expanded labeled dataset to obtain a student model, which then becomes the next teacher model.

These steps are repeated until convergence. Although simple and easy to implement, this method can suffer when the initial labeled samples are insufficient or unrepresentative of the entire data distribution, leading to a decision boundary that deviates from the true boundary and a teacher model with low generalization ability.

Similar to how we can better understand things from different perspectives, co-training assumes that data can be classified from various views, allowing for the training of different classifiers from these perspectives. This method involves dividing the feature set into distinct subsets and training multiple models based on these subsets. These models then predict labels for the unlabeled samples, and samples with high-confidence predictions are added to the training set. By leveraging multi-view features, co-training enhances the robustness of the model. However, this approach requires that the features can be reasonably divided into independent subsets.

Consistency regularization is based on the smoothness and clustering assumptions: data points with different labels are separated by low-density

regions, and similar data points yield similar outputs. Therefore, when a small perturbation is applied to an unlabeled data point, its prediction should not change significantly, ensuring output consistency.

2.5.4 Self-supervised learning

In recent years, self-supervised learning has garnered significant attention in the field of computer vision. This approach reduces the dependence on large amounts of labeled data by extracting useful features from unlabeled data. As an unsupervised machine learning method, it does not rely on any manually labeled data. Instead, it generates its own supervised information from large-scale unsupervised data through carefully designed auxiliary tasks, and the network is trained using this constructed supervised information. For example, in image processing tasks, an image can be randomly rotated, and the model is then tasked with predicting the rotation angle. By solving such tasks, the model learns valuable representations that can be used for downstream tasks.

Self-supervised tasks are crucial to self-supervised learning, as they generate pseudo-labels for training by performing specific operations on the data. Common self-supervised tasks in the field of computer vision include:

- **Image inpainting.** Image inpainting [PKD+16] randomly blocks a portion of the image and tasks the model with predicting the pixel values within the blocked area.
- **Rotation prediction.** Rotation prediction [GSK18] involves randomly rotating the image by a specified angle and tasking the model with predicting the rotation angle.
- **Image colorization.** Image colorization [ZIE16] involves converting a grayscale image into a color image or vice versa, and then tasking the model with predicting the original information.
- **Jigsaw puzzle solving.** Jigsaw puzzles [NF16] entail dividing an image into several pieces, shuffling these pieces randomly, and tasking the model with predicting the correct order of the pieces.
- **Image reconstruction.** Image reconstruction [HCX+22] involves encoding the image into a low-dimensional representation using an encoder and then decoding it through a decoder to reconstruct the orig-

inal image. The model is trained by minimizing the discrepancy between the reconstructed image and the original image.

- **Contrastive learning.** Contrastive learning [CKNH20] generates positive sample pairs by applying different data augmentations to the same image and creates negative sample pairs from different images. The model's objective is to bring the feature representations of positive sample pairs closer together and push the feature representations of negative sample pairs further apart.

In semantic segmentation, self-supervised learning is a promising approach to address the scarcity of labeled data, garnering significant attention in recent research [CEKK20, WZS+21, ZA22, HCX+22, DRTT24, RSE+23]. Wang et al. [WZS+21] extended contrastive learning to pixel-level classification tasks by using pixel-level contrastive loss, demonstrating its effectiveness for semantic segmentation. Additionally, [ZA22] introduced a method for self-supervised learning of object parts for semantic segmentation, which combines the vision transformer's capability to focus on objects with spatial dense clustering tasks. This approach guides the model to learn representations corresponding to various potential object categories. [DRTT24] proposed a self-supervised despeckling task for remote sensing image data, showing experimentally that self-supervised despeckling can extract semantically meaningful features in remote sensing images, which are beneficial for downstream segmentation tasks.

While self-supervised learning is highly appealing, it also encounters significant challenges in practical applications. Finding a suitable upstream task for a specific downstream task is not straightforward. The design of the upstream task directly influences the features learned by the model and determines the patterns the model focuses on during training. Moreover, self-supervised learning demands substantial computational resources. For instance, SimCLR [CKNH20] in contrastive learning involves extensive data augmentation operations on each batch and computes similarities among all sample pairs in a large batch, which is computationally expensive.

2.5.5 Generative algorithms

Supervised learning inevitably depends on labeled data. Using a generative algorithm to automatically generate data offers a way to reduce labor

costs. Such algorithms can randomly generate data samples that statistically resemble the training data. Commonly used generative algorithms include:

- **Generative Adversarial Networks (GANs).** GANs [GPAM+20] consist of two neural networks: the generator and the discriminator. The generator aims to produce fake data that resembles real data, while the discriminator strives to differentiate between real data and generated data. These networks compete with each other during training until the generator can produce data that closely matches the distribution of the training data. For example, [ABB+20] employed a GAN to capture the typical distribution of bacterial colonies on agar plates. This GAN-generated synthetic data and performed style transfer to enhance visual authenticity, thereby addressing the shortage of annotated images in CNN-based bacterial colony semantic segmentation. Similarly, [ACB+21] utilized GANs to synthesize high-quality retinal images along with corresponding pixel-level labels. They trained the segmentation network using this synthetic data and achieved results comparable to those obtained with real data.

- **Variational Autoencoders (VAEs).** VAEs [KW13] map input data to a latent space using an encoder and reconstruct the data in this latent space through a decoder. The algorithm trains the model by maximizing the likelihood function of the data, allowing it to generate new data samples. [SYG+21] used generative VAE to generate a large number of unlabeled images and used the feature embedding provided by it to train a segmentation VAE that maps images to segmentation templates using a limited number of labeled images, thereby obtaining the annotations corresponding to the generated images.

- **Autoregressive models.** Autoregressive models [VdOKE+16] generate data by learning the conditional probability distribution of the data, where each new data point is generated based on the preceding data points. In computer vision, PixelCNN [VDOKK16] is a model that employs autoregression to generate images. In [LPL+20], a VQ-VAE model is initially trained with lesion data to extract a finite discrete code for each input image. VQ-VAE also learns to reconstruct the image from this code. Next, the code corresponding to the lesion part is extracted, and PixelCNN is trained to capture the pattern of the lesion code. The trained PixelCNN is then used to generate syn-

thetic lesion codes. These synthetic codes are inserted into the normal brain tissue code provided by a lesion-free image using a statistical model. The generated lesion code can also produce a lesion mask. Finally, the VQ-VAE decoder reconstructs the synthetic image with the lesion and derives its corresponding annotation based on the lesion insertion position. These synthetic training samples, along with the real annotations, are used to enhance the training of the segmentation network.

A generative algorithm functions akin to an artist who, after studying and internalizing the creative rules present in existing works, becomes capable of producing new creations. These creations can range from realistic images to coherent text and natural speech. Like the artist, the algorithm not only imitates but also generates entirely novel and equally impressive works. This ability is particularly valuable in addressing the challenge of insufficient data by enabling the algorithm to 'imagine' additional data to enhance AI training.

However, while generative algorithms excel at producing new data, the quality of generated data in tasks like semantic segmentation can be variable. Generated images may lack accurate labels, especially in complex scenes; for instance, in urban landscape images generated by such algorithms, buildings might mistakenly be labeled as roads.

2.5.6 Active learning

In semantic segmentation, active learning [SDV22] is an effective and interactive strategy to reduce the need for manual labeling. The core idea is to select a batch of samples that are likely to be misclassified by the current model, have them manually labeled by human annotators, and then use these labels to improve the current machine learning model. Active learning typically involves three main scenarios: pool-based sampling [CK24], stream-based selective sampling [TS23], and query synthesis [TS23]. In pooled active learning, the system has a large pool of unlabeled data, and the model can look at all the unlabeled samples at each iteration and select the most valuable samples to label based on strategies such as uncertainty sampling [NSH22]. In streaming active learning, unlabeled data arrives one by one. Whenever a new sample arrives, the system decides whether to label the sample or discard it. In query synthesis active learning, the model does not rely on existing unlabeled datasets, but actively generates new samples [PLK+24] and requests their labeling. Of these, pool-

based sampling is the most well-known and follows a structured process: (1)A small portion of the dataset is manually labeled and used to train the model; (2)The model then predicts the labels for the remaining unlabeled data, and associate a confidence level to these predictions; (3)Sampling techniques are employed to select the next subset of data for manual annotation; (4)The selected subset is manually labeled and used to further train the model; (5)Above steps are repeated until the model reaches the desired level of accuracy. The success of active learning heavily relies on the sampling methods used to choose the data to manually label. Common sampling methods include:

- **Uncertainty sampling [NSH22].** The model selects pixel regions where its predictions are most uncertain. Uncertainty is often measured using entropy or prediction probabilities.
- **Diversity-based sampling [JYQS22].** This approach focuses on selecting a diverse range of samples to ensure that the model is exposed to a broad variety of features, helping to avoid overfitting to specific scenarios.
- **Committee-based sampling [SHV18].** Multiple models form a committee, and the areas where the models disagree most are selected for labeling, ensuring that the most contentious regions are addressed.

Compared to streaming-based and query synthesis active learning, our work adopts a pool-based interactive approach, which allows for more effective selection of informative samples from a predefined dataset. This methodology not only enhances the efficiency of the learning process by focusing on hard (and high-value) samples but also aligns with our thesis objectives by ensuring that the training data is both diverse and representative.

2.6 Conclusion

Semantic segmentation is a critical task in computer vision. To evaluate segmentation algorithms, several commonly used evaluation metrics have been introduced. Traditional semantic segmentation methods include threshold-based segmentation, edge detection-based segmentation, region-based segmentation, energy minimization segmentation, and segmentation based on traditional machine learning. However, the performance

of these traditional methods is often limited. With the advent of deep learning, extensive research has been conducted on CNN-based semantic segmentation algorithms, leading to the development of typical network structures. Despite their success, these methods are highly dependent on labeled data. This chapter introduces commonly used approaches to address the issue of insufficient annotated data. The following chapters will focus on strategies to reduce the annotation effort required for deep learning training.

Chapter 3

On the importance of diversity when training deep learning segmentation models with error-prone pseudo-labels

The key to training deep learning (DL) segmentation models lies in the collection of annotated data. The annotation process is, however, generally expensive in human resources. This chapter follows one of our publications[YRJ+24], where leverages deep or traditional machine learning methods trained on a small set of manually labeled data to automatically generate pseudo-labels on large datasets, which are then used to train so-called data-reinforced deep learning models. The relevance of the approach is demonstrated in two applicative scenarios that are distinct both in terms of task and pseudo-label generation procedures, enlarging the scope of the outcomes of our study. Our experiments reveal that (i) data reinforcement helps, even with error-prone pseudo-labels, (ii) convolutional neural networks have the capability to regularize their training with respect to labeling errors, and (iii) there is an advantage to increasing diversity when generating the pseudo-labels, either by enriching the manual annotation through accurate annotation of singular samples, or by consid-

ering soft pseudo-labels per sample when prior information is available about their certainty.

3.1 Introduction

Semantic segmentation, which consists in a pixel-wise classification, is a fundamental and essential task in the field of computer vision. With the recent development of deep learning, automatic image processing has become accurate in many fields, at the cost of the time-consuming annotation of large amounts of data [AZ20, CVL18, SE21]. To limit the annotation time, semi-supervised approaches have received extensive attention [Zho21, OHT20, ZYH+22, SDV22]. The main idea of semi-supervised methods is to learn a model using a small amount of labeled data and a large amount of unlabeled data. The key to the success of this strategy lies in how to leverage limited labeled data to generate pseudo-labels. Pseudo-labels are labels that are automatically generated, based on the knowledge extracted from the few labeled data. In contrast to ground-truth labels, pseudo-labels are subject to errors. Resorting to the automatic generation of pseudo-labels is motivated by the fact that manual annotation of the entire dataset is unrealistic in many practical applications, where the amount of data is large and the annotation is pixel-wise.

In practice, semi-supervised training is implemented in two steps [MDSR+20, XLHL20, AOA+20a, YZQ+22, ZZW+21]. In the first step, labeled data are used to train a relatively weak machine learning model. In the second step, the weak model is used to assign (error-prone) labels to unlabeled data, which are used to supervise the training of a hopefully more robust model as is shown in Figure 3.1.

This chapter focuses on the cases where the manual annotation load is kept low. In this specific case, traditional machine learning methods are known to achieve a good trade-off in terms of accuracy and generalization capabilities [Smi10, CWC+22b, FRD12, WH21]. Hence, we explores how traditional machine learning models can be integrated in a semi-supervised training approach either to facilitate the annotation of the few initial labeled data (first step) or to predict the pseudo-labels of unlabeled data (second step).

The envisioned semi-supervised framework is depicted in Figure 3.1. A small set of (semi-)manually annotated samples are considered to train an initial and relatively weak segmentation model. This initial model is

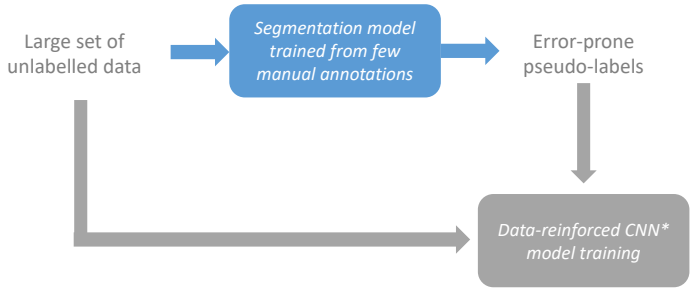


Figure 3.1: Overview of the envisioned semi-supervised framework. A small set of (semi-)manually annotated samples are considered to train a relatively weak segmentation model. This weak model is then used to annotate a potentially large set of unlabeled data, resulting in error-prone pseudo-labels, which are used to supervise the training of a convolutional neural network. This neural network, denoted CNN*, is named the *data-reinforced* model because it leverages the distribution of patterns in the unlabeled dataset to distill the knowledge embedded in the strictly manually supervised model.

then used to annotate a potentially large set of unlabeled data, resulting in error-prone labels, which are used to supervise the training of a convolutional neural network. This neural network is denoted CNN* in the following, and is named the *data-reinforced* model because it leverages the distribution of patterns in the unlabeled dataset to improve the strictly manually supervised model. In a sense, the unlabeled data and associated pseudo-labels are used to distill the knowledge captured by the supervised model.

Two flavors of the pseudo-label generation framework are presented in Figure 3.2a,b. They differ in the way they exploit traditional machine learning approaches to collect manual annotations and turn them into pseudo-labels. They have been designed to address the specificities of the segmentation problem they are dealing with.

Figure 3.2a considers a scenario where the objects to segment are easy to delineate manually. Hence, manual labels are assumed to be available for a few images and are used to supervise the training of a conventional semi-naive Bayesian estimation model. This model provides class probability estimates on unlabeled data, which are then used to define pseudo-labels and supervise the training of the data-reinforced CNN* model. Our experiments in Section 3.4 demonstrate that adopting a stochastic approach

to randomly turn the soft probabilities into pseudo-labels at each training iteration is advantageous since it limits the potential bias induced by systematic errors of the Bayesian estimator.

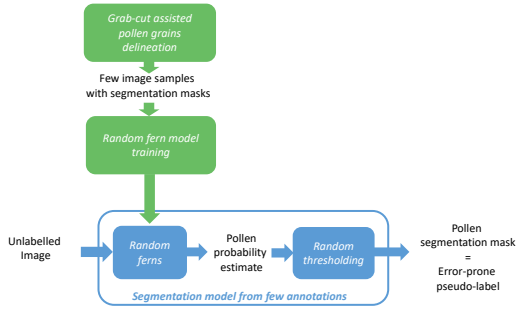
In contrast, Figure 3.2b considers a case where objects have complex shapes, thereby requiring some semi-automatic approach to be delineated. In our study, various machine learning models (random forests, gradient-boosted detection trees, and even CNNs) have been considered to assign labels a few initial images. In practice, those machine learning models are trained interactively by manually assigning labels to image parts (strokes or blocks). This is performed either on individual images (one model per image) or jointly on the whole set of initial images (one joint model for all images). When one model is trained per image, a CNN is then trained from the whole set of labeled images to aggregate the entire annotation knowledge in a single joint model. The gradient boosted trees (GBT) model is quite attractive for handling an interactive annotation procedure since it is convenient to update based on novel annotations. However, a single GBT has appears to be unable to generalize to a set of pictures. Hence, we propose to interactively learn one GBT per image, and then to use the resulting annotations to train a CNN as a pseudo-label generator. In all cases, the joint model is used to generate the pseudo-labels on unlabeled data. As a valuable and original outcome, our experiments reveal that the initial annotation methods that randomly select the manually annotated image part samples, or tend to regularize the manual annotation inputs (by capturing their main average trends, e.g., consequently to the use of Bayesian models), result in weaker data-reinforced models, compared to methods that favor and are able to preserve the manual annotation diversity.

In our experiments, the two scenarios correspond to pollen and crop images, respectively. Through quantitative and qualitative analysis, we find that, in both cases, the integration of conventional machine learning algorithms in the annotation process can achieve good results with a moderate manual annotation effort. The main outcomes of our study can be summarized as follows:

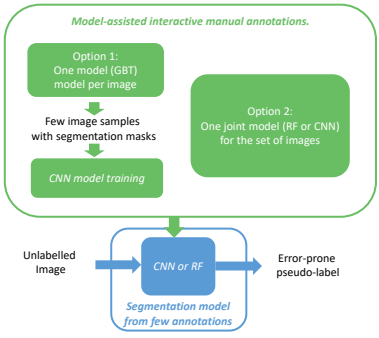
(1) Data-reinforced training based on large set of unlabeled data helps, even with error-prone pseudo-labels. This is attested by the higher segmentation accuracy achieved by the data-reinforced models, compared to their corresponding (error-prone) pseudo-label generator.

(2) CNNs training is robust to labeling errors, especially when those errors do not present systematic patterns.

(3) Adopting stochastic pseudo-labels or increasing the diversity of



(a)



(b)

Figure 3.2: Overview of the two scenarios envisioned in our study to collect the few manual annotations needed to train the pseudo-label generator. (a) **The object is easy to segment manually**, and a few manual labels are used to supervise the training of a conventional Bayesian estimation model providing class-probability estimates on unlabeled data. Those class probabilities are then used to define pseudo-labels and supervise the training of a CNN model. (b) **The object has complex shapes requiring some semi-automatic approach to be delineated.** Two options are envisioned. In Option 1, gradient-boosted detection trees are considered to interactively assign labels to a few initial images. The trees, which are Bayesian estimators in essence, appear to poorly generalize to new data. Hence, a CNN is considered to be trained from the few labeled data and is used to generate the pseudo-labels on unlabeled data. In Option 2, one model (either CNN or random forest) is progressively trained from the manual annotations that are interactively collected on a small set of images to correct the current model predictions. This final model is the used as the pseudo-label generator.

manual annotations through careful interactive selection of hard samples is shown to increase the benefit derived from data-reinforced training. This is supported by our experiments, which show that pseudo-label generators with the same prediction accuracy can lead to varying performance in the data-reinforced model. The best results are achieved with CNN pseudo-label generator trained using samples that are interactively selected to correct its prediction errors. Using a CNN avoids the dilution (of annotation information) inherent to probability estimates manipulated by Bayesian models, while the interactive selection of samples results in larger diversity than a random selection. Despite the fact that they are consistent across all the methods and models tested on our two datasets, our experimental results remain too preliminary to draw a definitive and general conclusion about the need for diverse annotations and an expressive pseudo-label generator. Our study is, however, sufficiently convincing to trigger further theoretical and experimental investigations.

3.2 Related Work

Semantic segmentation currently plays a very important role in many fields, such as medical image analysis, scene understanding, and robotic perception [MWY+22]. Although deep learning models have achieved very good performance in semantic segmentation [XLG+20], they require labeled data. The acquisition of annotations generally requires a lot of time and manpower, and many fields require experts to do the work. Therefore, considering the use of unannotated data in semantic segmentation has been investigated.

