Automatic counting of mounds on UAV images using computer vision and machine learning

Majid Nikougoftar Nategh

A Thesis

in

The Department

of

Concordia Institute for Information Systems Engineering

Presented in Partial Fulfillment of the Requirements

for the Degree of

Master of Applied Science (Quality System Engineering) at

Concordia University

Montréal, Québec, Canada

October 2022

CONCORDIA UNIVERSITY

School of Graduate Studies

This is to certify that the thesis prepared

By: **Majid Nikougoftar Nategh**

Entitled: **Automatic counting of mounds on UAV images using computer vision and machine learning**

and submitted in partial fulfillment of the requirements for the degree of

Master of Applied Science (Quality System Engineering)

complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the Final Examining Committee:

______________________________ Chair and Examiner
Dr. Arash Mohammadi

______________________________Examiner
Dr. Jun Yan

______________________________ Supervisor
Dr. Nizar Bouguila

______________________________ Supervisor
Dr. Wassim Bouachir

Approved by ______________________________
Dr. Abdessamad Ben Hamza, Chair
Department of Concordia Institute for Information Systems Engineering

____________ 2022

____
Mourad Debbabi, Dean
Faculty of Engineering and Computer Science

Abstract

Automatic counting of mounds on UAV images using computer vision and machine learning

Majid Nikougoftar Nategh

Site preparation by mounding is a commonly used silvicultural treatment that improves tree growth conditions by mechanically creating planting microsites called mounds. Following site preparation, an important planning step is to count the number of mounds, which provides forest managers with an estimate of the number of seedlings required for a given plantation block. In the forest industry, counting the number of mounds is generally conducted through manual field surveys by forestry workers, which is costly and prone to errors, especially for large areas. To address this issue, we present a novel framework exploiting advances in Unmanned Aerial Vehicle (UAV) imaging and computer vision to estimate the number of mounds on a planting block accurately. The proposed framework comprises two main components. First, we exploit a visual recognition method based on a deep learning algorithm for multiple object detection by pixel-based segmentation. This enables a preliminary count of visible mounds and other frequently seen objects on the forest floor (e.g., trees, debris, accumulation of water) to be used to characterize the planting block. Second, since visual recognition could be limited by several perturbation factors (e.g., mound erosion, occlusion), we employ a machine learning estimation function that predicts the final number of mounds based on the local block properties extracted in the first stage. We evaluate the proposed framework on a new UAV dataset representing numerous planting blocks with varying features. The proposed method outperformed manual counting methods in terms of relative counting precision, indicating that it has the potential to be advantageous and efficient under challenging situations.

Acknowledgments

First and foremost, I express my sincere gratitude to my supervisors, Prof. Nizar Bouguila and Prof. Wassim Bouachir. Their knowledge and experience were crucial in developing my research. It was my honor to be his student, and I will be forever thankful to them for providing me with this great opportunity.

I had the pleasure of having Ahmed by my side while working on my thesis, and I would like to express my gratitude to him for his advice and support. I would not be able to complete this research without his help.

Last but not least, I want to express my heartfelt gratitude to my wife and family for their support and encouragement throughout my studies in Canada. I would not have been able to finish my education without their unconditional support.

Contents

List of Figures

List of Tables

List of Abbreviations

AI	Artificial Intelligence
AP	Average Precision
COCO	Common Objects in Context
DL	Deep Learning
FPN	Feature Pyramid Network
HOG	Histograms of Oriented Gradients
HWT	Harr wavelet transform
mAP	mean Average Precision
Mask-RCNN	Mask Region-based Convolution Neural Network
ML	Machine Learning
MLP	Multilayer Perceptron
MSP	Mechanical site preparation
NMS	Non-Maximum Suppression
R-CNN	Region-based Convolutional Neural Network
RCP	Relative Counting Precision
ROI	Region of Interest
SVM	Support Vector Machine
SVR	Support Vector Regression
UAV	Unmanned Aerial Vehicle
VIA	VGG Image Annotator
YOLO	You Only Look Once

Chapter 1

Introduction

1.1 Context and research problem

In the domain of silviculture and tree regeneration, several significant limitations prevent optimal survival and early tree plantation yields. These issues include vegetation competition, low soil temperatures in the rooting zone, low soil aeration on wet sites, high water table, and nutrient inadequacy in the rooting zone of the outplants [1]. Positive reaction to alleviate such conditions that restrict tree performance happens only when forestry managers successfully employ site preparation techniques. The aims of these techniques are to modify soil conditions, reduce competing vegetation regrowth intensity [2] [1], and consequently create the appropriate microsites that are ready for planting. In this context, a microsite is defined as a region of a site where the microtopography (the shape of the ground surface), and surface soil components are uniforms [3].

Mechanical site preparation (MSP) has been promoted as a broad category of site preparation, which uses machinery to prepare an area and its soil for tree seedlings [2]. Mounding, scarification, and ripping/ploughing are three most frequent types of MSP currently used in this regard as shown in figure 1.1. In North America, mechanical site preparation by mounding (see figure 1.2) is a widely used technique due to the abiotic and biotic characteristics of North American terrains.

MSP refers to a silvicultural process of preparing a site to construct mounds for planting. The mounds are elevated planting spots that are free of water logging and with little vegetation competition in the soil [2].

A key issue after mounding is to precisely estimate the number of mounds created on each planting block, which corresponds to the number of tree seedlings to be planted. Note that, planting blocks are dynamic due to the influence of macro- and micro-climate factors (see figure 1.3). These components lead to a continuous change in the characteristics of the microsites [4]. In this regard, macroclimate refers to the large-scale atmospheric conditions. Sunlight (solar radiation), precipitation, wind direction and speed, temperature, and humidity of the surrounding air contribute to the macroclimate conditions [5]. The microclimate, in contrast, is the small-scale climate that is greatly influenced by macroclimatic factors [4]. The microclimate can change significantly based on the weather, topography, vegetation cover, and soil characteristics [6].

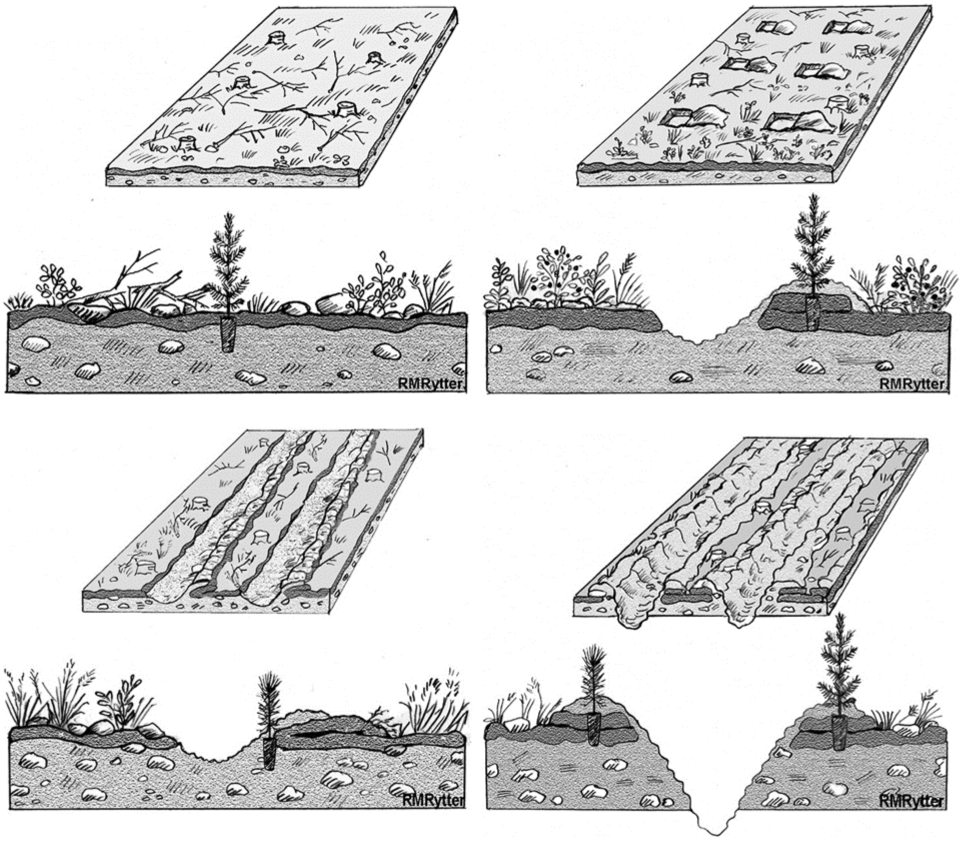


Figure 1.1: Schematic illustrations of mechanical site preparation methods.
Planting with no site preparation (top left), mounding with elevated planting spots (top right), disc trenching in continuous rows (bottom left), and ploughing with relatively deep furrows in rows (bottom right). Figure from [7].