Many methods based on unlabeled data have been studied to improve the performance of learning algorithms. Refs. [RDRS21, ZWH+21, WSCM20, LYB+19, LYZ+17] achieved good results in image classification using a semi-supervised approach, consisting in assigning pseudo-labels to unlabeled data, using a model trained from a few data. Ref. [PLP20] has been a precursor in proposing to initially train one classifier on a labeled dataset in order to make predictions on an unlabeled dataset, which uses an initial small labeled dataset to train deep or shallow neural networks and other non-interpretable but high-precision models on the self-training framework, thereby obtaining an enlarged dataset, which is used to train interpretable models such as random trees. In the first stage of this method, a certain amount of labeled data is needed to obtain high accuracy based

on CNN. Secondly, the simple model such as random trees has poor generalization ability. Semi-supervised segmentation approaches usually augment manually labeled training data by generating pseudo-labels for the unlabeled data and using these to generate segmentation models [CAS+22]. Consistency regularization and entropy minimization represent two prevalent strategies in using unlabeled data [ZZZ+20]. Consistency-based approaches [XLBY22, ZZL+22] operate on the premise that the model's output should remain stable under input perturbations. Conversely, entropy minimization [VJB+19, VGK+21, ZPL+23] suggests that unlabeled data can be exploited in achieving clear class distinctions. Alternatively, refs. [ZZW+21, XLHL20, CDI+19, BC22] introduced methods where pseudo-labels are generated from a universal teacher model to train a problem-specific student model. Those works assume that enough annotated data are available to teach a high-quality teacher. Errors made in the early iterations can be propagated and amplified in subsequent iterations, leading to a student model that may become increasingly confident in incorrect predictions. Moreover, the model's predictions on unlabeled data may lack accuracy when the initial dataset labeled to train the teacher is not sufficiently representative of the domain targeted by the student.

Although the above methods achieve good results based on labeled and unlabeled data, previous works have largely overlooked the relation between the strategy (including the selection of manually labeled samples, and the pseudo-label generator model trained from those samples) adopted to generate pseudo-labels and the benefit obtained when using those pseudo-labels (at constant average quality) to train a data-reinforced model. We propose integrating traditional machine learning methods with deep learning to address the challenge of scarce labeled data. Semi-automatic interactive approaches are considered to collect accurate labels on complex segmentation problems. Two distinct scenarios are envisioned. In the scenario depicted in Figure 3.2a, the objects to segment are relatively easy to discriminate from their background, and a simple Bayesian random fern model is sufficient to capture the knowledge contained in manually annotated samples. When using the random ferns to infer pseudo-labels on new data, prior information is available about the class-label certainty/probability. As an original contribution, we propose to exploit this prior to introduce diversity among the pseudo-labels. Specifically, each time a sample is considered in a training iteration, the segmentation map of each class is defined based on a random threshold, applied to the certainty level. In the second scenario, considered in Figure 3.2b,

the segmentation task is more complex, and a CNN is considered as an alternative to the Bayesian pseudo-label generator, to capture the knowledge associated to the (semi-)manual annotations, without 'averaging' it in a Bayesian inference framework. Moreover, in this second scenario, the samples to manually label are either selected randomly or to correct the errors of the model trained from previously selected and annotated data. As a main original outcome, our experiments reveal that, at constant pseudo-label average accuracy, the models trained from pseudo-labeled data are more accurate when those pseudo-labels reflect the pseudo-label generator uncertainty (first scenario) and when the pseudo-label generator preserves the diversity of annotations inherent to an interactive selection of samples to manually annotate (second scenario, with the interactive selection of samples and CNN aggregation of knowledge, without Bayesian dilution).

3.3 Methods

As explained in Section 3.1, we investigates how to leverage machine learning models when resorting to pseudo-labels to train a deep learning model on a large dataset.

In this case, a small amount of manually annotated data is used to train a model, and this manually supervised model is exploited for the automatic labeling of additional data. Research questions of interest are related to (i) the selection of the data to manually annotate, (ii) the choice of tools to manually annotate them, and (iii) the design of appropriate methods to turn the (semi-)manually collected knowledge into pseudo-labels on novel data samples.

Two distinct scenarios are considered in our experimental section. For each scenario, the above questions are addressed differently, but common lessons can be drawn for both of them, resulting in useful practical guidelines.

The rest of the section presents the multiple options envisioned by each step of our framework, depicted in Figure 3.1. Section 3.3.1 considers the (semi-)manual annotation of a data subset. Section 3.3.2 focuses on how to embed the knowledge captured through those manual annotations into a machine learning model, in charge of transferring this knowledge to novel data. Section 3.3.3 then introduces two strategies to pseudo-label additional training data for the purpose of training more accurate models when

considering a large set of (error-prone) pseudo-labeled data than the accurately labeled but smaller initial set of samples.

3.3.1 (Semi-)Manual Annotation

Multiple approaches are considered to manually assign a background/-foreground segmentation mask to a given training sample. In practice, a specific off-the-shelf approach will be selected based on the case at hand, i.e., on the shape of the foreground region to discriminate from its background.

The investigated methods encompass the following:

- The manual delineation of the foreground object contours. This solution is suited to smooth and regular object shapes, and is adopted in the following to delineate the pollen grains in our first experimental scenario.
- Semi-manual methods, relying on interactive user annotations to refine and increment the training set that is used to train a machine learning model. In this approach, there is a large pool of unlabeled data. The model can view all the unlabeled samples at once. The user then selects samples where the model makes prediction errors for annotation. Such an interactive approach, falling in the pool-based active learning category presented (see Section 2.5.6), suits in cases where the foreground and background are visually dissimilar but are separated by an irregular border, requiring time for fully manual delineation.

Among the interactive training methods, we differentiate methods that train a single model to segment the whole set of images to manually annotate, from the ones that learn a specific model to label each individual image. In the first category, the Ilastik 1.3.3 [BKK+19] updates a random forest from user-defined foreground strokes until it achieves a sufficiently accurate segmentation of the training data. Another approach investigated in our experiments considers a set of small patches extracted randomly within the whole set of images, to train a fully convolutional neural network (CNN). In contrast, the RootPainter approach [SHP+22] selects the patches interactively. Specifically, it trains, jointly for all images, a CNN based on the user interactive selection and labeling of patches that are wrongly predicted by the CNN segmentation model at some point in the training.

In the second category, our LeafPainter method adopts gradient-boosted decision trees (GBTs) [SZK+17], and the manual annotation of small patches that are interactively selected by the user in response to the GBT prediction errors. One GBT model is learned for each image (GBT has been implemented using the LightGBM library version 3.2.0 from Microsoft (Ke et al., 2017) so as to support a fluid interactive annotation process. LeafPainter was created with the Python 3.10 programming language, using the Flask library (Grinberg, 2018). The interface was implemented in HTML5, CSS, and Javascript. The code is available at `https://github.com/charlesro/LeafpainterQT`, and a visual demonstration is available at `https://www.youtube.com/watch?v=lMpzxDt40Lc`). GBTs have the advantage of being computationally simple to update when new annotations are provided by the user. They, however, suffer from a lack of generalization capabilities, preventing the use of a single GBT for all images.

In the rest of the paper, those methods are denoted as sRF, CNN(sRP), sCNN, and iGBT respectively. The s or i prefixes refer to the fact that a model is trained for the entire set of images or for each individual image, respectively. Through interactive annotation, sCNN and iGBT are exposed to different samples at various stages of training, increasing the diversity of annotations. For sCNN and iGBT, in the early stages, there are fewer annotations, but as training progresses, users randomly correct the model's errors. This not only enhances diversity but also gradually increases the number of annotations. Our experiments reveal that increasing annotation diversity by learning a distinct model for each image or by interactively training a CNN (i.e., a model offering a large expressivity) improves the benefit drawn from unlabeled data (see Section 3.3.3). In addition, when utilizing manual annotation time to quantify the amount of labeling, our experiments also demonstrate that fewer annotations are required to reach a specific model accuracy when the diversity of annotations increases.

3.3.2 Training a Pseudo-Label Generator

Once a number of images (or image patches) have been properly labeled, they are used to train a machine learning model, using conventional supervised training. This model is the one in charge of predicting pseudo-labels.

Two types of models are considered at this stage.

The first one follows a Bayesian approach. Several variants are available, including random forests [SZ20] and random ferns [BDVJ+16, PDV17]. They can achieve good results with only a small amount of labeled data.

Moreover, since those methods naturally provide a class posterior for each pixel, multiple pixel-wise segmentation masks can be generated by thresholding the estimated probability map with more or less conservative thresholds. Our experiments reveal that this diversity increases the benefit drawn from unlabeled data when using them as training samples (Section 3.3.3). In practice, our experiments rely on random ferns to segment pollen grain microscopy images. Random forests (RFs) have also been trained from a set of canopy images, to be used directly as a pseudo-label generator (see Table 3.1).

The second one adopts a recent convolutional deep learning model. These kinds of models are more expressive and offer better generalization capabilities than conventional machine learning models when dealing with 'in the wild' observations [BY24, NBMS17, KKB22], namely, with observation conditions that are weakly constrained, as is the case for the canopy images considered in our second experimental scenario. In this scenario, plant images are (semi-)manually segmented, either using one GBT model per image or a single convolutional network for the whole set of images (as proposed in [SHP+22] and presented in Section 3.3.1). When one GBT is trained per image, which offers fluent interactions with the user given the computational efficiency associated to GBT updates, the resulting segmentation masks are used to supervise a convolutional neural network (denoted as CNN(iGBT) in Table 3.1).

Table 3.1: Names of models on outdoor RGB canopy image dataset. X indicates that the test model has been trained on manually labeled data only, without pseudo-labeled data exploitation.

(Semi_)Manual Annotation	Pseudo-Label Generator	Test Model
sRF [BKK+19]	X	sRF
	sRF	CNN*_sRF
sRP	CNN(sRP)	CNN*_CNN(sRP)
	X	CNN(sRP)
iGBT	CNN(iGBT)	CNN*_CNN(iGBT)
	X	CNN(iGBT)
sCNN [SHP+22]	sCNN	CNN*_sCNN
	X	sCNN

Table 3.2: Foreground IoU of different methods on pollen test dataset as a function of the number of manually labeled images. In total, 250 pseudo-label samples, and 50 test samples are considered.

	Foreground IoU (%)					
Method	**1 Images**	**5 Images**	**10 Images**	**15 Images**	**25 Images**	**50 Images**
CNN	37.4	58.9	69.9	72.4	80.4	81.6
Random Ferns	74.7	74.0	77.3	80.0	79.6	79.2
CNN*_0.45	72.2	76.0	76.0	73.7	74.7	74.3
CNN*_0.50	80.0	81.3	82.2	82.4	82.6	82.7
CNN*_0.55	75.0	73.3	76.2	80.3	79.7	78.2
CNN*_(0.45,0.55)	**84.5**	**83.3**	**84.3**	**84.8**	**85.0**	**85.1**
CNN*_CNN	45.7	65.9	73.8	75.7	82.8	82.9

3.3.3 Training a CNN from Pseudo-Labels

Given a segmentation model trained with manually labeled data, our study considers the exploitation of this model to generate/predict pseudo-labels on unlabeled data so as to train a so-called *data-reinforced* segmentation CNN model, denoted CNN* in the following, with the * symbol indicating the use of pseudo-labels on a large dataset to train the model.

When the manually supervised model follows a Bayesian paradigm, a posterior background/foreground class probability map is predicted for each unlabeled image. Instead of using a fixed threshold to turn this probability map into a foreground/background binary mask, which is likely to induce systematic errors in the supervision, we envision the use of distinct thresholds each time a sample is considered during training. Specifically, each time unlabeled data are considered during training, a threshold is randomly selected using a uniform distribution on a pre-defined range interval (a,b), to produce a corresponding mask of pseudo-labels. In this way, the reference segmentation mask of the same picture is different at distinct training iterations, thus increasing the diversity of labels. Associating multiple (and thus diverse) labels to a single image smoothly favors the labels that are most consistent with a given visual pattern across images, and is expected to mitigate the impact of erroneous pseudo-labels. Our experiments demonstrate that this supervision, named soft-supervision in the following, results in improved generalization performance of the model trained on unlabeled samples (see Tables 3.2 and 3.3).

With a CNN trained from manual annotation, the access to a probability map, whilst possible [LLKDV22, G+16, SG18], is less obvious since it re-

quires multiple predictions to estimate the prediction certainty. Therefore, in our experiments, we consider a single mask per image when exploiting unlabeled data based on pseudo-labels generated with a CNN model.

3.4 Experimental Validation

This section considers semantic image segmentation in two use cases that significantly differ in their image acquisition set-up, which is known to affect generalization, and in the regularity of the shapes to be segmented, which directly affects the annotation strategy. For both use cases, the section considers experiments to demonstrate that assigning pseudo-labels to unlabeled data based on models trained on a few (semi-)manually annotated samples improves the quality of the resulting data-reinforced CNN* segmentation model, compared to a CNN model trained only on the manually labeled data.

3.4.1 Datasets

Pollen Grain Microscopy Images

The proposed method is evaluated on a pollen dataset. The pollen files are scanned by biologists, including Anne-Laure Jacquemart, in pure pollen files and uploaded to Cytomine [RHV+19]. The resolution of each file is up to 30,000 by 30,000, which limits the training of the model, so we crop 800 by 800 images in those large images. Fifty of these 800×800 images are extracted to be manually labeled. An additional set of 300 images is

Table 3.3: Pixel accuracy of different methods on pollen test dataset under different labeled images.

Method	Pixel Accuracy (%)					
	1 Images	5 Images	10 Images	15 Images	25 Images	50 Images
CNN	75.2	86.2	94.6	95.1	97.3	97.4
Random Ferns	97.0	96.9	97.3	97.6	97.5	97.5
CNN*_0.45	90.1	94.4	96.0	96.2	95.5	95.7
CNN*_0.50	97.0	97.9	98.0	97.4	97.7	97.8
CNN*_0.55	97.6	97.4	97.6	98.1	98.0	97.9
CNN*_(0.45,0.55)	**98.4**	**98.1**	**98.2**	**98.2**	**98.3**	**98.2**
CNN*_CNN	73.5	83.3	91.9	92.7	98.2	98.1

extracted from other high-resolution original images. Of these, 250 remain unlabeled, while the other 50 are manually labeled to define a test set.

RGB Canopy Images

The dataset consists of 2386 zenithal RGB images of experimental plots, each containing either a single vegetable species or a mixture of two different vegetables. From the dataset, 280 images were randomly extracted to be manually annotated and serve as a test set. There are four vegetable species in total: fennel, dwarf bean, cabbage, and kale. The dataset includes images of each species in pure culture as well as all possible combinations of two vegetables. We are interested in differentiating plant and non-plant pixels.

Each plot was photographed four times over the course of the four-month (July, August, September, and October 2021) experiment, resulting in images of the plants at different stages of growth and under varying sunlight conditions. The photos were taken in Belgium at either the Lauzelle farm (50.680, 4.618) or one of 20 volunteer market gardeners' plots. The resulting images exhibit a wide range of contexts, including variations in soil types, crop arrangements, and non-plant objects present on the plots, as well as differences in cultivation techniques among the market gardeners. The cameras used consist of 2 OEM OV2640 (Omnivision) camera modules (OmniVision, Santa Clara, CA, USA), RGB, 2Mpixel, embedded in an ESP32-Cam module (Espressif Systems, Shanghai, China), installed parallel to each other.

3.4.2 Methods and Terminology

Several variants of the methods described in Section 3.3 have been considered for each use case. They are summarized in Tables 3.4 and 3.1, and are presented below.

Pollen Grain Microscopy

The manual annotation of microscopy pollen grain images is implemented using Grabcut [LZG+18], which is both efficient and effective given the simplicity of the image content. Those manually labeled images are then used to train a random ferns model [OCLF09], which in turn is used to assign pseudo-labels to the unlabeled images. In practice, we utilize 100 ferns

and each fern consists of 10 tests. The point neighborhood is typically defined using a small square window. This square window has a size of (2r + 1) by (2r + 1) pixels and is centered around the pixel of interest. Within this window, binary tests are performed. Five of them compare the absolute difference between two pixels with t, where t is derived from the statistical distribution of the absolute differences between any two pixels in the 800 by 800 image, and t is randomly selected from the interval (20, 40). This choice is motivated by the observation that the majority of difference values fall within the range of 0 to 20. Consequently, when two pixels exhibit difference values within this range, they are either both part of the foreground or both part of the background. Therefore, these five tests are capable of distinguishing edges from non-edges. The other five tests involve comparing pixel value with the difference between the peak and the variance of the histogram of the 800 by 800 image. Given that most of the image area represents the background, the peak of the histogram is indicative of the background location. As the pixel values of the background exhibit slight variations, we utilize both the peak and the variance of the histogram to characterize the background. Consequently, these five tests effectively distinguish between the background and foreground.

The last column in Table 3.4 presents the test models, i.e., the models whose performance are measured on the test set and reported in our experimental section. The CNN and random ferns models are trained on manually labeled data only, while the CNN* models refer to data-reinforced models, meaning that they are trained on the entire training set based on pseudo-labels derived either from the random ferns or the manually trained CNN. When pseudo-labels are predicted by the CNN, the data-reinforced model is denoted CNN*_CNN. When deriving pseudo-labels

Table 3.4: Models consideredon pollen grain microscopy image dataset. See 3.4.2 for a description of the models associated to the terminology. X indicates that the test model has been trained on manually labeled data only, without pseudo-labeled data exploitation.

(Semi_)Manual Annotation	Pseudo-Label Generator	Test Model
Manual Delineation	X	CNN
	X	Random Ferns
	Random Ferns τ	CNN*_τ
	Random Ferns (τ_1,τ_2)	CNN*_(τ_1,τ_2)
	CNN	CNN*_CNN

from the random ferns, the pseudo-labels are generated either by thresholding the probability map predicted by random ferns at a fixed threshold τ, leading to a trained convolutional model that is denoted CNN*_τ, or by considering a uniform distribution of thresholds in an interval (τ_1, τ_2) to randomly select the threshold each time an image is considered along the SGD training iterations (see Section 3.3.2). This last definition of pseudo-labels results in a trained data-reinforced model denoted as CNN*_(τ_1, τ_2)

Outdoor RGB Canopy Images

The (semi-)manual segmentation of leaves requires more elaborated tools than a Grabcut method, due to the variation in illumination conditions, the presence of shadows, and the frequent occlusions by background objects or plants. Therefore, machine learning models are considered to label a subset of the training set through the iterative annotation of image parts.

The subset of samples to label consists of 800 samples, corresponding to 800×600 images, extracted from full resolution images. Those samples are jointly considered by the Ilastik software [BKK+19], which trains a random forest (RF) model (100 trees, with a maximal depth of 50) based on foreground strokes that are interactively drawn by the user. This approach is denoted sRF, with the prefix 's' referring to the fact that the whole set of images to manually label is jointly processed. As an alternative, the sRP method randomly samples patches in the same whole set and trains a CNN based on their annotation (see below for CNN parameters). Two more methods are considered to manually annotate the 800 image samples. They are interactive and proceed iteratively by adopting the corrective annotation protocol recommended in [SHP+22]. A small subset of samples is first annotated (based on the delineation of small patches corresponding to a single class), without referring to a model prediction. Those annotations are used to train an initial version of the model. Subsequent annotations correct the model predictions and are used to update the model. Two types of image segmentation models are envisioned, resulting in two distinct methods. The first one corresponds to the gradient-boosted decision trees [SZK+17] model, with default parameters. In that case, due to the experienced lack of generalization capabilities of GBT, one model is trained for each of the 800 images, and the method is denoted *iGBT*, with the 'i' prefix referring to the fact that one GBT is trained for each individual image. The second one is a fully convolutional neural network (CNN). Its expressivity is sufficient to capture the knowledge associated to the entire

set of 800 images. Therefore, a single model is considered for the whole set of manually labeled images. It is denoted $sCNN$. The notation CNN*_X is adopted to refer to the data-reinforced convolutional network trained with pseudo-labels generated by model X.

3.4.3 Metrics and Implementation Details

Evaluation Metrics

In semantic segmentation, the intersection over union is a standard metric to assess segmentation models. It calculates the intersection over union ratio of ground truth and predicted binary masks for the foreground. Formally, it is defined as:

$$\text{IoU} = \frac{|G \cap P|}{|G \cup P|} \tag{3.1}$$

with G denoting the ground truth, and P the model prediction associated to the foreground.

Another metric used is pixel accuracy, which evaluates the model by calculating the proportion of correctly classified pixels to the total number of pixels in the image as

$$\text{PA} = \frac{\sum_{i=0}^{1} p_{ii}}{\sum_{i=0}^{1} \sum_{j=0}^{1} p_{ij}} \tag{3.2}$$

where p_{ii} is the number of pixels whose prediction and ground truth are both i. p_{ij} is the number of pixels whose prediction is i and label is j.