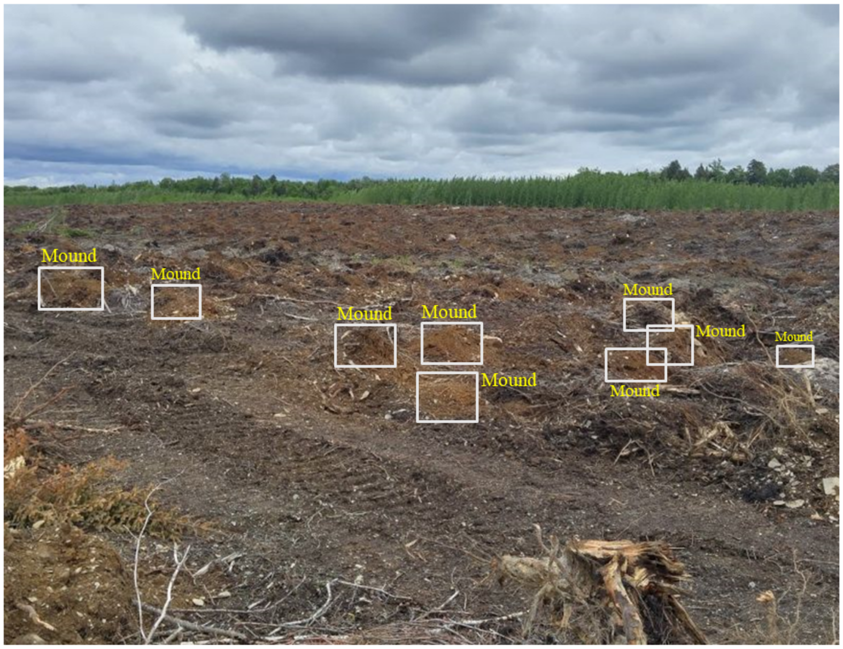


Figure 1.2: Examples of mounds that were constructed mechanically in the balsam fir-white birch bioclimatic domain in Quebec, Canada.

However, when the planting block is mechanically prepared, the field displays noticeable irregularities between different planting microsites (mounds) within the block, as seen in figure 1.2. The resulting mounds are also various in terms of appearance, size, and shape. In addition, complex background, and terrain characteristics, such as overlapping and occlusion of mounds, presence of tree and their shadows on surface of ground, water flow, and remnants of woody debris, create challenges in accurate mound counting.

Manual counting is done to estimate the number of tree seedling to be planted in each block. To do so, forest workers count mechanically created mounds on a section of the site, and use the result to estimate a total number for the entire site, assuming that mound density remains constant on each plantation block. This approach, however, requires forestry workers to go around the field in order to count the visible mounds, which is time consuming, expensive, and prone to human errors. Furthermore, mound density often varies on the same block depending on the characteristics of each zone.

Motivated by recent advances in sensor technology used in drone platforms for data collection, forestry managers also used visual interpretation of Unmanned Aerial Vehicle (UAV) images as an

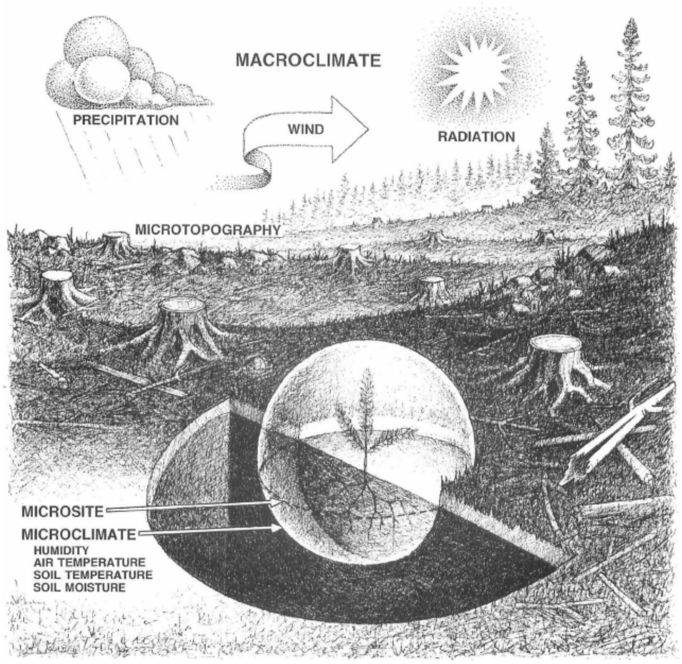


Figure 1.3: Dynamic factors that change the characteristics of microsites.
Figure from [4]

alternative to field manual counting. Image interpretation and analysis are thus performed by human operators, in order to detect, identify, and count mounds on UAV orthomosaics. However, this requires a skilled human interpreter, and due to perception variation among humans on the nature of the objects, data quality, and scale from one site to the next, this method is often inefficient.