CNN Network

The CNN network used in our work consists of an encoder and decoder, following a U-Net shape [RFB15a, IR20]. Compared to the U-net introduced in [YJJ+22a], we use a seven-layer network. Each layer in the network consists of a convolution operation, followed by batch normalization and the Rectified Linear Unit (ReLU) activation function. The encoder is composed of four layers, with each layer being followed by a downsampling operation. Following the bottleneck layer, we have the decoder. In order to facilitate training with larger batch sizes on the pollen dataset, we aim to minimize model parameters. To achieve this, we simplify the

decoder section by reducing the number of layers to two, each preceded by an upsampling operation, compared to the classic U-Net architecture. And a ResNet50 (pretrained on ImageNet) is used as an encoder for the canopy images.

During the training, random rotate90, flip, transpose, and brightness contrast adjustment are used as data augmentation methods.

Loss

The loss function adopted to train the CNN is defined in equation (3.5) as the Log-Cosh Dice Loss [Jad20]:

$$coshx = \frac{e^{x} + e^{-x}}{2} \tag{3.3}$$

$$DiceLoss = 1 - \frac{2\sum_{i=1}^{N}\sum_{c=1}^{C} y_i^c \cdot p_i^c}{\sum_{i=1}^{N}\sum_{c=1}^{C} y_i^{c2} + \sum_{i=1}^{N}\sum_{c=1}^{C} p_i^{c2}} \tag{3.4}$$

$$Loss = \log(cosh(DiceLoss)) \tag{3.5}$$

In equation (3.4), y_i^c is a binary indicator (0 or 1) indicating whether pixel i lies in class c or not, and p_i^c is the softmax output of the network for pixel i and class c.

Training

Stochastic gradient descent algorithm with 0.9 momentum and 0.0001 weight decay is used with a batch size of 4. In total, 100 training epochs are considered. We use 0.01 as the initial learning rate, and an exponential learning rate scheduler is implemented: $lr = lr_{init} \cdot 0.97^{epoch}$. We save the last 10 models from a training session and use the mean prediction as the result. All models are trained with Pytorch [PGM+19b] on Nvidia GeForce GTX 1080 Ti 11Gb GPUs (Nvidia, Santa Clara, CA, USA).

3.4.4 Results and Discussion

The methods in Tables 3.4 and 3.1 were experimentally tested on the pollen grain microscopy image dataset and the outdoor RGB canopy image dataset.

Pollen Grain Microscopy

In Tables 3.2 and 3.3, we present the results of our experiments on the pollen grain microscopy dataset. The results obtained with the models trained only from manually labeled data are provided as a baseline but also because they reflect the level of error affecting the pseudo-labels. They indicate that the random ferns method performs significantly better than the CNN method when only one annotated image is available. However, as the annotated data increase, the CNN method overcomes the random ferns. Specifically, when using 50 annotated images, the CNN achieves a foreground IoU result of 81.6%, surpassing random ferns, which achieves 79.2%. The pixel accuracy is also comparable, with values of 97.4% for CNN and 97.5% for random ferns. When using random ferns and CNN as pseudo-label generators, even with only one annotated image, the CNN*_0.50 yields an IoU result of 80.0%, outperforming random ferns, which achieves 74.7%. The experimental results clearly indicate that leveraging unlabeled data through pseudo-labels enhances CNN* model quality compared to a model (either random ferns or CNN) trained solely on manually labeled samples. Interestingly, Tables 3.2 and 3.3 also demonstrate that CNN*_(0.45,0.55) consistently outperforms CNN*_0.45, CNN*_0.50, and CNN*_0.55, indicating that increasing the diversity of pseudo-labels is beneficial. Additionally, Figure 3.3 illustrates the predictions of various methods on the pollen test set. The models depicted in the figure are trained using only a single labeled image. Through the comparison between the predictions and the ground truth, it is observed that employing a fixed threshold to generate pseudo-labels causes a bias in training [AOA+20b] , resulting in inferior predictions compared to using a varying threshold.

Outdoor RGB Canopy Images

Tables 3.5 and 3.6 present the outcomes of experiments conducted on the RGB canopy dataset by Charles in [YRJ+24]. Table 3.5 shows the foreground IoU for various methods on the outdoor RGB canopy image dataset under different manual annotation times. Here, the manual annotation time denotes the time needed to annotate the training set used to learn the pseudo-label generator. It includes both the time to manually correct the errors affecting the current model prediction, and the (computational) time to update the model. Table 3.6 presents the corresponding pixel accuracy. The results indicate that the data-reinforced CNN* consistently out-

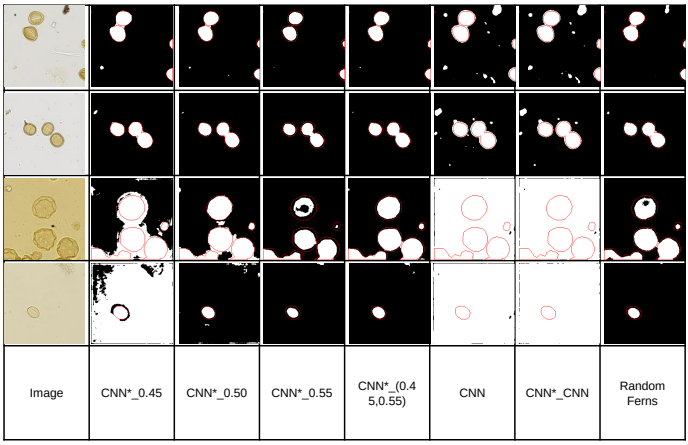


Figure 3.3: Qualitative results on pollen test dataset. See Table 3.4 for model names definition. All models are trained using 1 labeled image. Red contour is ground truth.

performs the methods that are solely trained on manually labeled data. This demonstrates that leveraging unlabeled data through pseudo-labels is beneficial. Note that the quality of pseudo-labels is measured by the test performance obtained by models trained from manually labeled data. Without surprise, for a given pseudo-label generator model, the higher the pseudo-label quality, the higher the CNN* quality.

A similar observation can be made in Figure 3.4a, where each CNN* model achieves test performance that is generally superior to that obtained by the model from which the CNN* training pseudo-labels have been predicted. Hence, we conclude that resorting to pseudo-labels helps in reducing the human annotation workload. This is confirmed by Figure 3.4b, which compares the pixel accuracy on the test set between the CNN* model and its corresponding pseudo-label generator. The points in the figure are consistently located above the diagonal, emphasizing that leveraging unlabeled data through pseudo-labels improves the model quality compared to a model trained only on manually labeled samples.

Figure 3.4a also clearly reveals that the boost obtained by the CNN* compared to the model used to predict the pseudo-labels is much larger for CNN(iGBT) and sCNN than for CNN(sRP) and sRF. For example, when the model trained on manually labeled samples reaches about 83%, CNN*_CNN(iGBT) reaches 94.5%, while CNN*_sRF only reaches 88.6%.

This observation was not expected, and is thus quite interesting.

This unexpected discrepancy is attributed to the combination of two factors. First, compared to CNN(sRP), the sRF, CNN(iGBT), and sCNN models benefit from more diverse manually labeled samples since those samples are not randomly selected but instead chosen interactively as image parts where the current state of the model fails. Second, compared to sRF, CNN(iGBT) and sCNN build on a CNN model, which is known to have sufficient expressivity to capture a large annotation diversity without regularizing it [ZBH+21], i.e., without aggregating the annotation cues through average probabilities, as done by Bayesian models like random ferns or random forests. While the enriched representation captured by a CNN model does not necessarily translate into improvements on the test images (e.g., 83% test accuracy both for the CNN(iGBT) and sRF models), it proves to be beneficial when confronted with the distribution of unlabeled data, i.e., when leveraging their predicted labels to train a data-reinforced CNN*.

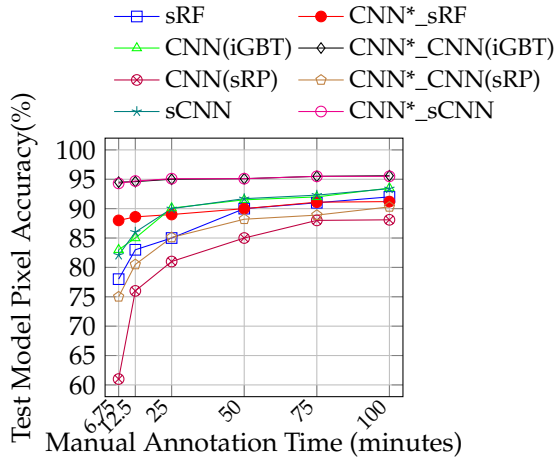
Figure 3.5 displays the visual results of different models on the test set. From the results, it is evident that CNN*_CNN(iGBT) exhibits superior segmentation performance.

Table 3.5: IoU results on RGB canopy images under different annotation times (minutes). In total, 280 test samples are considered.

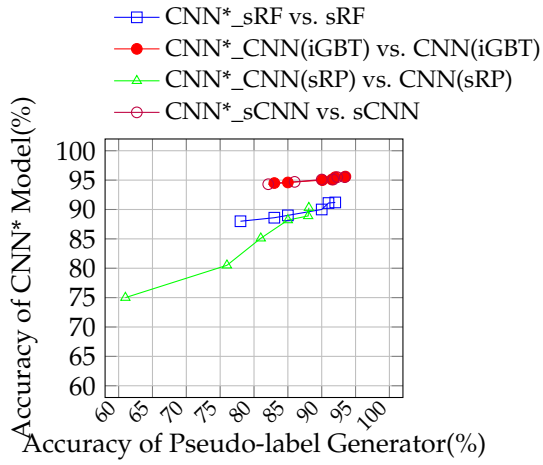
	Foreground IoU (%)					
Annotation Time	**6.75**	**12.5**	**25**	**50**	**75**	**100**
sRF	63.0	69.5	72.3	81.7	83.3	84.2
CNN*_sRF	79.6	80.2	81.8	82.8	83.5	83.7
CNN(iGBT)	69.5	72.4	81.8	83.5	84.3	86.9
CNN*_CNN(iGBT)	88.7	89.2	89.6	89.9	90.7	90.8
CNN(sRP)	42.9	67.2	74.2	78.7	79.9	82.2
CNN*_CNN(sRP)	61.3	68.1	78.6	80.2	81.8	83.5
sCNN	69.3	72.6	81.9	83.3	84.3	86.9
CNN*_sCNN	**89.0**	**89.4**	**89.6**	**89.9**	**90.8**	**91.0**

3.5 Conclusions

In our quest to address the challenge of training CNN models with limited manual annotation resources, we delve into the fusion of traditional machine learning techniques with deep learning, focusing on predicting



(a)



(b)

Figure 3.4: (a) Test accuracy vs. manual annotation time on RGB canopy images, for the models defined in Table 3.1. (b) CNN* test accuracy vs. test accuracy obtained by its corresponding pseudo-label generator on RGB canopy images, for the models defined in Table 3.1.

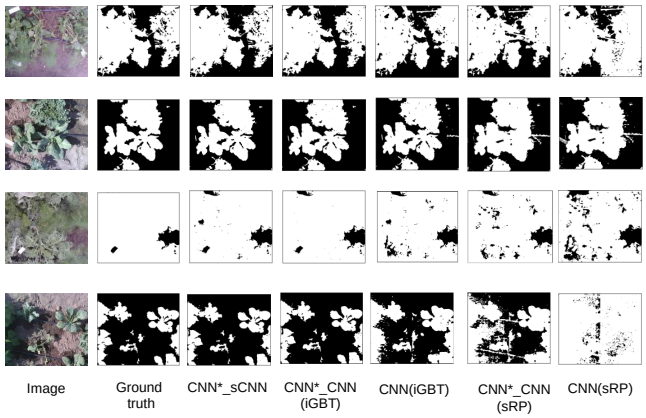


Figure 3.5: Qualitative results on RGB canopy images as obtained with a 6.75 min annotation time. See Table 3.1 for the model names definition.

pseudo-labels. We investigate two scenarios that are representative of typical simple (no occlusion, reasonably stable appearance) and complex (shape variations and occlusions) object image segmentation problems. Through experiments across two datasets, we discover that integrating unlabeled data via pseudo-labels emerges as a potent strategy, markedly enhancing model efficacy compared to models trained solely on manually labeled samples. We extensively investigate the relation between (i) the strategy adopted to collect manual annotations and turn them into a pseudo-label generator, and (ii) the benefit drawn from pseudo-labels. Our study un-

Table 3.6: Pixel accuracy results on RGB canopy images under different annotation times (minutes).

	Pixel Accuracy (%)					
Annotation Time	**6.75**	**12.5**	**25**	**50**	**75**	**100**
sRF	78.3	83.2	85.1	89.7	91.0	91.8
CNN*_sRF	88.1	88.6	89.1	90.0	91.1	91.2
CNN(iGBT)	83.1	85.2	90.1	91.5	92.0	93.5
CNN*_CNN(iGBT)	94.5	94.6	95.0	**95.1**	**95.5**	**95.6**
CNN(sRP)	61.1	75.9	81.3	85.2	87.8	88.1
CNN*_CNN(sRP)	75.2	80.5	85.1	88.2	88.9	90.3
sCNN	82.1	85.9	90.0	91.7	92.3	93.4
CNN*_sCNN	**94.3**	**94.7**	**95.1**	**95.1**	**95.5**	95.5

derscores the significance of pseudo-label diversity. Enriching the pseudo-label generation can be achieved explicitly by considering multiple pseudo-labels per image or implicitly through interactive annotation methods to select and annotate diverse samples to train a pseudo-label generator that offers high expressivity (thereby avoiding regularizing/averaging the diversity inherent to selected samples). Our experiments reveal that increased diversity improves the data-reinforced model accuracy, both at constant pseudo-label generator accuracy and at constant annotation quantity, when the annotation quantity is measured based on the annotation time. These insights collectively enrich our understanding of the advantages and tactics associated with harnessing pseudo-labels, offering valuable guidance for improving model performance when human resources are limited for annotation.

Chapter 4

Using pure pollen species when training a CNN to segment pollen mixtures

Recognizing the types of pollen grains and estimating their proportion in pollen mixture samples collected in a specific geographical area is important for agricultural, medical, and ecosystem research. In the publication [YJJ+22b], on which this chapter is based, we adopt a convolutional neural network for the automatic segmentation of pollen species in microscopy images, and proposes an original strategy to train such network at reasonable manual annotation cost. Our approach is founded on a large dataset composed of pure pollen images. It first (semi-)manually segments foreground, i.e. pollen grains, and background in a fraction of those images, and use the resulting annotated dataset to train a universal pollen segmentation CNN. In the second step, this model is used to automatically segment a large number of additional pure pollen images, so as to supervise the training of a pollen species segmentation model. Despite the fact that it has been trained from pure images only, the model is shown to provide accurate segmentation of species in pollen mixtures. Our experiments also demonstrate that dedicating a model to the segmentation of a subset of the available pure pollen species makes it possible to train a bin pollen class, corresponding to pollen species that are not in the subset of species recognized by the model. This strategy is useful to cope with unexpected species in a mixture.

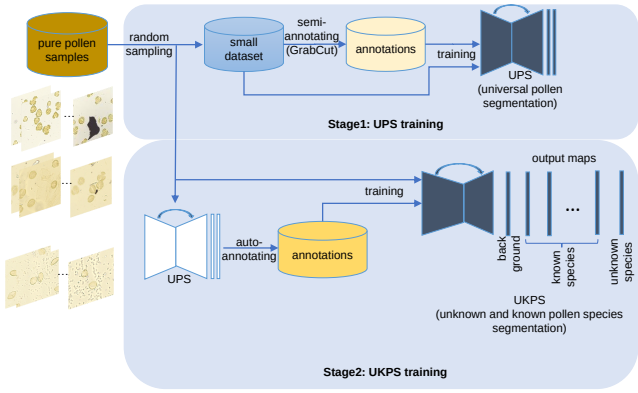


Figure 4.1: Overview of our method. In the first stage, we (semi-)manually annotate a small subset using the GrabCut algorithm and then train UPS model capable of distinguishing pollen grains and background. In the second stage, the UPS model is used to automatically annotate a large number of additional pure pollen images to train UKPS model, which can differentiate between known pollen species and unknown species.

4.1 Introduction

Sexual reproduction in flowering plants involves the production, transfer and deposition of pollen grains to fertilise the ovules and thus enable the formation of seeds. The morphology of pollen grains depends on the pollination mode and the identity of the plant species. For example, the pollen of wind-pollinated trees, such as pines, has two air sacs that allow it to be dispersed over long distances. The identification of pollen types allows many applications such as the reconstruction of the history and evolution of ecosystems since the ice ages, including the influences of human populations through pollen trapped in peat accumulations (palaeoecology), alerts of allergic peaks by monitoring pollen counts in the air (allergology), pollen trapped in honey for certification of single-flower or controlled origin honey (melisso-palynology) or pollen transported and collected by insect pollinators as food resources (pollination ecology and pollinator health).

The traditional methods of recognizing pollen types use light microscopy on fresh pollen, acetolysed pollen, or scanning electron microscopy [KZK+22]. Recognition is based on the size, shape, thickness and ornamentation of the exine, the number and the shape of germination

pits or pores. Such identification is time-consuming and requires considerable recognition expertise. New DNA barcoding techniques are being developed, allowing precise identification from DNA profile banks [GK22]. They are expensive and time-consuming. The development of techniques based on automatic recognition offers a remarkable opportunity and is highly valued by palynology researchers [TBTA11].

Features such as shape [RCF+06], texture [RCF+06], contour [TBTA11] have been used to segment pollen in early machine learning research. With the emergence of deep learning [LBH15], convolutional neural networks [AAMA17] are now widely used due to its ability to learn end-to-end, to directly convert image pixels into predictions of interest. However, deep learning requires a large amount of labeled data [BOT+20], and the annotation of mixture pollen images is extremely difficult. Only a high-level expert can distinguish different pollen in an image. This is due to the relatively high similarity of pollen and to the fact that pollen particles are 3D objects, which might appear quite differently when observed from different angles. Hence, labeling mixed pollen images is a difficult, tedious, and time-consuming task.

In this chapter, as shown in Figure 4.1, a method is proposed to learn how to segment individual pollen species in pollen mixtures, based on a large dataset composed of pure pollen images. To save manual annotation time, our method first trains a universal i.e. species-agnostic, pollen segmentation CNN based on the (semi-)manual segmentation of pollen grains in a small subset of images randomly selected in the dataset. This model is then used to automatically segment a large number of additional pure pollen images, so as to supervise the training of our pollen species segmentation CNN. Despite it is trained on pure images, our model is shown to accurately segment species in pollen mixtures. To handle scenarios for which all possible pollen species are not known/available a priori, we propose an alternative multi-model training strategy, in which each model is trained to recognize only a subset of the species available in the training set, while assigning an 'unknown' label to other species. This strategy makes it possible to segment and identify as 'unknown' a pollen grain that has not been seen during training. To the best of our knowledge, our work is the first one to formulate and address mixture pollen analysis as a segmentation problem. This is in contrast with most previous works, which have to isolate individual pollen grains, before recognizing them based on a classifier.

The rest of the chapter is organized as follows. Section 4.2 surveys the

related works. Section 4.3 describes our proposed method. Section 4.4 demonstrates the relevance of our approach on a dataset including more than one hundred pollen species.

4.2 Related Work

Current pollen classification methods are mainly based on manually extracted features and automatically learned features [dGBBdS19]. They generally consider images that correspond to a bounding box, including a single pollen grains. In [MNC+15] hand-crafted texture features extracted, to be processed based on Fisher Discriminant Analysis (FDA) [YLCZ17], to finally classify the pollen with K-Nearest Neighbors (KNN) [ZLZ+17]. The authors in [TBTA11, BOT+20] implemented pollen classification on single pollen grain dataset by investigating different contour and texture extraction methods. Since the manual selection of appropriate feature requires the combined expertise of biologists and computer vision scientists, it remains a tedious task, especially because it needs to be repeated each time a novel pollen species has to be handled. Therefore, deep learning has been considered to define features automatically, based on examples.

In [GPEM21, dGBBdS19, SA18], researchers proposed to use convolutional neural networks to classify pollen images, but in these methods, the image background in the dataset is uniform, and there is only one pollen grain in the images. The above method fails if there are multiple overlapping pollen grains or multiple pollen types in the image, or when there are pollen types in the image that are unknown to the model. To address this limitation, we propose to formulate the pollen recognition problem as a pixel-wise[DMMJ16] segmentation problem. In this formulation, the CNN predicts a label in each image pixel. This label determines whether the pixel covers the background or a pollen grain and, in the latter, specifies to which pollen species the pixel corresponds. However, training such a CNN using conventional supervision methods, whilst possible and effective [LSD15, KHG+19], requires the annotation of a large amount of images containing the various pollen species that have to be recognized by the model. Such annotation of pollen mixture images requires biological expertise, which is time-consuming and error-prone.

In order to solve the tedious task of labeling large dataset, we first (semi-)manually segment foreground pixels, i.e. the pollen grains, on a small set of images, and use the resulting annotations to train a Universal

Pollen Segmentation (UPS) model. This model is then used to automatically segment a large number of additional pure pollen images to supervise a multi-class segmentation model that is designed to differentiate pollen species and, optionally, to identify pollen grains that correspond to species that have not been considered during training.

4.3 Methods

This section introduces the scheme proposed to quantify the amount of individual pollen species in a pollen mixture microscopy image. It relies on pixel-wise labelling of microscopy images, using convolutional neural networks (CNNs). As shown in Figure 4.1, the method considers the training of a universal, species agnostic, pollen segmentation CNN model (named UPS in the following), to automatically annotate pure pollen species images. Those images and their annotations are then used to supervise the training of a model that is able to discriminate between species. Optionally, an additional output label is considered to indicate that a pixel corresponds to an *unknown* species of pollen, i.e. a pollen species that has not been considered during training. Depending whether this *unknown* species option is activated or not, we refer to those species segmentation models as UKPS (for Unknown and Known Pollen Species) or KPS, respectively.

4.3.1 Pollen Grain Segmentation

As depicted in Figure 4.2, our universal pollen segmentation CNN consists in a 9 layer U-Net[RFB15b], followed by two output feature maps connected to a sigmoid to predict a binary value in each image pixel. Training this CNN requires the definition of pollen grain segmentation masks in a number of image samples. This foreground/background segmentation is done (semi-)manually using the GrabCut algorithm[RKB04]. GrabCut takes as minimal input a bounding-box that is manually-defined around the foreground object of interest, and splits the bounding-box inner pixels in two connected regions that most differ in terms of their distribution. The annotation procedure is both effective and efficient since the foreground pollen grains are highly contrasted compared to the background, making GrabCut appropriate for this task. In practice, only a few hours have been required to label 6000 grains in 1000 images.

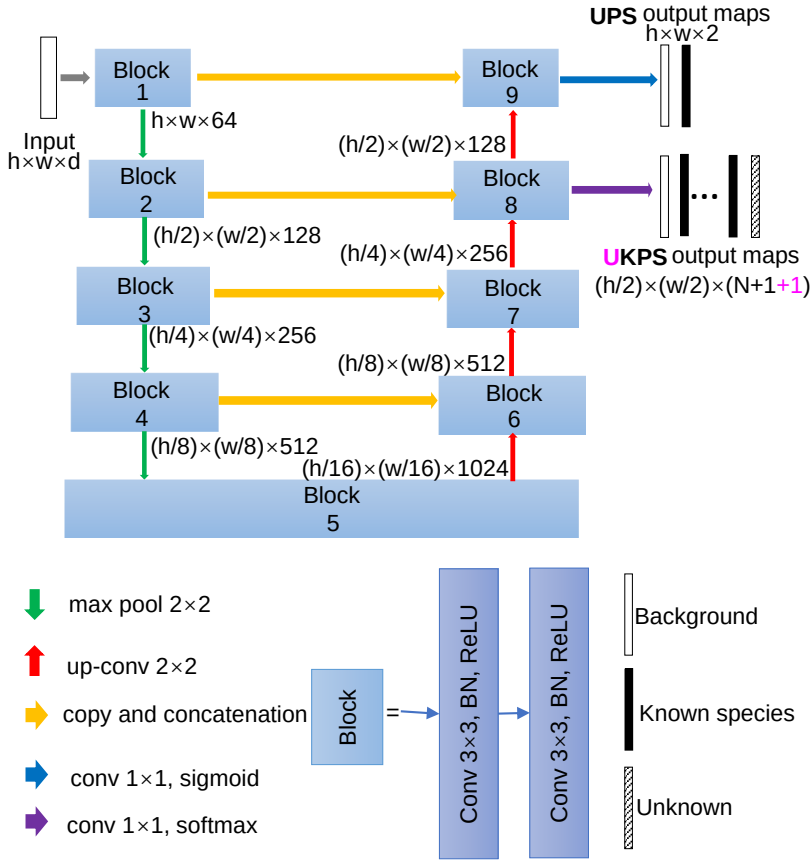


Figure 4.2: Architectures of our models. We have three kinds of models. They use the same architecture for encoder. UPS has 9 blocks, and KPS(without unknown species map compared with UKPS) and UKPS have 8 blocks. In the figure, h, w, d represent the height, width and channel of the input image respectively. N is the number of known pollen species by the model. Block1 - Block9 are convolution+batchnormalization+ReLU.

4.3.2 Pollen Species Segmentation

The CNN adopted to recognize and segment pollen species in pollen mixtures is depicted in Figure 4.2. It adopts a similar U-Net backbone than the one adopted for pollen grain segmentation, but omits the last decoder layer to limit the resolution and thus the memory footprint associated to the output maps of the network. In practice, one output map is devoted to each of the C possible categories that the model is expected to recognize. A softmax operator is applied to the C activations outputted by the network in each pixel, so as to predict a vector with positive components that sum to one. Each of these vectors is close to one-hot vector that is supposed to encode the category of its corresponding pixel.

Two types of pollen species segmentation CNNs have been envisioned in our study, depending on how they define the output categories they aim at recognizing.

The first one, named *known pollen species* (KPS), is certainly the most natural. It assigns one specific output to each of the N pollen species represented in the training set, and considers an additional bin class category to identify the pixels that do not correspond to one of the known species (combining background pixels and unknown pollen grains). However, the experiments presented in Section 4 (particularly Figure 4.5) reveal that such a model fails in labelling a pollen grain that is not part of the N known species properly, i.e. as bin class category.

The second strategy, named *unknown and known pollen species* (UKPS), aims at addressing this limitation. It proposes to partition the N pollen species into K disjoint subsets, and to learn a model for each subset, thereby leading to K models. Specifically, similar to KPS, the model associated to the i^{th} subset assigns one output map to each of the species in the subset but, in contrast to KPS, assigns non-pollen background pixels and unknown pollen species to two distinct outputs. This is made possible since the pollen grains of $(N - N_i)$ species that are not in the i^{th} subset can be used to supervise the training of the unknown pollen category.

4.4 Experiments and Discussion

In this section, we present experiments to demonstrate that our data annotation and training strategies are effective in learning convolutional neural networks that are able to segment known and unknown pollen species in pollen mixtures.

4.4.1 Implementation Details

Dataset. A total of 149 pure pollen files with inconsistent backgrounds as shown in Figure 4.1, and 10 mixture pollen files were scanned and uploaded to Cytomine [MRS+16]. Each file corresponds to a large-scale image. Some of them have resolutions over 30000 by 30000. Each pure pollen file includes a single species of pollen. Those pure files include species from the Asteraceae [HS11], Brassicaceae [SB11], Fabaceae [RST22], Lamiaceae [Raj12], Rosaceae [GFP+20] families (the number of pollen species in each family is 10, 21, 26, 23, and 12, respectively) and 67 species outside this set of families. Large-scale images implies higher computational and memory requirements during model training. Therefore, we crop a large-scale image into multiple 800 by 800 images. We split our 149 pure pollen species into known (131 species, used for training) and unknown (18 species, kept for testing, and never seen at training) species. A small set of ten 800x800 images was randomly selected among the known species to train the UPS network. To train and evaluate the KPS model, about 30 images have been randomly selected for training, 10 images for validation and 10 images for testing. When considering UKPS, we have considered K=5 subsets of known species, corresponding to the Asteraceae, Brassicaceae, Fabaceae, Lamiaceae, and Rosaceae. Each subset was used to learn the known species their respective model, and the species remaining out of the 131 species available for training were used to supervise the unknown pollen class, thereby resulting in 5 models, denoted by the name of their associated family.

Semi-manual annotation. We selected a small dataset containing 10 800x800 images in each of the 131 pure species pollen images. We used Grabcut to segment the pollen grains in those images in an effective manner. It took about 8 hours to annotate the whole set of images.

Data augmentation. The researchers pointed out in [WMP] that the more data the model can access, the more effective the model is. To further increase the amount of data that the model can access, we use random rotate90, flip, transpose, and brightness contrast as data augmentation methods during training.

Loss. The loss function adopted in this paper is defined in equation (4.3). It combines the cross-entropy and the Dice Loss.

$$L_{CE} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C} y_i^c \cdot \log p_i^c \tag{4.1}$$

$$L_{Dice} = 1 - \frac{2\sum_{i=1}^{N}\sum_{c=1}^{C} y_i^c \cdot p_i^c}{\sum_{i=1}^{N}\sum_{c=1}^{C} y_i^{c\,2} + \sum_{i=1}^{N}\sum_{c=1}^{C} p_i^{c\,2}} \tag{4.2}$$

$$Loss = 0.9 \cdot L_{CE} + L_{Dice} \tag{4.3}$$

In equation (4.1), y_i^c is a binary indicator (0 or 1) indicating whether pixel i lies in class c or not, and p_i^c is the softmax or sigmoid output of the network for pixel i and class c.

Metric. Mean intersection over union is a standard metric to assess segmentation models. It calculates the intersection over union ratio of ground-truth and predicted regions for each class, and then averages it over the classes. Formally, it is defined as:

$$\text{mIoU} = \frac{1}{C+1}\sum_{i=0}^{C}\frac{|G_i \cap P_i|}{|G_i \cup P_i|} \tag{4.4}$$

$$= \frac{1}{C+1}\sum_{i=0}^{C}\frac{TP_i}{FN_i + FP_i + TP_i} \tag{4.5}$$

with G_i denoting the ground truth, P_i the model prediction associated to class i, and C the number of classes. TP_i, FN_i and FP_i represent true positive, false negative and false positive detection of pixel belonging to class i.

Training. The model parameters are initialized using Kaiming normal [HZRS15]. We use SGD with 0.9 momentum and 0.0001 weight decay. UPS training uses a batch size of 2, and the initial learning rate is 0.01. The learning rate is reduced by a factor of 0.97 at each epoch, using an exponential learning rate scheduler. To train KPS and UKPS, to include more variety in each batch, we resize the images from 800×800 to 400×400 before network input. The batch size can therefore be increased to 16. The initial learning rate is set to 0.01. The exponential learning rate decay factor is set to 0.99. We use early stopping to stop training if the metric does not improve on the validation set for 30 epochs.

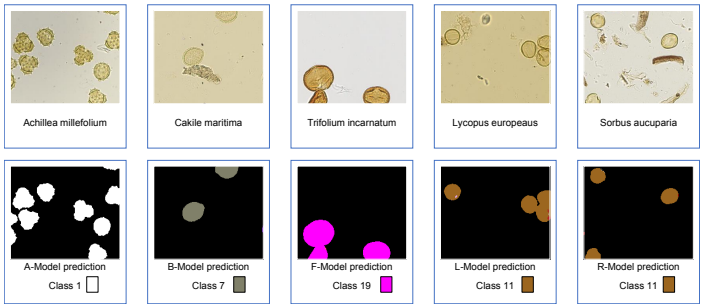


Figure 4.3: Predictions on pure pollen images of known species by our proposed UKPS models. The legend under each prediction defines the UKPS model that has been used (see text for details), and shows the color-label that corresponds to the species present in each image. We can see that while the models are mostly accurate, they have trouble in some cases to correctly segment the borders of pollen grains, given them a different species label (as can be seen by the wrong colors appearing in the last two images)

Hardware. All models are trained with Pytorch [PGM+19a] on Nvidia GeForce GTX 1080 Ti 11Gb GPUs.

4.4.2 Visual Assessment

A few representative examples of pollen species segmentation results are presented in Figure 4.3. For convenience, here we name the family-specific UKPS models based on the first letter of the corresponding family name. Specifically, Asteraceae model, Brassicaceae model, Fabaceae model, Lamiaceae model, and Rosaceae model are referred to as A-Model, B-Model, F-Model, L-Model, and R-Model respectively). We observe that our proposed UKPS can accurately recognize known pollen species in pure images, even though they have trouble in some cases to correctly segment the borders of pollen grains(as can be seen in the last two images in 4.3).

More interestingly, Figure 4.4 considers the segmentation of pollen mixtures. There, we observe that the proposed UKPS can successfully segment known pollen species seen by the model in the mixed pollen grain images, while only ever having been trained on pure pollen images. 4.4a shows the ground truth and corresponding prediction for the image containing two pollen species, Tanacetum vulgare (known by A-Model) and Lotus corniculatus species (known by F-Model). In 4.4b, the input image includes

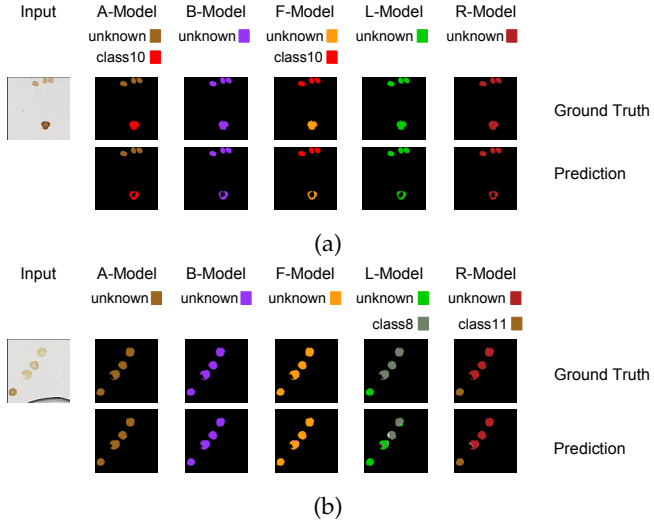


Figure 4.4: Visualization of the predictions for images of a mixture of pollen grains using multiple family-specific models based on UKPS. (a) There are two types of pollen: Tanacetum vulgare (class 10, in the Asteraceae family) and Lotus corniculatus (class 10, in the Fabaceae family). They are correctly classified as known by their specific models (A and F), while they are classified as unknown by the other models. (b) As in (a) the two types of pollen, Lamium galeobdolon (class 8, Lamiaceae) and Sorbus aucuparia (class 11, Rosaceae) are segmented as known species only by their respective models (the L-model and R-model).

Lamium galeobdolon (known by L-Model) and Sorbus aucuparia grains (known by R-Model). Although the central region of Tanacetum vulgare grain in 4.4a is misclassified as background and some pixels of Lamium galeobdolon in 4.4b are segmented as unknown type, it is reasonable because our mixture pollen images and the pure pollen images used for training have been collected and observed with different set-up, which results in significant visual differences between the scan samples provided by biologists. The discrepancy between training and testing images is illustrated by the images presented in 4.6.

A side contribution of this chapter consists in proposing a solution to recognize unknown pollen species as pollen grains, without assigning them to one of the (known) species considered during training. As a baseline, we consider a combination of UPS and KPS. The approach is depicted

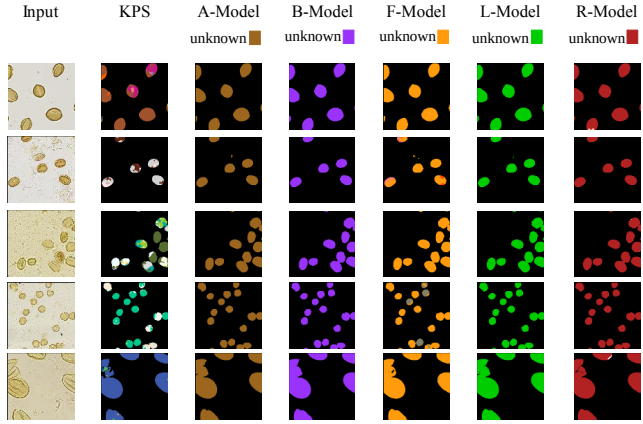


Figure 4.5: Predictions for unknown species using different models.

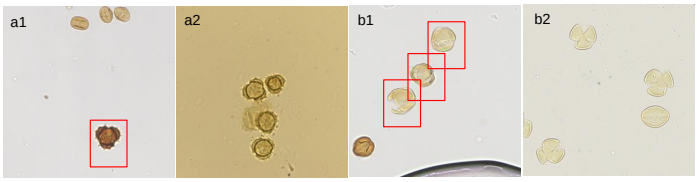


Figure 4.6: The discrepancy between training and testing images. In the figure, a1 and b1 are testing images, a2 is Tanacetum vulgare(boxed in a1) image of the training, and b2 is Lamium galeobdolon(boxed in b1).

in 4.7. The UPS model predicts whether a pixel is part of a pollen grain or not. Instead, KPS differentiates the known pollen species, and is supposed to assign pixels that do not correspond to a known species, either because they correspond to background or to an unknown pollen species, to the additional bin class of KPS. Hence, as shown in 4.7, combining the predictions from UPS and KPS should make it possible to identify unknown pollen species, which are the pixels labelled as pollen by UPS and bin class by KPS.

4.5 shows the predictions of five unknown pollen species. The KPS of baseline can classify N known pollen species, and is supposed to assign a bin class label to pixels that do not correspond to a known pollen species, either because they correspond to background or are unknown. We however observe in 4.5 that KPS segments unknown species as known types rather than the background. In contrast, each UKPS model appears to suc-

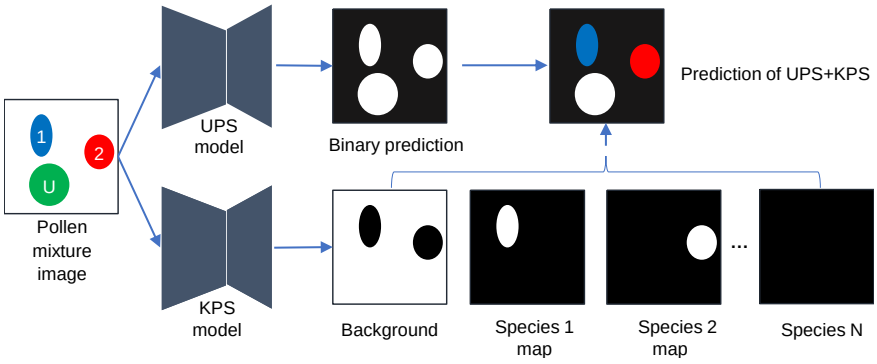


Figure 4.7: Baseline.The UPS model predicts whether a pixel is part of a pollen grain or not. KPS differentiates the known pollen species, and is supposed to assign pixels that do not correspond to a known species as background. Green pollen grain is unknown in the figure, so it should be background in the prediction of KPS.

cessfully segment unseen species into the appropriate unknown class.

4.4.3 Quantitative Assessment

To provide a quantitative evaluation of our method, we have tested the baseline and five UKPS models. The IoU of each class is shown in Figure 4.8, 4.9, 4.10, 4.11, 4.12 generated based on test images that only contain pollen species that are available at training (meaning from the 131 species considered when training KPS and UKPS). We observe that, in general,

	IoU (%)	
Model	Background	Unknown
UPS+KPS(see 4.7)	88.06	5.71
Asteraceae model	99.26	94.97
Brassicaceae model	99.13	69.20
Fabaceae model	99.21	88.82
Lamiaceae model	99.24	90.98
Rosaceae model	99.18	91.19

Table 4.1: The IoU of background and unknown type on unknown pollen species images.