1.2 Research objectives

Even though most recent research on aerial sensing has concentrated on UAV-based applications, some critical problems, such as mound counting, were not addressed by taking advantage of advances in Artificial Intelligence (AI) and UAV imagery. Therefore, this thesis aims to fully leverage recent advances in AI and develop a computer vision framework on a UAV platform that makes human work more accessible and efficient.

UAVs are also referred to as Remotely Piloted Aircraft (RPA) or, as the term has become more widely used, drones [6]. UAVs have a number of advantages over piloted aircraft, satellite-based imaging, ground-based sensing, and actuation systems in the forestry practices. Their advantages include being compact, having minimal operational and maintenance costs, requiring less human interaction, being less dangerous when in use, being autonomous, having better controlled imaging with changeable zoom and angle of view, and having higher degrees of agility [8]. These devices are now more widely available and have largely replaced satellites and aircraft in numerous data collection tasks due to their relatively low cost and strong operational capacity. This has greatly reduced the risk and time of manual fieldwork [9]. Using UAV imagery, the objectives of this thesis are presented on two levels.

- **General objective:** The primary goal of our work is to automate counting the number of mounds on planting blocks that have been prepared mechanically. Automating the task of counting mounds reduces errors associated with manual counting, eliminates the need for time-consuming and expensive field surveys, and increases precision by preventing the complicated and inaccurate treatment of seedlings in the field, which may cause financial losses and planting operation delays.
- **Specific objectives:** The secondary goals of this study are as follows:
    1. Construct a dataset of high-resolution UAV images representing several plantation blocks with different characteristics,
    2. Develop a new method for estimating the number of mounds by exploiting advances in computer vision, machine learning, and UAV imagery,
    3. Evaluate the proposed method experimentally on our dataset using appropriate metrics.

1.3 Contributions

The main contributions of this thesis can be categorized into the two following aspects:

1. **Contributions to the forest industry:** This study uses UAV imagery combined with advanced computer vision and ML methods to address an important forestry management problem. Our work has the potential to improve fieldwork conditions and significantly reduce time, money, and resource consumption for forest managers, by automatically estimating the number of planting microsites.

2. **Contributions to the research community:** This study stimulates the interest of the scientific community and contributes to advancing knowledge in the field of object detection and counting on crowded scenes. In this direction, the methods, and procedures resulting from our work were accepted for publication at the IEEE International Conference on Machine Learning and Applications (ICMLA) [10]. We anticipate that disseminating our methods and results will contribute to advancing knowledge on solving real-world problems using machine learning and computer vision methods.

1.4 Method overview

This thesis proposes a new computer vision method for fast and accurate counting of the number of mounds on mechanically prepared planting blocks by integrating two approaches:

1. Local image segmentation using deep learning methods,

2. Patch-level correction by applying regression models.

We initially leverage a two-stage object detector method to perform classification and select the classes of detected objects. To avoid training our model from scratch, we used transfer learning. Then, we applied Mask Region-based Convolution Neural Network (Mask-RCNN) [11] using ResNet101 and Feature Pyramid Network (FPN) as the backbone of the network. This first prediction step is essential for characterizing each image region by quantifying the presence of relevant object instances.

Since mounds can be destroyed or occluded following their creation (e.g., due to erosion, presence of debris, and trees), we cannot rely only on visual detection performed during the first step to estimate their number. Thus, the second step of our method consists of performing a patch-level correction to obtain the final prediction of mound count based on preliminary object counts. In other words, we performed patch-correction based on the visually detected objects of our local segmentation model, to predict a final corrected number of mounds for each patch of a given block.

Finally, we evaluate our framework on a dataset of UAV orthmosaics with various block properties. The obtained results emphasize the importance of using both models sequentially.

1.5 Thesis outline

This thesis is organized as follows. In Chapter 2, we review the most important studies and related works regarding visual object counting based on traditional machine learning and deep learning approaches. Chapter 3 introduces a detailed description of the proposed methodology. Chapter 4 presents experimental results. Finally, chapter 5 concludes the thesis.