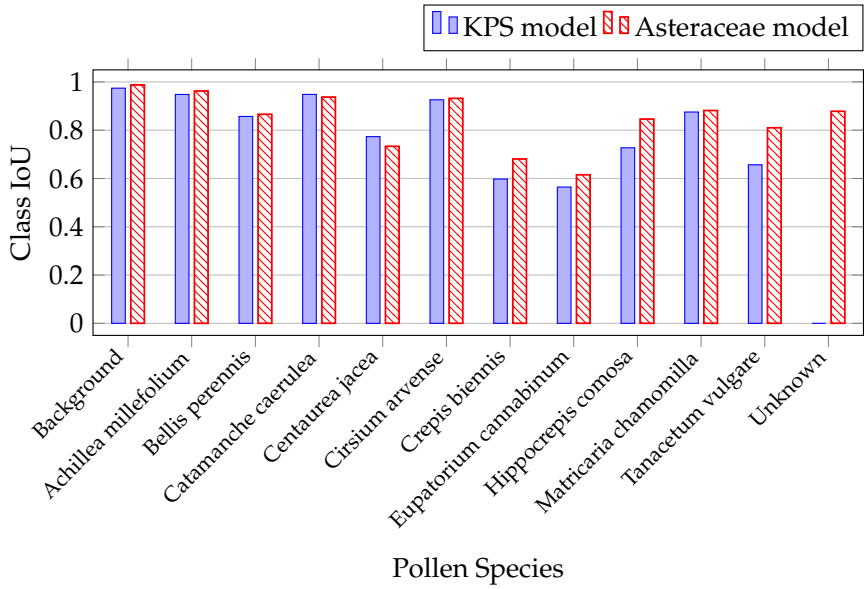


Figure 4.8: Class IoU of the Asteraceae family using KPS and Asteraceae model

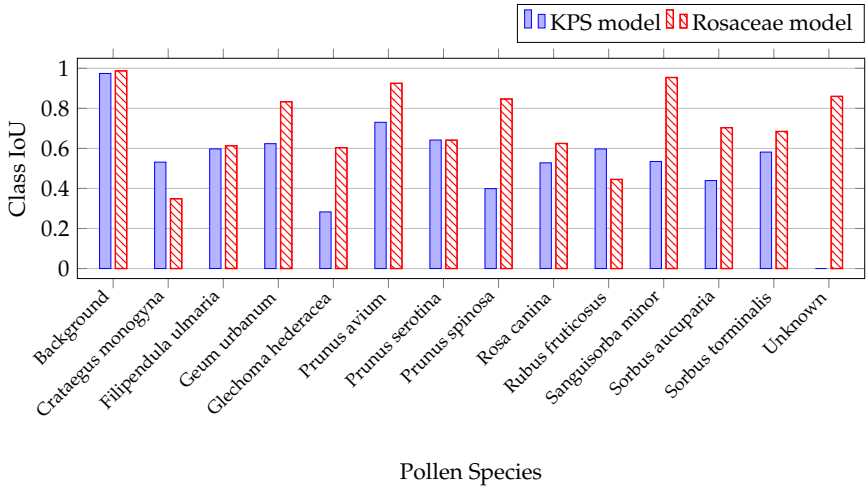


Figure 4.9: Class IoU of the Rosaceae family using KPS and Rosaceae model

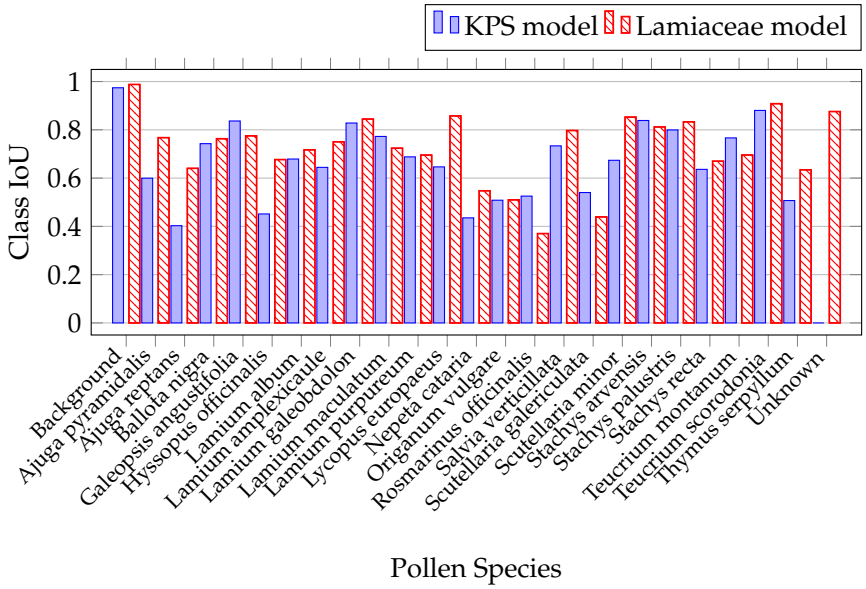


Figure 4.10: Class IoU of the Lamiaceae family using KPS and Lamiaceae model

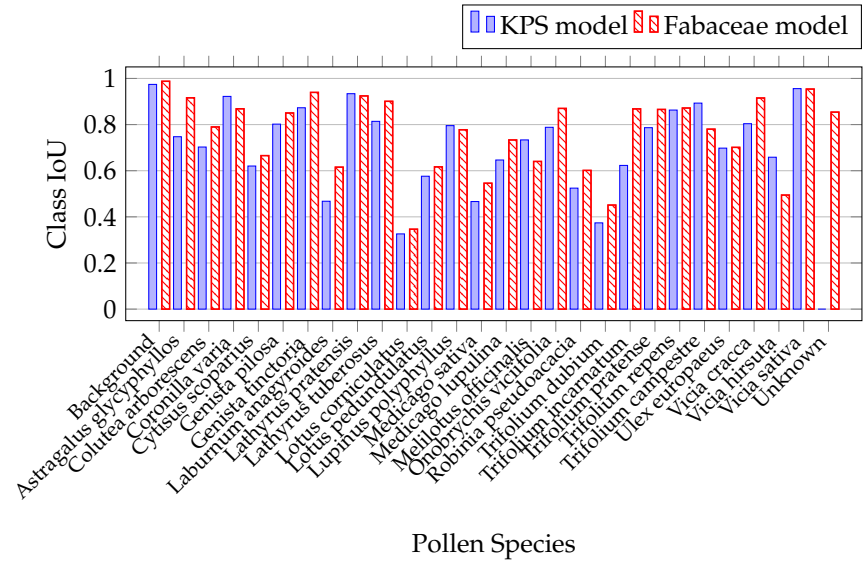


Figure 4.11: Class IoU of the Fabaceae family using KPS and Fabaceae model

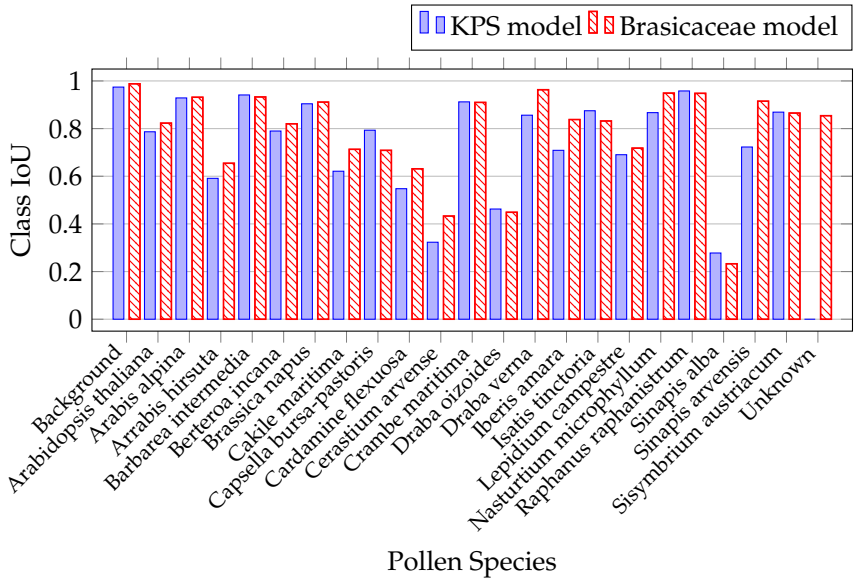


Figure 4.12: Class IoU of the Brasicaceae family using KPS and Brasicaceae model

most of the species that each UKPS model can see in the corresponding family has a slightly higher IoU than the baseline, and what is important is that the UKPS model can identify unknown species well. And the baseline can theoretically identify unknown pollen types, but experiments show that it fails. 4.1 presents the IoU of background and unknown category on the whole 18 absolutely unknown species. The results show that our proposed UKPS can segment unknown pollen types well, compared with the baseline.

4.5 Conclusion

With the wide application of deep learning in computer vision, automatic pollen segmentation becomes realistic. However, annotating pollen grain images is a very time-consuming task, especially when dealing with pollen mixtures. Moreover, the dataset collected can not include all pollen species, and there is some bias between the training set and that in the real application. In this chapter, we propose to train a universal pollen segmentation model using a small set of data that can be annotated at reasonable cost.

This model is then used to automatically annotate a large dataset of images that have the particularity to include a single pollen species. Hence, the pollen segmentation mask actually provides a pollen species segmentation, which can be used to train a pollen species segmentation model. The UKPS can successfully assign pollen species unseen by the model into unknown class rather than known classes. The resulting model is shown to provide accurate segmentation of species, including in pollen mixtures. Our experiments also demonstrate that dedicating a model to the segmentation of a subset of the pure pollen species available at training makes it possible to train a bin pollen class, dedicated to all pollen classes (including the one unavailable at training) that are not part of the model-specific subset of species.

Chapter 5

Graph-cut-assisted CNN training for pulmonary embolism segmentation

We present a novel algorithm for pulmonary embolism segmentation, designed to alleviate the need for expert annotation. Our approach integrates deep learning with a conventional image segmentation techniques, operating in two distinct stages. Specifically, graph cut is used for initial segmentation, followed by manual refinement, to define the labels required to train a CNN. This CNN is then employed to generate pseudo-labels on a large dataset, enabling the training of an improved CNN*. Our findings demonstrate enhanced performance of CNN* over CNN. Overall, the CNN* builds on a very limited amount of manual intervention. Moreover, the injection of expert knowledge in the graph-cut avoids the need for expert knowledge in this manual intervention.

5.1 Introduction

Pulmonary embolism (PE) typically results from the obstruction of a pulmonary artery by a blood clot. These thrombi originate from deep veins in the lower extremities, dislodging and traversing through the bloodstream to occlude the pulmonary arteries. PE is the third most common cardiovascular condition, with its annual incidence rate showing an increasing

trend. Timely diagnosis and intervention play pivotal roles in mitigating morbidity associated with PE.

Computed tomography pulmonary angiography (CTPA) is widely used in the diagnosis of PE[HLGR10]. However, each patient's CTPA typically comprises numerous images, rendering the manual review process by physicians laborious and susceptible to errors. Automatic computer vision methods have thus been considered to segment PE. A PE binary segmentation map identifies the scan voxels that correspond to arteries whose oxygenation is affected by the presence of a clot. Early approaches have implemented segmentation of blood vessels, followed by discrimination between PEs and non-PEs based on features such as blood vessel intensity and size[ÖOŞB14]. Recently, [LYW22] introduced a 2D CNN model for PE segmentation. However, the model's training necessitates a substantial volume of annotated data. Collecting a representative set of data is expensive in terms of human expertise, due to the diverse nature of PE and its low contrast with surrounding tissues, thereby rendering the annotation task particularly demanding. [PGR+23] proposed an automated PE segmentation approach eliminating the need for manual outlining. However, this method involves manual labeling of the central extra-pulmonary arteries and veins by experts.

In this chapter, we propose to rely on anatomical knowledge to generate the PE annotations required to train a CNN. This is because PE is known to appear as darker voxels in the pulmonary artery. Hence, once pulmonary artery has been segmented, segmenting the PE becomes trivial, and even a non-expert becomes able to correct the segmentation errors resulting from simple intensity thresholding. In our work, the anatomical knowledge is encoded in the form of a set of 3D scan positions where the vascular system components are expected to be located. Those points are exploited to associate scores to each voxels, reflecting their likelihood to be part of one of each anatomical part of interest. A graph-cut algorithm is then considered to regularize those scores and assign anatomical labels to voxels. Optional manual refinement of the extracted PE is envisioned before using a limited number of (semi-)automatically extracted PE maps to train a CNN. The trained model is then applied to automatically define pseudo-labels on a larger, unlabeled dataset. These pseudo-labels are used to train a so-called data-enriched CNN*, which appears to improve the initial CNN.

5.2 Materials and methods

5.2.1 Dataset

The dataset utilized in this study originates from the RSNA/STR Pulmonary Embolism Detection Challenge hosted on Kaggle[SWC+20]. It encompasses 7,279 CTPA scans, manually annotated, with 401 scans identified as containing central PE. Among these 401 scans, 10 scans are kept as a test set, and PE segmentation maps are manually delineated for each test sample. The 391 remaining scans are considered to train a neural network, with minimal manual annotation effort.

5.2.2 Label generator based on graph cuts

Pulmonary embolism typically presents with complex morphology and regions in medical images. Traditional segmentation algorithms often struggle to accurately capture its boundaries, particularly when the edges between the embolic area and surrounding tissues are blurred or have low contrast. The graph cut algorithm [DOIB12] offers an elegant approach to inject prior information to the image intensity values. By minimizing an energy function that incorporates both prior information related to the human anatomy and appearance consistency criterions for the regions-of-interest, it can globally optimize the segmentation, reducing the risk of local optima and producing smooth, accurate segmentation boundaries.

Drawing inspiration from the methodology outlined in [BDVJ+16], we employ graph-cut to segment the cava vena, aorta, pulmonary artery trunk, and background in the chest CT scans. Following the anatomical segmentation, a post-processing based on intensity thresholding is applied to facilitate PE candidate identification, which categorizes regions into two distinct classes: background and PE.

This procedure comprises five stages: (1) preprocessing to enhance the quality of raw CT scans, (2) employing quickshift[VS08] to generate superpixels, followed by (3) estimating class scores for each superpixel by aggregating data, considering intensities, and region characteristics, (4) constructing a graph based on the class scores to connect superpixels, then employing graph cut with α-expansions to turn pixel scores into labels, (5) finally, relying on pixel intensity to identify PE pixels among pulmonary arteries. Some regions that contain PE are sometimes erroneously classified as background. Therefore, we identify high intensity background re-

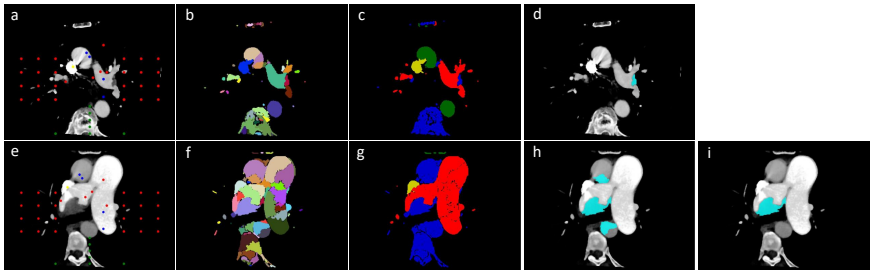


Figure 5.1: Outcomes of the procedure of label generator.(a,e)The knowledge map(points on a 2D plane linked to spine(green), vena cava(yellow), aorta(blue) and pulmonary artery(red)) is applied to the preprocessed result. (b,f) Quickshift result. (c,g) Graph-cut result(background in blue, pulmonary arteries in red, vena cava in yellow, aorta in green). (d,h)Post processed result(PE in cyan). (i)Manually refined result.

gions that are adjacent to regions labeled as pulmonary artery as PE. Since the automatically generated PE maps are not always correct, they are manually refined when needed.

The first stage comprises body and lung masking, and cropping procedures designed to exclude extraneous information beyond the lung boundaries, followed by three intensity-based operations(windowing, histogram equalization and Curvature Anisotropic Diffusion filter[W+98]) aimed at enhancing image quality and diminishing noise. In the second stage, we employ a superpixel approach to seamlessly cluster pixels into coherent regions. This method enhances computational efficiency while ensuring accurate boundary delineation. Within this study, we implement quickshift, a cutting-edge technique, recognized for its capacity to rapidly perform hierarchical segmentation across a spectrum of scales. Results are illustrated in Figure 5.1

In the third stage, our approach entails computing class scores sequentially, as follows:

$$s_b = max(\sigma(I_p); \max_{k \in K_{spine}} (Ker(p\_a, k))) \tag{5.1}$$

where s_b is the score of background, I_p is the intensity of the superpixel p, Ker() is the Gaussian Kernel function measuring similarity, p_a is p's coordinates in the axial plane, the first term is the alpha-beta-sigmoid function [HM01] of I_p , scoring regions with lower intensities as background.

Background encompasses spine and other superpixels not linked to the other three semantic categories. The second is obtained using the knowledge map, a basic guide for anatomical segmentation, with points on a 2D plane (as is shown in Figure 5.1) linked to spine(green), vena cava(yellow), aorta(blue) and pulmonary artery(red). And K_{spine} represents the set of reference points in the knowledge map for the spine.

$$s_{vc} = I_p \cdot Ker(p\_a, VC) \cdot w(z_p) \tag{5.2}$$

In equation 5.2, VC is the vena cava's center in the axial plane. Its value at the scan's top is based on a predefined approximate position common to any scan. Then, other slices update the vena cava along its vertical structure. And $w(z_p)$ is a weight used to decrease the score in vertical regions where the vena cava is unlikely to be found. z_p denotes the vertical component of p

$$S_{aorta} = I_p \cdot w(z_p) \cdot \max_{k \in K_{aorta}} (Ker(p\_a, k)) \tag{5.3}$$

In this equation, K_{aorta} represents the set of reference points in the knowledge map for the aorta.

The scoring function for pulmonary artery is defined as:

$$S_{pa} = I_p \cdot w(z_p) \cdot (\max_{k \in K_{pa}} (Ker(p\_a, k)) + InLung(p)) \tag{5.4}$$

where InLung() provides a binary indication of whether the centroid of a given superpixel p lies within the lung mask, this term is to identify smaller pulmonary arteries distributed throughout the lungs.

Having obtained the score functions for each superpixel and anatomical-semantic class, we proceed to the fourth stage. In this stage, we construct a graph by connecting superpixels horizontally and vertically. Subsequently, we define the weight function for the edges.

Following[BDVJ+16], we define an energy function for graph-cut based image segmentation:

$$E(f) = \sum_{p \in P} D_p(f_p) + \sum_{(p,q) \in N} W(p,q)(1 - \delta(f_p, f_q)) \tag{5.5}$$

where p denotes the superpixel index, with $p \in \{1, 2, \ldots, \text{n\_superpix}\}$.

Each superpixel p is assigned to a class $f_p \in$ {background, pulmonary artery, aorta, vena cava}. The distance $D_p(f_p)$ of a superpixel p to the class f_p is defined as $1 - s_{f_p}(p)$. δ is the Kronecker delta, N includes all pairs of superpixels (p,q) such that p is in P and q is a neighbor of p. W(p,q) defines the energy penalty induced when assigning distinct labels to superpixel p and its neighbor q.

The weight of a vertical edge(between adjacent slices), aiming to assign higher weights to superpixels with larger intersections, is calculated as following:

$$W(p,q) = \frac{n(intersect(p,q))}{min\{n(p), n(q)\}} \tag{5.6}$$

For the horizontal edge weight(within a slice), it is calculated based on the semantic scores of each superpixel.

$$W(p,q) = 1 - \frac{1}{1 + \alpha e^{-\beta M(p,q)}} \tag{5.7}$$

where M(p, q) is defined by:

$$M(p,q) = \min_{k \in \{p,q\}} (s_{M_k}(k) - s_{M2_k}(k)) \tag{5.8}$$

where α and β are parameters to fine-tune the model. M_k and $M2_k$ denote the class with highest and second highest score for superpixel k, respectively. Equation (8) induces a large M, and thus a small penalty which does not discourage a border, only when both pixels have a high score compared to the second best.

Then, we perform graph cut and identify PE. This is followed by manual refinement of the obtained results.

5.2.3 Pseudo-label generator

After obtaining annotated data, we proceed to train a CNN model using supervised learning methods. This model will be utilized to generate pseudo-labels for a large amount of unlabeled data, which will then be used for training CNN*. CNN* is trained on a large dataset(with 361 samples) with pseudo-labels.

time(minutes)		20	40	80	120
CNN	foreground_iou	0.31	0.33	0.40	0.42
	precision	0.48	0.50	0.53	0.57
	sensitivity	0.47	0.50	0.61	0.63
CNN*	foreground_iou	0.32	0.37	0.42	0.44
	precision	0.48	0.53	0.55	0.58
	sensitivity	0.50	0.55	0.67	0.69

Table 5.1: Performance of CNN and CNN* under different manual refining time. Foreground IoU: Measures overlap between predicted and actual areas, calculated as intersection over union (IoU). Precision: Measures accuracy of positive predictions, computed as true positives divided by true positives plus false positives. Sensitivity: Evaluates ability to identify positive samples, determined by true positives divided by true positives plus false negatives.

5.3 Experiment results

We segment 10 scans from the dataset using graph cut and manually refine the segmentation results, as shown in Figure 5.1 . The segmentation process via graph cut is intuitively understandable for humans. It is based on the minimization of a cost function that relies on easily interpretable criteria, such as the similarity between neighboring pixels or the distance to prior values. Therefore, this method introduces a form of transparency in the training process, ensuring that the AI is guided by explicit rules that are comprehensible to humans. The refined results are treated as the test dataset. Subsequently, we manually refine a varying number of images from 30 scans, using 20 minutes(5 scans), 40 minutes(10 scans), 80 minutes(20 scans), and 120 minutes(30 scans), respectively only the manually refined scans are used as training sets for CNN. And CNN* is trained on 361 scans with pseudo labels generated by CNN.

The network architecture employed in this study is based on the classic U-net[RFB15a]. We utilizes Adam as the optimizer and Log-Cosh Dice Loss[Jad20] as the loss function. The batch size is set to 16, and images are resized to 256x256. During training, data augmentation techniques including rotation, flipping, transposition, brightness, and contrast adjustments are applied to enhance data diversity. The initial learning rate for CNN training is set to 0.001, while for CNN* training, it is initialized to 0.00001 with exponential decay learning rate. All models are trained on Nvidia

GeForce GTX 321 1080 Ti 11Gb GPUs

Table 5.1 displays the metrics of foreground IoU, precision, and sensitivity on the test set for CNN and CNN* using different annotation time. Foreground IoU indicates the degree of overlap between the predicted PE and the actual PE. Precision denotes the proportion of pixels predicted as PE that are genuinely PE pixels, while sensitivity reflects the ratio of all true PE pixels that are accurately predicted.

From the experimental results, it is evident that the performance of CNN* has been enhanced compared to CNN. The improvement in precision and sensitivity indicates a reduction in error rates in identifying PE, thus enhancing the model's ability to capture true targets.

5.4 Conclusion

In this chapter, a pulmonary embolism segmentation algorithm that reduces reliance on expert annotation is proposed. The method integrates deep learning with traditional image segmentation and is conducted in two steps. First, graph cuts are employed for preliminary segmentation, followed by minimal manual correction to generate labels for training the CNN. Then, the CNN leverages large-scale data to create pseudo-labels, which are used for further training of an enhanced CNN model (CNN*). The results demonstrate that CNN* outperforms the original CNN. The integration of graph cuts with deep learning aims to provide transparency and guarantees in decision-making by constraining and thus somehow explaining how and why an AI system arrives at certain conclusions. This is particularly important in medical diagnosis, where explainability enhances trust and reliability by providing doctors with insights into the model's reasoning. This approach could potentially serve as a foundation for a new direction in explainable AI, particularly through the injection of domain-specific priors.

Chapter 6

Conclusion

Convolutional neural networks (CNNs) have demonstrated remarkable performance in semantic segmentation, typically relying on extensive labeled datasets. However, obtaining such data in practical applications is challenging and entails significant human and time costs. This thesis explores methods for achieving semantic segmentation with limited manual annotation, aiming to reduce the dependency on extensive labeled datasets.

6.1 Main Results

Our work focuses on three primary approaches to mitigate the need for manual labeling: utilizing pseudo-labels, leveraging single-class samples, and employing graph-cut methods with prior information. Through a series of experiments and analyses, we have demonstrated the effectiveness and practical applicability of these methods.

Chapter 3 addresses the challenge of training convolutional neural networks (CNNs) when labeled data is scarce. We propose a data augmentation approach that combines traditional machine learning and deep learning techniques. Our study focuses on two image segmentation scenarios, representing simple and complex objects, respectively. By experimenting with these two datasets, we examine strategies for collecting manual annotations and transforming them into pseudo-label generators, as well as the advantages provided by pseudo-labels. Our experiments reveal that integrating unlabeled data through pseudo-labels is a highly effective strat-

egy for enhancing model performance. Furthermore, the inclusion of diverse pseudo-labels contributes significantly to the improvement of the model's accuracy and robustness. This chapter demonstrates that leveraging pseudo-labels and unlabeled data can greatly bolster CNN training, especially in situations where labeled data is limited.

In Chapter 4, we explored how to train multi-class segmentation models using single-class images, with a focus on pollen segmentation as a case study. Labeling pollen grain images is a highly time-consuming task, particularly when dealing with mixtures of different pollen species. Furthermore, the collected datasets cannot encompass all pollen species, and there is a notable discrepancy between the training set and the data encountered in real-world applications. Our approach leverages pure pollen species images to train a convolutional neural network (CNN) for the segmentation of pollen mixtures. Specifically, we begin by training a general pollen segmentation model using a small dataset that can be annotated at a reasonable cost. This model is subsequently employed to automatically annotate a large dataset comprising images of single pollen species. The resulting segmentation masks provide detailed pollen species segmentation information, which can then be used to train a specialized pollen species segmentation model. The UKPS model demonstrates the capability to classify previously unseen pollen species as unknown classes, rather than incorrectly assigning them to known classes. The final model exhibits high accuracy in segmenting various pollen species, including those in pollen mixtures. Our experiments further indicate that a model tailored to segment a subset of pure pollen species available during training can effectively train an 'other pollen class'. This class is designed to handle all pollen species not included in the model's specific subset, even those species not present in the training data. In summary, this approach addresses the challenge of high annotation costs while enhancing the model's adaptability and accuracy in practical applications.

In Chapter 5, we propose a novel pulmonary embolism segmentation algorithm aimed at reducing the reliance on expert annotation. Our approach integrates deep learning with traditional image segmentation techniques in a two-stage process. Initially, we use graph cuts for preliminary segmentation, followed by manual refinement to establish the labels necessary for training a CNN. This CNN is subsequently employed to generate pseudo-labels on a large dataset, which are used to train an enhanced CNN*. Our results demonstrate that the enhanced CNN* significantly outperforms the initial CNN, achieving superior segmentation per-

formance with minimal manual intervention. Additionally, incorporating expert knowledge into the graph cuts phase minimizes the need for extensive expert involvement in the manual refinement process. In summary, this chapter highlights how our algorithm effectively leverages limited expert input to achieve high-quality segmentation, showcasing the potential of combining traditional techniques with deep learning to enhance medical image analysis.

Overall, our research demonstrates significant advancements in semantic segmentation, offering practical solutions across various domains. These methods not only reduce the dependency on extensive labeled datasets but also minimize the associated human and time costs. By advancing the state of semantic segmentation with minimal manual annotation, this work contributes to the broader field of computer vision and opens new avenues for efficient and effective data labeling techniques.

6.2 Future perspectives

With these exciting developments, we anticipate future advancements in semantic segmentation and the broader applications of our proposed methods in challenging domains. The findings of this thesis provide a foundation for ongoing research and innovation aimed at achieving high-performance semantic segmentation with reduced reliance on extensive manual annotations. However, several open questions merit further exploration.

Chapter 3 highlights unexpected yet fascinating observations. The experiment suggests that utilizing pseudo-label training with diverse data enhances model performance. However, it prompts the question: is diversity universally beneficial? Under what circumstances does diversity notably improve segmentation outcomes? A deeper understanding of these nuances could refine segmentation methodologies.

In Chapter 4, while the proposed method demonstrates effectiveness with relatively simple shapes like pollen, its suitability for more complex shapes, such as plant leaves in natural environments where different species coexist, remains unclear. Can the model be trained to accurately segment mixed leaf images alongside pure leaf images? This area presents an intriguing challenge for future investigation.

In Chapter 5, efforts to reduce dependency on expert annotations in pulmonary embolism segmentation by integrating prior knowledge with the graph cut algorithm are commendable. Yet, the algorithm's compu-

tational complexity and sensitivity to parameter selection pose significant challenges, particularly for large-scale images. Moving forward, developing a systematic approach to embed prior expert knowledge into models without relying on manually crafted definitions, such as graph cut, remains a critical open problem.

Addressing these challenges will pave the way for advancements in semantic segmentation, making these techniques more robust and applicable across diverse and complex scenarios.

Bibliography

[AAMA17] Saad Albawi, Tareq Abed Mohammed, and Saad ALZAWI. Understanding of a Convolutional Neural Network. August 2017. doi:10.1109/ICEngTechnol.2017.8308186.

[ABB+20] Paolo Andreini, Simone Bonechi, Monica Bianchini, Alessandro Mecocci, and Franco Scarselli. Image generation by gan and style transfer for agar plate image segmentation. *Computer methods and programs in biomedicine*, 184:105268, 2020.

[ACB+21] Paolo Andreini, Giorgio Ciano, Simone Bonechi, Caterina Graziani, Veronica Lachi, Alessandro Mecocci, Andrea Sodi, Franco Scarselli, and Monica Bianchini. A two-stage gan for high-resolution retinal image generation and segmentation. *Electronics*, 11(1):60, 2021.

[AFP+22] Massih-Reza Amini, Vasilii Feofanov, Loic Pauletto, Lies Hadjadj, Emilie Devijver, and Yury Maximov. Self-training: A survey. *arXiv preprint arXiv:2202.12040*, 2022.

[AHY+18] Md Zahangir Alom, Mahmudul Hasan, Chris Yakopcic, Tarek M Taha, and Vijayan K Asari. Recurrent residual convolutional neural network based on u-net (r2u-net) for medical image segmentation. *arXiv preprint arXiv:1802.06955*, 2018.

[AOA+20a] Eric Arazo, Diego Ortego, Paul Albert, Noel E O'Connor, and Kevin McGuinness. Pseudo-labeling and confirmation bias in deep semi-supervised learning. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2020.

[AOA+20b] Eric Arazo, Diego Ortego, Paul Albert, Noel E. O'Connor, and Kevin McGuinness. Pseudo-labeling and confirmation bias in deep semi-supervised learning. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2020. doi:10.1109/IJCNN48605.2020.9207304.

[AZ20] Daniel W Apley and Jingyu Zhu. Visualizing the effects of predictor variables in black box supervised learning models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 82(4):1059–1086, 2020.

[BC22] Maria Baldeon Calisto. Teacher-student semi-supervised approach for medical image segmentation. In *MICCAI Challenge on Fast and Low-Resource Semi-supervised Abdominal Organ Segmentation*, pages 152–162. Springer, 2022.

[BD15] Richard E Bellman and Stuart E Dreyfus. *Applied dynamic programming*, volume 2050. Princeton university press, 2015.

[BDVJ+16] Arnaud Browet, Christophe De Vleeschouwer, Laurent Jacques, Navrita Mathiah, Bechara Saykali, and Isabelle Migeotte. Cell segmentation with random ferns and graph-cuts. In *2016 IEEE International Conference on Image Processing (ICIP)*, pages 4145–4149. IEEE, 2016.

[BKK+19] Stuart Berg, Dominik Kutra, Thorben Kroeger, Christoph N Straehle, Bernhard X Kausler, Carsten Haubold, Martin Schiegg, Janez Ales, Thorsten Beier, Markus Rudy, et al. Ilastik: interactive machine learning for (bio) image analysis. *Nature methods*, 16(12):1226–1232, 2019.

[BKR11] Andrew Blake, Pushmeet Kohli, and Carsten Rother. *Markov random fields for vision and image processing*. MIT press, 2011.

[BLJ+19] Eliott Brion, Jean Léger, Umair Javaid, John Lee, Christophe De Vleeschouwer, and Benoit Macq. Using planning cts to enhance cnn-based bladder segmentation on cone beam ct. In *Medical Imaging 2019: Image-Guided Procedures, Robotic Interventions, and Modeling*, volume 10951, pages 410–417. SPIE, 2019.

[BM98] Avrim Blum and Tom Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on Computational learning theory*, pages 92–100, 1998.

[BOT+20] Sebastiano Battiato, Alessandro Ortis, Francesca Trenta, Lorenzo Ascari, Mara Politi, and Consolata Siniscalco. Detection and Classification of Pollen Grain Microscope Images. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 4220–4227, Seattle, WA, USA, June 2020. IEEE. doi:10.1109/CVPRW50498.2020.00498.

[BT93] Dimitris Bertsimas and John Tsitsiklis. Simulated annealing. *Statistical science*, 8(1):10–15, 1993.

[BY24] Yong Yi Bay and Kathleen A Yearick. Machine learning vs deep learning: The generalization problem. *arXiv preprint arXiv:2403.01621*, 2024.

[CA79] Guy Barrett Coleman and Harry C Andrews. Image segmentation by clustering. *Proceedings of the IEEE*, 67(5):773–785, 1979.

[ÇAL+16] Özgün Çiçek, Ahmed Abdulkadir, Soeren S Lienkamp, Thomas Brox, and Olaf Ronneberger. 3d u-net: learning dense volumetric segmentation from sparse annotation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2016: 19th International Conference, Athens, Greece, October 17-21, 2016, Proceedings, Part II 19*, pages 424–432. Springer, 2016.

[CAS+22] Adithi D Chakravarthy, Dilanga Abeyrathna, Mahadevan Subramaniam, Parvathi Chundi, and Venkataramana Gadhamshetty. Semantic image segmentation using scant pixel annotations. *Machine Learning and Knowledge Extraction*, 4(3):621–640, 2022.

[CCZ+20] Bowen Cheng, Maxwell D Collins, Yukun Zhu, Ting Liu, Thomas S Huang, Hartwig Adam, and Liang-Chieh Chen. Panoptic-deeplab: A simple, strong, and fast baseline for bottom-up panoptic segmentation. In *Proceedings of the*

IEEE/CVF conference on computer vision and pattern recognition, pages 12475–12485, 2020.

[CDI+19] Anthony Cioppa, Adrien Deliege, Maxime Istasse, Christophe De Vleeschouwer, and Marc Van Droogenbroeck. Arthus: Adaptive real-time human segmentation in sports through online distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019.

[CEKK20] Krishna Chaitanya, Ertunc Erdil, Neerav Karani, and Ender Konukoglu. Contrastive learning of global and local features for medical image segmentation with limited annotations. *Advances in neural information processing systems*, 33:12546–12558, 2020.

[CK24] Davide Cacciarelli and Murat Kulahci. Active learning for data streams: a survey. *Machine Learning*, 113(1):185–239, 2024.

[CKNH20] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

[CLY+21] Jieneng Chen, Yongyi Lu, Qihang Yu, Xiangde Luo, Ehsan Adeli, Yan Wang, Le Lu, Alan L Yuille, and Yuyin Zhou. Transunet: Transformers make strong encoders for medical image segmentation. *arXiv preprint arXiv:2102.04306*, 2021.

[CLZJ20] Pengguang Chen, Shu Liu, Hengshuang Zhao, and Jiaya Jia. Gridmask data augmentation. *arXiv preprint arXiv:2001.04086*, 2020.

[CLZZ23] Jiaqi Chen, Jiachen Lu, Xiatian Zhu, and Li Zhang. Generative semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7111–7120, 2023.

[CO16] Kang-Sun Choi and Ki-Won Oh. Subsampling-based acceleration of simple linear iterative clustering for superpixel segmentation. *Computer Vision and Image Understanding*, 146:1–8, 2016.

[CPK+14] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. *arXiv preprint arXiv:1412.7062*, 2014.

[CPK+17] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017.

[CPSA17] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017.

[CSS10] Sonya A Coleman, Bryan W Scotney, and Shanmugalingam Suganthan. Edge detecting for range data using laplacian operators. *IEEE Transactions on Image Processing*, 19(11):2814–2824, 2010.

[CVL18] Mauro Castelli, Leonardo Vanneschi, and Álvaro Rubio Largo. Supervised learning: classification. *por Ranganathan, S., M. Grisbskov, K. Nakai y C. Schönbach*, 1:342–349, 2018.

[CWC+22a] Hu Cao, Yueyue Wang, Joy Chen, Dongsheng Jiang, Xiaopeng Zhang, Qi Tian, and Manning Wang. Swin-unet: Unet-like pure transformer for medical image segmentation. In *European conference on computer vision*, pages 205–218. Springer, 2022.

[CWC+22b] Haihua Chen, Lei Wu, Jiangping Chen, Wei Lu, and Junhua Ding. A comparative study of automated legal text classification using random forests and deep learning. *Information Processing & Management*, 59(2):102798, 2022.

[CZP+18] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 801–818, 2018.

[CZYW17] Xiaohui Chen, Chen Zheng, Hongtai Yao, and Bingxue Wang. Image segmentation using a unified markov random field model. *IET Image Processing*, 11(10):860–869, 2017.

[DDS+09] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[Dec86] Rina Dechter. Learning while searching in constraint-satisfaction problems. 1986.

[DG22] Shasvat Desai and Debasmita Ghose. Active learning for improved semi-supervised semantic segmentation in satellite images. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 553–563, 2022.

[dGBBdS19] André R. de Geus, Celia A.Z. Barcelos, Marcos A. Batista, and Sérgio F. da Silva. Large-scale Pollen Recognition with Deep Learning. In *2019 27th European Signal Processing Conference (EUSIPCO)*, pages 1–5, September 2019. doi:10.23919/EUSIPCO.2019.8902735.

[DMC15] Nameirakpam Dhanachandra, Khumanthem Manglem, and Yambem Jina Chanu. Image segmentation using k-means clustering algorithm and subtractive clustering algorithm. *Procedia Computer Science*, 54:764–771, 2015.

[DMMJ16] M. Dyrmann, A. K. Mortensen, H. Midtiby, and R. Jørgensen. Pixel-wise classification of weeds and crops in images by using a Fully Convolutional neural network. *undefined*, 2016.

[DOIB12] Andrew Delong, Anton Osokin, Hossam N Isack, and Yuri Boykov. Fast approximate energy minimization with label costs. *International journal of computer vision*, 96:1–27, 2012.

[Dos20] Alexey Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[DRTT24] Emanuele Dalsasso, Clément Rambour, Nicolas Trouvé, and Nicolas Thome. Merlin-seg: self-supervised despeckling for

label-efficient semantic segmentation. *Computer Vision and Image Understanding*, 241:103940, 2024.

[DSYA24] Fatemeh Daneshfar, Sayvan Soleymanbaigi, Pedram Yamini, and Mohammad Sadra Amini. A survey on semi-supervised graph clustering. *Engineering Applications of Artificial Intelligence*, 133:108215, 2024.

[DWY13] Cai-Xia Deng, Gui-Bin Wang, and Xin-Rui Yang. Image edge detection algorithm based on improved canny operator. In *2013 International Conference on Wavelet Analysis and Pattern Recognition*, pages 168–172. IEEE, 2013.

[EAN20] Nastaran Enshaei, Safwan Ahmad, and Farnoosh Naderkhani. Automated detection of textured-surface defects using unet-based semantic segmentation network. In *2020 IEEE International Conference on Prognostics and Health Management (ICPHM)*, pages 1–5. IEEE, 2020.

[ECC02] Nicolas Eveno, Alice Caplier, and P-Y Coulon. Key points based segmentation of lips. In *Proceedings. IEEE International Conference on Multimedia and Expo*, volume 2, pages 125–128. IEEE, 2002.

[FD05] Daniel Freedman and Petros Drineas. Energy minimization via graph cuts: Settling what is possible. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, pages 939–946. IEEE, 2005.

[FHSR+20] Di Feng, Christian Haase-Schütz, Lars Rosenbaum, Heinz Hertlein, Claudius Glaeser, Fabian Timm, Werner Wiesbeck, and Klaus Dietmayer. Deep multi-modal object detection and semantic segmentation for autonomous driving: Datasets, methods, and challenges. *IEEE Transactions on Intelligent Transportation Systems*, 22(3):1341–1360, 2020.

[FKDS23] Yue Fan, Anna Kukleva, Dengxin Dai, and Bernt Schiele. Revisiting consistency regularization for semi-supervised learning. *International Journal of Computer Vision*, 131(3):626–643, 2023.

[FRD12] Björn Fröhlich, Erik Rodner, and Joachim Denzler. Semantic segmentation with millions of features: Integrating multiple cues in a combined random forest approach. In *Asian conference on computer vision*, pages 218–231. Springer, 2012.

[G+16] Yarin Gal et al. Uncertainty in deep learning. 2016.

[GCZ+24] Jie Gui, Tuo Chen, Jing Zhang, Qiong Cao, Zhenan Sun, Hao Luo, and Dacheng Tao. A survey on self-supervised learning: Algorithms, applications, and future trends. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[GEW06] Pierre Geurts, Damien Ernst, and Louis Wehenkel. Extremely randomized trees. *Machine learning*, 63:3–42, 2006.

[GFP+20] P. Garcia-Oliveira, M. Fraga-Corral, A. G. Pereira, C. Lourenço-Lopes, C. Jimenez-Lopez, M. A. Prieto, and J. Simal-Gandara. Scientific basis for the industrialization of traditionally used plants of the Rosaceae family. *Food Chemistry*, 330:127197, November 2020. doi:10.1016/j.foodchem.2020.127197.

[GK22] Morgan R Gostel and W John Kress. The expanding role of dna barcodes: Indispensable tools for ecology, evolution, and conservation. *Diversity*, 14(3):213, 2022.

[GPAM+20] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

[GPEM21] James A. Grant-Jacob, Matthew Praeger, Robert W. Eason, and Ben Mills. Semantic segmentation of pollen grain images generated from scattering patterns via deep learning. *J. Phys. Commun.*, 5(5):055017, May 2021. doi:10.1088/2399-6528/ac016a.

[GSK18] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728*, 2018.

[GVZI+21] Seyed Abolfazl Ghasemzadeh, Gabriel Van Zandycke, Maxime Istasse, Niels Sayez, Amirafshar Moshtaghpour,

and Christophe De Vleeschouwer. Deepsportlab: a unified framework for ball detection, player instance segmentation and pose estimation in team sports scenes. *arXiv preprint arXiv:2112.00627*, 2021.

[HCX+22] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009, 2022.

[HLGR10] Andetta R Hunsaker, Michael T Lu, Samuel Z Goldhaber, and Frank J Rybicki. Imaging in acute pulmonary embolism with special clinical scenarios. *Circulation: Cardiovascular Imaging*, 3(4):491–500, 2010.

[HM01] Qiang Huo and Bin Ma. Online adaptive learning of continuous-density hidden markov models based on multiple-stream prior evolution and posterior pooling. *IEEE transactions on speech and audio processing*, 9(4):388–398, 2001.

[HMKS19] Dan Hendrycks, Mantas Mazeika, Saurav Kadavath, and Dawn Song. Using self-supervised learning can improve model robustness and uncertainty. *Advances in neural information processing systems*, 32, 2019.

[HS11] P. P. J. Herman and N. Swelankomo. Asteraceae. *Bothalia*, 41(1):176–178, December 2011. doi:10.4102/abc.v41i1.39.

[HUAM+19] Markus Hofmarcher, Thomas Unterthiner, José Arjona-Medina, Günter Klambauer, Sepp Hochreiter, and Bernhard Nessler. Visual scene understanding for autonomous driving using semantic segmentation. *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, pages 285–296, 2019.

[HZRS15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1026–1034, 2015.

[HZRS16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[IR20] Nabil Ibtehaz and M Sohel Rahman. Multiresunet: Rethinking the u-net architecture for multimodal biomedical image segmentation. *Neural networks*, 121:74–87, 2020.

[Jad20] Shruti Jadon. A survey of loss functions for semantic segmentation. In *2020 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)*, pages 1–7. IEEE, 2020.

[JGR+18] Feng Jiang, Aleksei Grigorev, Seungmin Rho, Zhihong Tian, YunSheng Fu, Worku Jifara, Khan Adil, and Shaohui Liu. Medical image semantic segmentation based on deep learning. *Neural Computing and Applications*, 29:1257–1265, 2018.

[JSX+17] Chao Jin, Fei Shi, Dehui Xiang, Lichun Zhang, and Xinjian Chen. Fast segmentation of kidney components using random forests and ferns. *Medical physics*, 44(12):6353–6363, 2017.

[JYQS22] Qiuye Jin, Mingzhi Yuan, Qin Qiao, and Zhijian Song. One-shot active learning for image segmentation via contrastive learning and diversity-based sampling. *Knowledge-Based Systems*, 241:108278, 2022.

[Kel03] Carl T Kelley. *Solving nonlinear equations with Newton's method*. SIAM, 2003.

[KGLK21] Muhammad Zubair Khan, Mohan Kumar Gajendran, Yugyung Lee, and Muazzam A Khan. Deep neural architectures for medical image semantic segmentation. *IEEE Access*, 9:83002–83024, 2021.

[KHG+19] Alexander Kirillov, Kaiming He, Ross Girshick, Carsten Rother, and Piotr Dollár. Panoptic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9404–9413, 2019.

[KHS10] Rehanullah Khan, Allan Hanbury, and Julian Stoettinger. Skin detection: A random forest approach. In *2010 IEEE international conference on image processing*, pages 4613–4616. IEEE, 2010.

[KKB22] Kenji Kawaguchi, Leslie Pack Kaelbling, and Yoshua Bengio. Generalization in deep learning. In *Mathematical Aspects of Deep Learning*. Cambridge University Press, 2022. doi:10.1017/9781009025096.003.

[KMK+22] Jiwon Kim, Youngjo Min, Daehwan Kim, Gyuseong Lee, Junyoung Seo, Kwangrok Ryoo, and Seungryong Kim. Conmatch: Semi-supervised learning with confidence-guided consistency regularization. In *European Conference on Computer Vision*, pages 674–690. Springer, 2022.

[KMR+23] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4015–4026, 2023.

[KN19] Byeongkeun Kang and Truong Q Nguyen. Random forest with learned representations for semantic segmentation. *IEEE Transactions on Image Processing*, 28(7):3542–3555, 2019.

[KNN21] Mithun Kumar Kar, Malaya Kumar Nath, and Debanga Raj Neog. A review on progress in semantic image segmentation and its application to medical images. *SN computer science*, 2(5):397, 2021.

[KP06] Zoltan Kato and Ting-Chuen Pong. A markov random field image segmentation model for color textured images. *Image and Vision Computing*, 24(10):1103–1114, 2006.

[KRA+14] Kamran Khan, Saif Ur Rehman, Kamran Aziz, Simon Fong, and Sababady Sarasvady. Dbscan: Past, present and future. In *The fifth international conference on the applications of digital information and web technologies (ICADIWT 2014)*, pages 232–238. IEEE, 2014.

[KVB88] Nick Kanopoulos, Nagesh Vasanthavada, and Robert L Baker. Design of an image edge detection filter using the sobel operator. *IEEE Journal of solid-state circuits*, 23(2):358–367, 1988.

[KW13] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[KZ+12] Zoltan Kato, Josiane Zerubia, et al. Markov random fields in image segmentation. *Foundations and Trends® in Signal Processing*, 5(1–2):1–155, 2012.

[KZK+22] Kainat Kanwal, Muhammad Zafar, Amir Muhammad Khan, Tariq Mahmood, Qamar Abbas, Fethi Ahmet Ozdemir, Mushtaq Ahmad, Shazia Sultana, Anam Fatima, Aqsa Aziz, et al. Implication of scanning electron microscopy and light microscopy for oil content determination and seed morphology of verbenaceae. *Microscopy research and technique*, 85(2):789–798, 2022.

[Lam21] Alex Lamb. A brief introduction to generative models. *arXiv preprint arXiv:2103.00265*, 2021.

[LBBH98] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[LBD+20] Jean Léger, Eliott Brion, Paul Desbordes, Christophe De Vleeschouwer, John A Lee, and Benoit Macq. Cross-domain data augmentation for deep-learning-based male pelvic organ segmentation in cone beam ct. *Applied Sciences*, 10(3):1154, 2020.

[LBH15] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, May 2015. doi:10.1038/nature14539.

[LBJ+18] Jean Léger, Eliott Brion, Umair Javaid, John Lee, Christophe De Vleeschouwer, and Benoit Macq. Contour propagation in ct scans with convolutional neural networks. In *Advanced Concepts for Intelligent Vision Systems: 19th International Conference, ACIVS 2018, Poitiers, France, September 24–27, 2018, Proceedings 19*, pages 380–391. Springer, 2018.

[Li09] Stan Z Li. *Markov random field modeling in image analysis*. Springer Science & Business Media, 2009.

[LIIU16] Jinho Lee, Brian Kenji Iwana, Shouta Ide, and Seiichi Uchida. Globally optimal object tracking with fully convolutional networks. *arXiv preprint arXiv:1612.08274*, 2016.

[LLJ+19] Tao Lei, Peng Liu, Xiaohong Jia, Xuande Zhang, Hongying Meng, and Asoke K Nandi. Automatic fuzzy clustering framework for image segmentation. *IEEE Transactions on Fuzzy Systems*, 28(9):2078–2092, 2019.

[LLKDV22] Jean Léger, Lisa Leyssens, Greet Kerckhofs, and Christophe De Vleeschouwer. Ensemble learning and test-time augmentation for the segmentation of mineralized cartilage versus bone in high-resolution microct images. *Computers in Biology and Medicine*, 148:105932, 2022.

[LLV16] Laszlo Lefkovits, Szidónia Lefkovits, and Mircea-Florin Vaida. Brain tumor segmentation based on random forest. *Memoirs of the Scientific Sections of the Romanian Academy*, 39(1):83–93, 2016.

[LPL+20] Chenghao Liu, Fengqian Pang, Yanlin Liu, Kongming Liang, Xiuli Li, Xiangzhu Zeng, and Chuyang Ye. Semi-supervised brain lesion segmentation using training images with and without lesions. In *2020 IEEE 17th International Symposium on Biomedical Imaging (ISBI)*, pages 279–282. IEEE, 2020.

[LS01] Lifeng Liu and Stan Sclaroff. Region segmentation via deformable model-guided split and merge. In *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, volume 1, pages 98–104. IEEE, 2001.

[LS08] Ma Li and Richard C Staunton. Optimum gabor filter design and local binary patterns for texture segmentation. *Pattern Recognition Letters*, 29(5):664–672, 2008.

[LSD15] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[LSP+18] Wei Liu, Heng Shi, Shang Pan, Yongkun Huang, and Yingbin Wang. An improved otsu multi-threshold image segmentation algorithm based on pigeon-inspired optimization. In *2018 11th international congress on image and signal processing, BioMedical engineering and informatics (CISP-BMEI)*, pages 1–5. IEEE, 2018.

[LYB+19] Jiamin Liu, Jianhua Yao, Mohammadhadi Bagheri, Veit Sandfort, and Ronald M Summers. A semi-supervised cnn learning method with pseudo-class labels for atherosclerotic vascular calcification detection. In *2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019)*, pages 780–783. Ieee, 2019.

[LYW22] Zhenhong Liu, Hongfang Yuan, and Huaqing Wang. Camwnet: An effective solution for accurate pulmonary embolism segmentation. *Medical Physics*, 49(8):5294–5303, 2022.

[LYZ+17] Bing Liu, Xuchu Yu, Pengqiang Zhang, Xiong Tan, Anzhu Yu, and Zhixiang Xue. A semi-supervised convolutional neural network for hyperspectral image classification. *Remote Sensing Letters*, 8(9):839–848, 2017.

[LZG+18] Yubing Li, Jinbo Zhang, Peng Gao, Liangcheng Jiang, and Ming Chen. Grab cut image segmentation based on image region. In *2018 IEEE 3rd international conference on image, vision and computing (ICIVC)*, pages 311–315. IEEE, 2018.

[MDSR+20] Robert Mendel, Luis Antonio De Souza, David Rauber, Joao Paulo Papa, and Christoph Palm. Semi-supervised segmentation based on error-correcting supervision. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIX 16*, pages 141–157. Springer, 2020.

[MLG+18] Radek Mackowiak, Philip Lenz, Omair Ghori, Ferran Diego, Oliver Lange, and Carsten Rother. Cereals-cost-effective region-based active learning for semantic segmentation. *arXiv preprint arXiv:1810.09726*, 2018.

[MM20] Ishan Misra and Laurens van der Maaten. Self-supervised learning of pretext-invariant representations. In *Proceedings*

of the IEEE/CVF conference on computer vision and pattern recognition, pages 6707–6717, 2020.

[MNA16] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *2016 fourth international conference on 3D vision (3DV)*, pages 565–571. Ieee, 2016.

[MNC+15] J. Víctor Marcos, Rodrigo Nava, Gabriel Cristóbal, Rafael Redondo, Boris Escalante-Ramírez, Gloria Bueno, Óscar Déniz, Amelia González-Porto, Cristina Pardo, François Chung, and Tomás Rodríguez. Automated pollen identification using microscopic imaging and texture analysis. *Micron*, 68:36–46, January 2015. doi:10.1016/j.micron.2014.09.002.

[MRCW+19] Rose M Rustowicz, Robin Cheong, Lijing Wang, Stefano Ermon, Marshall Burke, and David Lobell. Semantic segmentation of crop type in africa: A novel dataset and analysis of deep learning methods. In *Proceedings of the IEEE/cvf conference on computer vision and pattern recognition workshops*, pages 75–82, 2019.

[MRS+16] Raphaël Marée, Loïc Rollus, Benjamin Stévens, Renaud Hoyoux, Gilles Louppe, Rémy Vandaele, Jean-Michel Begon, Philipp Kainz, Pierre Geurts, and Louis Wehenkel. Collaborative analysis of multi-gigapixel imaging data using Cytomine. *Bioinformatics*, 32(9):1395–1401, May 2016. doi:10.1093/bioinformatics/btw013.

[MWH16] Oskar Maier, Matthias Wilms, and Heinz Handels. Image features for brain lesion segmentation using random forests. In *Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries: First International Workshop, Brainles 2015, Held in Conjunction with MICCAI 2015, Munich, Germany, October 5, 2015, Revised Selected Papers 1*, pages 119–130. Springer, 2016.

[MWY+22] Yujian Mo, Yan Wu, Xinneng Yang, Feilin Liu, and Yujun Liao. Review the state-of-the-art technologies of semantic segmentation based on deep learning. *Neurocomputing*, 493:626–646, 2022.

[NB07] Rory Tait Neilson and McDonald BN. Image segmentation by weighted aggregation with gradient orientation histograms. In *Southern African Telecommunication Networks and Applications Conference, SATNAC*, 2007.

[NBMS17] Behnam Neyshabur, Srinadh Bhojanapalli, David McAllester, and Nati Srebro. Exploring generalization in deep learning. *Advances in neural information processing systems*, 30, 2017.

[NF16] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *European conference on computer vision*, pages 69–84. Springer, 2016.

[Nik11] Mila Nikolova. Energy minimization methods. *Handbook of mathematical methods in imaging*, pages 138–186, 2011.

[NMI10] Samina Naz, Hammad Majeed, and Humayun Irshad. Image segmentation using fuzzy clustering: A survey. In *2010 6th international conference on emerging technologies (ICET)*, pages 181–186. IEEE, 2010.

[NMT+18] Eric Nalisnick, Akihiro Matsukawa, Yee Whye Teh, Dilan Gorur, and Balaji Lakshminarayanan. Do deep generative models know what they don't know? *arXiv preprint arXiv:1810.09136*, 2018.

[NSH22] Vu-Linh Nguyen, Mohammad Hossein Shaker, and Eyke Hüllermeier. How to measure uncertainty in uncertainty sampling for active learning. *Machine Learning*, 111(1):89–122, 2022.

[NSZ20] Behnam Neyshabur, Hanie Sedghi, and Chiyuan Zhang. What is being transferred in transfer learning? *Advances in neural information processing systems*, 33:512–523, 2020.

[OCLF09] Mustafa Ozuysal, Michael Calonder, Vincent Lepetit, and Pascal Fua. Fast keypoint recognition using random ferns. *IEEE transactions on pattern analysis and machine intelligence*, 32(3):448–461, 2009.

[OHT20] Yassine Ouali, Céline Hudelot, and Myriam Tami. An overview of deep semi-supervised learning. *arXiv preprint arXiv:2006.05278*, 2020.

[ÖOŞB14] Haydar Özkan, Onur Osman, Sinan Şahin, and Ali Fuat Boz. A novel method for pulmonary embolism detection in cta images. *Computer methods and programs in biomedicine*, 113(3):757–766, 2014.

[OSF+18] Ozan Oktay, Jo Schlemper, Loic Le Folgoc, Matthew Lee, Mattias Heinrich, Kazunari Misawa, Kensaku Mori, Steven McDonagh, Nils Y Hammerla, Bernhard Kainz, et al. Attention u-net: Learning where to look for the pancreas. *arXiv preprint arXiv:1804.03999*, 2018.

[OSYO17] Baris U Oguz, Russell T Shinohara, Paul A Yushkevich, and Ipek Oguz. Gradient boosted trees for corrective learning. In *International Workshop on Machine Learning in Medical Imaging*, pages 203–211. Springer, 2017.

[PBK16] Daniel F Polan, Samuel L Brady, and Robert A Kaufman. Tissue segmentation of computed tomography images using a random forest algorithm: a feasibility study. *Physics in medicine & biology*, 61(17):6553, 2016.

[PDV17] Pascaline Parisot and Christophe De Vleeschouwer. Scene-specific classifier for effective and efficient team sport players detection from a single calibrated camera. *Computer Vision and Image Understanding*, 159:74–88, 2017.

[PEPD20] Jizong Peng, Guillermo Estrada, Marco Pedersoli, and Christian Desrosiers. Deep co-training for semi-supervised image segmentation. *Pattern Recognition*, 107:107269, 2020.

[PGM+19a] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In

Advances in Neural Information Processing Systems, volume 32. Curran Associates, Inc., 2019.

[PGM+19b] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

[PGR+23] Jiantao Pu, Naciye Sinem Gezer, Shangsi Ren, Aylin Ozgen Alpaydin, Emre Ruhat Avci, Michael G Risbano, Belinda Rivera-Lebron, Stephen Yu-Wah Chan, and Joseph K Leader. Automated detection and segmentation of pulmonary embolisms on computed tomography pulmonary angiography (ctpa) using deep learning but without manual outlining. *Medical Image Analysis*, 89:102882, 2023.

[Pie10] Matti Pietikäinen. Local binary patterns. *Scholarpedia*, 5(3):9775, 2010.

[PKD+16] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2536–2544, 2016.

[PLK+24] Haonan Peng, Shan Lin, Daniel King, Yun-Hsuan Su, Waleed M Abuzeid, Randall A Bly, Kris S Moe, and Blake Hannaford. Reducing annotating load: Active learning with synthetic images in surgical instrument segmentation. *Medical Image Analysis*, 97:103246, 2024.

[PLP20] Emmanuel Pintelas, Ioannis E Livieris, and Panagiotis Pintelas. A grey-box ensemble model exploiting black-box accuracy and white-box intrinsic interpretability. *Algorithms*, 13(1):17, 2020.

[PT01] Regina Pohle and Klaus D Toennies. Segmentation of medical images using adaptive region growing. In *Medical Imaging 2001: Image Processing*, volume 4322, pages 1337–1346. SPIE, 2001.

[PTB19] Jose Pena, Yumin Tan, and Wuttichai Boonpook. Semantic segmentation based remote sensing data fusion on crops detection. *Journal of Computer and Communications*, 7(7):53–64, 2019.

[Qia19] Kun Qian. Automated detection of steel defects via machine learning based on real-time semantic segmentation. In *Proceedings of the 3rd international conference on video and image processing*, pages 42–46, 2019.

[QYA+23] Imran Qureshi, Junhua Yan, Qaisar Abbas, Kashif Shaheed, Awais Bin Riaz, Abdul Wahid, Muhammad Waseem Jan Khan, and Piotr Szczuko. Medical image segmentation using deep semantic-based methods: A review of techniques, applications and emerging trends. *Information Fusion*, 90:316–352, 2023.

[Raj12] R. R. Raja. Medicinally potential plants of Labiatae (Lamiaceae) family: An overview. *Research Journal of Medicinal Plant*, 6(3):203–213, 2012.

[RCF+06] M. Rodríguez-Damián, E. Cernadas, Arno Formella, Manuel Fernandez-Delgado, and Pilar Sa-Otero. Automatic detection and classification of grains of pollen based on shape and texture. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 36(4):531–542, August 2006. doi:10.1109/TSMCC.2005.855426.

[RDRS21] Mamshad Nayeem Rizve, Kevin Duarte, Yogesh S Rawat, and Mubarak Shah. In defense of pseudo-labeling: An uncertainty-aware pseudo-label selection framework for semi-supervised learning. *arXiv preprint arXiv:2101.06329*, 2021.

[Rek12] Karel Rektorys. *Variational methods in mathematics, science and engineering*. Springer Science & Business Media, 2012.

[RFB15a] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted*

intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18, pages 234–241. Springer, 2015.

[RFB15b] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. In Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro F. Frangi, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241, Cham, 2015. Springer International Publishing. doi:10.1007/978-3-319-24574-4_28.

[RGH+24] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024.

[RHV+19] Ulysse Rubens, Renaud Hoyoux, Laurent Vanosmael, Mehdy Ouras, Maxime Tasset, Christopher Hamilton, Rémi Longuespée, and Raphaël Marée. Cytomine: toward an open and collaborative software platform for digital pathology bridged to molecular investigations. *PROTEOMICS–Clinical Applications*, 13(1):1800057, 2019.

[RKB04] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. "GrabCut": Interactive foreground extraction using iterated graph cuts. *ACM Trans. Graph.*, 23(3):309–314, August 2004. doi:10.1145/1015706.1015720.

[RLL+21] Beanbonyka Rim, Sungjin Lee, Ahyoung Lee, Hyo-Wook Gil, and Min Hong. Semantic cardiac segmentation in chest ct images using k-means clustering and the mathematical morphology method. *Sensors*, 21(8):2675, 2021.

[RM03] Ren and Malik. Learning a classification model for segmentation. In *Proceedings ninth IEEE international conference on computer vision*, pages 10–17. IEEE, 2003.

[RRDR09] Shital Adarsh Raut, M Raghuvanshi, R Dharaskar, and Adarsh Raut. Image segmentation–a state-of-art survey for

prediction. In *2009 international conference on advanced computer control*, pages 420–424. IEEE, 2009.

[RSE+23] Luca Rettenberger, Marcel Schilling, Stefan Elser, Moritz Böhland, and Markus Reischl. Self-supervised learning for annotation efficient biomedical image segmentation. *IEEE Transactions on Biomedical Engineering*, 70(9):2519–2528, 2023.

[RST22] Samuel Paul Raj, Pravin Raj Solomon, and Baskar Thangaraj. Fabaceae. In Samuel Paul Raj, Pravin Raj Solomon, and Baskar Thangaraj, editors, *Biodiesel from Flowering Plants*, pages 291–363. Springer, Singapore, 2022. doi:10.1007/978-981-16-4775-8_19.

[Rud16] Sebastian Ruder. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*, 2016.

[SA18] Víctor Sevillano and José L. Aznarte. Improving classification of pollen grain images of the POLEN23E dataset through three different applications of deep learning convolutional neural networks. *PLOS ONE*, 13(9):e0201807, September 2018. doi:10.1371/journal.pone.0201807.

[SB11] Renate Schmidt and Ian Bancroft. *Genetics and Genomics of the Brassicaceae*. January 2011. doi:10.1007/978-1-4419-7118-0.

[SCZ08] Florian Schroff, Antonio Criminisi, and Andrew Zisserman. Object class segmentation using random forests. In *BMVC*, pages 1–10, 2008.

[SDG23] Jules Sanchez, Jean-Emmanuel Deschaud, and François Goulette. Domain generalization of 3d semantic segmentation in autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 18077–18087, 2023.

[SDV22] Niels Sayez and Christophe De Vleeschouwer. Accelerating the creation of instance segmentation training sets through bounding box annotation. In *2022 26th International Conference on Pattern Recognition (ICPR)*, pages 252–258. IEEE, 2022.

[SDVD+23] Niels Sayez, Christophe De Vleeschouwer, Véronique Delouille, Sabrina Bechet, and Laure Lefèvre. Sunscc: Segmenting, grouping and classifying sunspots from ground-based observations using deep learning. *Journal of Geophysical Research: Space Physics*, 128(12):e2023JA031548, 2023.

[SE21] Hassan Sarmadi and Alireza Entezami. Application of supervised learning to validation of damage detection. *Archive of Applied Mechanics*, 91(1):393–410, 2021.

[SG12] K Somasundaram and SP Gayathri. Brain segmentation in magnetic resonance images using fast fourier transform. In *2012 International Conference on Emerging Trends in Science, Engineering and Technology (INCOSET)*, pages 164–168. IEEE, 2012.

[SG18] Lewis Smith and Yarin Gal. Understanding measures of uncertainty for adversarial example detection. *arXiv preprint arXiv:1803.08533*, 2018.

[SGAR+18] Mennatullah Siam, Mostafa Gamal, Moemen Abdel-Razek, Senthil Yogamani, Martin Jagersand, and Hong Zhang. A comparative study of real-time semantic segmentation for autonomous driving. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 587–597, 2018.

[SHP+22] Abraham George Smith, Eusun Han, Jens Petersen, Niels Alvin Faircloth Olsen, Christian Giese, Miriam Athmann, Dorte Bodin Dresbøll, and Kristian Thorup-Kristensen. Rootpainter: deep learning segmentation of biological images with corrective annotation. *New Phytologist*, 236(2):774–791, 2022.

[SHV18] Zhao Shuyang, Toni Heittola, and Tuomas Virtanen. An active learning method using clustering and committee-based sample selection for sound event classification. In *2018 16th International Workshop on Acoustic Signal Enhancement (IWAENC)*, pages 116–120. IEEE, 2018.

[SJPH15] Jeany Son, Ilchae Jung, Kayoung Park, and Bohyung Han. Tracking-by-segmentation with online gradient boosting de-

cision tree. In *Proceedings of the IEEE international conference on computer vision*, pages 3056–3064, 2015.

[SK19] Connor Shorten and Taghi M Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of big data*, 6(1):1–48, 2019.

[SKZ+21] Andreas Steiner, Alexander Kolesnikov, Xiaohua Zhai, Ross Wightman, Jakob Uszkoreit, and Lucas Beyer. How to train your vit? data, augmentation, and regularization in vision transformers. *arXiv preprint arXiv:2106.10270*, 2021.

[Smi10] A Smith. Image segmentation scale parameter optimization and land cover classification using the random forest algorithm. *Journal of Spatial Science*, 55(1):69–79, 2010.

[Sne95] Ian Naismith Sneddon. *Fourier transforms*. Courier Corporation, 1995.

[SWC+20] Anouk Stein, Carol Wu, Chris Carr, Errol Colak, George Shih, Jeff Rudie, John Mongan, Julia Elliott, Luciano Prevedello, Marc Kohli, Phil Culliton, and Robyn Ball. RSNA STR pulmonary embolism detection. https://www.kaggle.com/competitions/rsna-str-pulmonary-embolism-detection, 2020.

[SYG+21] Veit Sandfort, Ke Yan, Peter M Graffy, Perry J Pickhardt, and Ronald M Summers. Use of variational autoencoders with unsupervised learning to detect incorrect organ segmentations at ct. *Radiology: Artificial Intelligence*, 3(4):e200218, 2021.

[SZ20] Matthias Schonlau and Rosie Yuyan Zou. The random forest algorithm for statistical learning. *The Stata Journal*, 20(1):3–29, 2020.

[SZC+23] Xiangwen Shi, Shaobing Zhang, Miao Cheng, Lian He, Xianghong Tang, and Zhe Cui. Few-shot semantic segmentation for industrial defect recognition. *Computers in Industry*, 148:103901, 2023.

[SZK+17] Si Si, Huan Zhang, S Sathiya Keerthi, Dhruv Mahajan, Inderjit S Dhillon, and Cho-Jui Hsieh. Gradient boosted decision

trees for high dimensional sparse output. In *International conference on machine learning*, pages 3182–3190. PMLR, 2017.

[TBTA11] C. Travieso, J. Briceño, J. R. Ticay-Rivas, and J. B. Alonso. Pollen classification based on contour features. *2011 15th IEEE International Conference on Intelligent Engineering Systems*, 2011. doi:10.1109/INES.2011.5954712.

[Tom12] Carlo Tomasi. Histograms of oriented gradients. *Computer Vision Sampler*, pages 1–6, 2012.

[TS10] Lisa Torrey and Jude Shavlik. Transfer learning. In *Handbook of research on machine learning applications and trends: algorithms, methods, and techniques*, pages 242–264. IGI global, 2010.

[TS23] Alaa Tharwat and Wolfram Schenck. A survey on active learning: State-of-the-art, practical challenges and research directions. *Mathematics*, 11(4):820, 2023.

[ULPG22] Rubén Usamentiaga, Dario G Lema, Oscar D Pedrayes, and Daniel F Garcia. Automated surface defect detection in metals: A comparative review of object detection and semantic segmentation using deep learning. *IEEE Transactions on Industry Applications*, 58(3):4203–4213, 2022.

[UM90] Fatih Ulupinar and Gérard Medioni. Refining edges detected by a log operator. *Computer vision, graphics, and image processing*, 51(3):275–298, 1990.

[VdOKE+16] Aaron Van den Oord, Nal Kalchbrenner, Lasse Espeholt, Oriol Vinyals, Alex Graves, et al. Conditional image generation with pixelcnn decoders. *Advances in neural information processing systems*, 29, 2016.

[VDOKK16] Aäron Van Den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. In *International conference on machine learning*, pages 1747–1756. PMLR, 2016.

[VEH20] Jesper E Van Engelen and Holger H Hoos. A survey on semi-supervised learning. *Machine learning*, 109(2):373–440, 2020.

[VGK+21] Sulaiman Vesal, Mingxuan Gu, Ronak Kosti, Andreas Maier, and Nishant Ravikumar. Adapt everywhere: unsupervised adaptation of point-clouds and entropy minimization for multi-modal cardiac image segmentation. *IEEE Transactions on Medical Imaging*, 40(7):1838–1851, 2021.

[VJB+19] Tuan-Hung Vu, Himalaya Jain, Maxime Bucher, Matthieu Cord, and Patrick Pérez. Advent: Adversarial entropy minimization for domain adaptation in semantic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2517–2526, 2019.

[VKR08] Sara Vicente, Vladimir Kolmogorov, and Carsten Rother. Graph cut based image segmentation with connectivity priors. In *2008 IEEE conference on computer vision and pattern recognition*, pages 1–8. IEEE, 2008.

[VL07] Ulrike Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17:395–416, 2007.

[VS08] Andrea Vedaldi and Stefano Soatto. Quick shift and kernel methods for mode seeking. In *Computer Vision–ECCV 2008: 10th European Conference on Computer Vision, Marseille, France, October 12-18, 2008, Proceedings, Part IV 10*, pages 705–718. Springer, 2008.

[VSP+17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[W+98] Joachim Weickert et al. *Anisotropic diffusion in image processing*, volume 1. Teubner Stuttgart, 1998.

[WFMF02] Koon-Pong Wong, Dagan Feng, Steven R Meikle, and Michael J Fulham. Segmentation of dynamic pet images using cluster analysis. *IEEE Transactions on nuclear science*, 49(1):200–207, 2002.

[WH21] Pengchao Wei and Ronny Hänsch. Random ferns for semantic segmentation of polsar images. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–12, 2021.

[WHMH14] Xi Wang, Ronny Hänsch, Lizhuang Ma, and Olaf Hellwich. Comparison of different color spaces for image segmentation using graph-cut. In *2014 International Conference on Computer Vision Theory and Applications (VISAPP)*, volume 1, pages 301–308. IEEE, 2014.

[WKW16] Karl Weiss, Taghi M Khoshgoftaar, and DingDing Wang. A survey of transfer learning. *Journal of Big data*, 3:1–40, 2016.

[WLC07] Ming-Ni Wu, Chia-Chen Lin, and Chin-Chen Chang. Brain tumor detection using color-based k-means clustering segmentation. In *Third international conference on intelligent information hiding and multimedia signal processing (IIH-MSP 2007)*, volume 2, pages 245–250. IEEE, 2007.

[WMP] Jason Wang, Serra Mall, and Luis Perez. The Effectiveness of Data Augmentation in Image Classification using Deep Learning. page 8.

[WSCM20] Colin Wei, Kendrick Shen, Yining Chen, and Tengyu Ma. Theoretical analysis of self-training with deep networks on unlabeled data. *arXiv preprint arXiv:2010.03622*, 2020.

[WZS+21] Xinlong Wang, Rufeng Zhang, Chunhua Shen, Tao Kong, and Lei Li. Dense contrastive learning for self-supervised visual pre-training. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3024–3033, 2021.

[XFC+20] Shuai Xie, Zunlei Feng, Ying Chen, Songtao Sun, Chao Ma, and Mingli Song. Deal: Difficulty-aware active learning for semantic segmentation. In *Proceedings of the Asian conference on computer vision*, 2020.

[XLBY22] Haiming Xu, Lingqiao Liu, Qiuchen Bian, and Zhen Yang. Semi-supervised semantic segmentation with prototype-based consistency regularization. *Advances in Neural Information Processing Systems*, 35:26007–26020, 2022.

[XLG+20] Zhitao Xiao, Bowen Liu, Lei Geng, Fang Zhang, and Yanbei Liu. Segmentation of lung nodules using improved 3d-unet neural network. *Symmetry*, 12(11):1787, 2020.

[XLHL20] Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V Le. Self-training with noisy student improves imagenet classification. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10687–10698, 2020.

[XWY+21] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. *Advances in neural information processing systems*, 34:12077–12090, 2021.

[XYY+20] Yingda Xia, Dong Yang, Zhiding Yu, Fengze Liu, Jinzheng Cai, Lequan Yu, Zhuotun Zhu, Daguang Xu, Alan Yuille, and Holger Roth. Uncertainty-aware multi-view co-training for semi-supervised medical image segmentation and domain adaptation. *Medical image analysis*, 65:101766, 2020.

[XYZ+17] Pengfeng Xiao, Min Yuan, Xueliang Zhang, Xuezhi Feng, and Yanwen Guo. Cosegmentation for object-based building change detection from high-resolution remotely sensed images. *IEEE Transactions on Geoscience and Remote Sensing*, 55(3):1587–1603, 2017.

[YBS10] Xujiong Ye, Gareth Beddoe, and Greg Slabaugh. Automatic graph cut segmentation of lesions in ct using mean shift superpixels. *International journal of biomedical imaging*, 2010, 2010.

[YJJ+22a] Nana Yang, Victor Joos, Anne-Laure Jacquemart, Christel Buyens, and Christophe De Vleeschouwer. Using pure pollen species when training a cnn to segment pollen mixtures. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 1695–1704, June 2022.

[YJJ+22b] Nana Yang, Victor Joos, Anne-Laure Jacquemart, Christel Buyens, and Christophe De Vleeschouwer. Using pure pollen species when training a cnn to segment pollen mixtures. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1695–1704, 2022.

[YLCZ17] Haishan Ye, Yujun Li, Cheng Chen, and Zhihua Zhang. Fast Fisher Discriminant Analysis with Randomized Algorithms. *Pattern Recognition*, 72, June 2017. doi:10.1016/j.patcog.2017.06.029.

[YLL20] Jie You, Wei Liu, and Joonwhoan Lee. A dnn-based semantic segmentation for detecting weed and crop. *Computers and Electronics in Agriculture*, 178:105750, 2020.

[YM12] Faliu Yi and Inkyu Moon. Image segmentation: A survey of graph-cut methods. In *2012 international conference on systems and informatics (ICSAI2012)*, pages 1936–1941. IEEE, 2012.

[YMW+20] Jining Yan, Lin Mu, Lizhe Wang, Rajiv Ranjan, and Albert Y Zomaya. Temporal convolutional networks for the advance prediction of enso. *Scientific reports*, 10(1):8055, 2020.

[YRJ+24] Nana Yang, Charles Rongione, Anne-Laure Jacquemart, Xavier Draye, and Christophe De Vleeschouwer. On the importance of diversity when training deep learning segmentation models with error-prone pseudo-labels. *Applied Sciences*, 14(12):5156, 2024.

[YWZ+21] Hongtai Yao, Xianpei Wang, Le Zhao, Meng Tian, Zini Jian, Li Gong, and Bowen Li. An object-based markov random field with partition-global alternately updated for semantic segmentation of high spatial resolution remote sensing image. *Remote Sensing*, 14(1):127, 2021.

[YZQ+22] Lihe Yang, Wei Zhuo, Lei Qi, Yinghuan Shi, and Yang Gao. St++: Make self-training work better for semi-supervised semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4268–4277, 2022.

[ZA22] Adrian Ziegler and Yuki M Asano. Self-supervised learning of object parts for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14502–14511, 2022.

[ZBH+21] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning

(still) requires rethinking generalization. *Communications of the ACM*, 64(3):107–115, 2021.

[ZG22] Xiaojin Zhu and Andrew B Goldberg. *Introduction to semi-supervised learning*. Springer Nature, 2022.

[Zho21] Zhi-Hua Zhou. Semi-supervised learning. In *Machine Learning*, pages 315–341. Springer, 2021.

[ZIE16] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III 14*, pages 649–666. Springer, 2016.

[ZLW18] Zhengxin Zhang, Qingjie Liu, and Yunhong Wang. Road extraction by deep residual u-net. *IEEE Geoscience and Remote Sensing Letters*, 15(5):749–753, 2018.

[ZLZ+17] Shichao Zhang, Xuelong Li, Ming Zong, Xiaofeng Zhu, and Debo Cheng. Learning k for kNN Classification. *ACM Transactions on Intelligent Systems and Technology*, 8:1–19, January 2017. doi:10.1145/2990508.

[ZP22] Haixia Zhang and Qingxiu Peng. Pso and k-means-based semantic segmentation toward agricultural products. *Future Generation Computer Systems*, 126:82–87, 2022.

[ZPL+23] Guangjie Zeng, Hao Peng, Angsheng Li, Zhiwei Liu, Chunyang Liu, Philip S Yu, and Lifang He. Unsupervised skin lesion segmentation via structural entropy minimization on multi-scale superpixel graphs. *arXiv preprint arXiv:2309.01899*, 2023.

[ZRSTL18] Zongwei Zhou, Md Mahfuzur Rahman Siddiquee, Nima Tajbakhsh, and Jianming Liang. Unet++: A nested u-net architecture for medical image segmentation. In *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support: 4th International Workshop, DLMIA 2018, and 8th International Workshop, ML-CDS 2018, Held in Conjunction with MICCAI 2018, Granada, Spain, September 20, 2018, Proceedings 4*, pages 3–11. Springer, 2018.

[ZWCK20] Ziang Zhang, Chengdong Wu, Sonya Coleman, and Dermot Kerr. Dense-inception u-net for medical image segmentation. *Computer methods and programs in biomedicine*, 192:105395, 2020.

[ZWH+21] Bowen Zhang, Yidong Wang, Wenxin Hou, Hao Wu, Jindong Wang, Manabu Okumura, and Takahiro Shinozaki. Flexmatch: Boosting semi-supervised learning with curriculum pseudo labeling. *Advances in Neural Information Processing Systems*, 34:18408–18419, 2021.

[ZYCL19] Ri-Gui Zhou, Han Yu, Yu Cheng, and Feng-Xin Li. Quantum image edge extraction based on improved prewitt operator. *Quantum Information Processing*, 18:1–24, 2019.

[ZYH+22] Mingkai Zheng, Shan You, Lang Huang, Fei Wang, Chen Qian, and Chang Xu. Simmatch: Semi-supervised learning with similarity matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14471–14481, 2022.

[ZZK+20] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 13001–13008, 2020.

[ZZL+22] Bin Zhang, Yongjun Zhang, Yansheng Li, Yi Wan, Haoyu Guo, Zhi Zheng, and Kun Yang. Semi-supervised deep learning via transformation consistency regularization for remote sensing image semantic segmentation. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2022.

[ZZW+21] Yi Zhu, Zhongyue Zhang, Chongruo Wu, Zhi Zhang, Tong He, Hang Zhang, R Manmatha, Mu Li, and Alexander J Smola. Improving semantic segmentation via efficient self-training. *IEEE transactions on pattern analysis and machine intelligence*, 2021.

[ZZZ+20] Yuliang Zou, Zizhao Zhang, Han Zhang, Chun-Liang Li, Xiao Bian, Jia-Bin Huang, and Tomas Pfister. Pseudoseg:

Designing pseudo labels for semantic segmentation. *arXiv preprint arXiv:2010.09713*, 2020.