Chapter 2

Related Works

In the domain of UAV-based automated technologies, there have been numerous advancements in recent decades. Recent developments in UAVs, including camera capabilities, flight control systems, and navigation systems, have enhanced the usage of UAV images in remote sensing applications [12].

The main advantages of UAVs are their capacity to launch and land vertically, and to measure flight speed and altitude with little interference from the weather. In addition, they have high-resolution cameras with adjustable angles, which enable them to produce their findings quickly and precisely [12, 13, 14]. In recent years, UAV-based remote sensing and cutting-edge Artificial Intelligence (AI), Machine Learning (ML), and Deep Learning (DL) approaches have been deployed increasingly to a variety of applications in forestry and agriculture. This covers many applications including counting crop seedlings [15], ornamental plant detection and counting [16], biomass estimation [17], citrus tree extraction [18], tree species classification [19], fire monitoring [20], and animal counting [21]. Most of these applications are based on visual object counting, also known as crowd counting or crowd density, which is a computer vision task that encompasses all issues and challenges associated with estimating the number of times a specific object appears in an image [22].

Early crowd counting approaches mostly leveraged image processing and computer vision techniques [23, 24, 25]. These technologies have advanced to the point that individuals can now utilize computers to analyze image data and extract information from images. This has led to the emergence of computer vision-based crowd density estimation [26] [27].

2.1 Traditional crowd counting approaches

Traditional Crowd counting methods are based on image processing, and regression learning models. The basic concepts behind these techniques mainly comprise image acquisition, image pre-processing, feature extraction, feature analysis, and classification, as well as the computation of the final crowd density map [28]. The most important phase of the aforementioned techniques is the extraction and analysis of image features.

In literature, crowd counting using traditional image processing techniques relies on hand-crafted features and are divided into two conceptually distinct categories based on the extracted feature category: direct detection approaches and indirect detection approaches. Figure 2.1 presents the taxonomy of traditional crowd counting approaches.

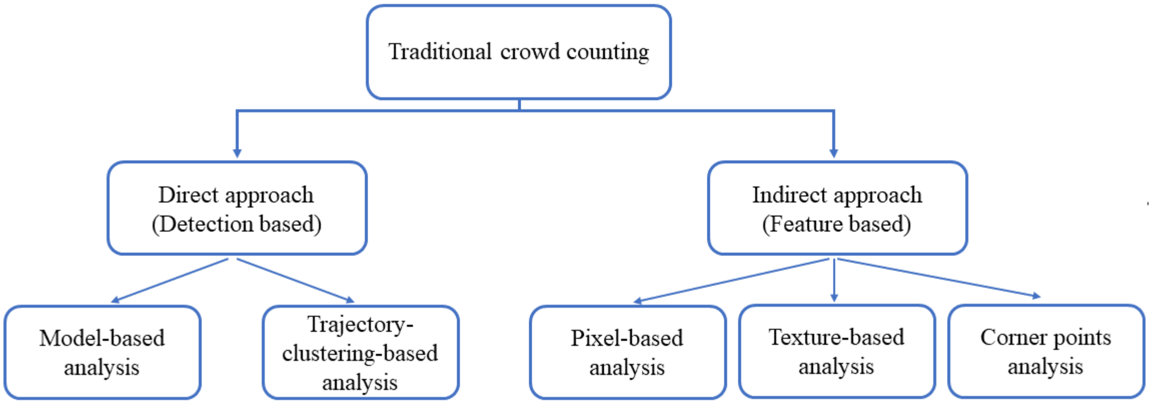

Traditional crowd counting
Direct approach (Detection based)
Indirect approach (Feature based)
Model-based analysis
Trajectory-clustering-based analysis
Pixel-based analysis
Texture-based analysis
Corner points analysis

Figure 2.1: Taxonomy of traditional crowd counting.
Figure from [26]

2.1.1 Direct approach for crowd counting

Direct detection method (also known as detection-based method) has been widely used in many approaches [29, 30, 31, 32]. These approaches localize the position of each object in a single input image, and the number of detections is subsequently used as the crowd count. Traditionally, low-level features including Haar wavelets [30], Histograms of Oriented Gradients (HOG) [29], edgelet [31], and shapelet [33] are used as region descriptors. Then a mainstream classifier such as Support Vector Machine (SVM) [34], boosted trees [35] and random forests [36] is trained for classification. Finally, the number of object instances that the classifier produces on a test image is considered as the crowd count. Although detection-based methods have been effectively employed in low-density crowds, their performance decreases substantially when applied in high-density crowds with small and obscured objects, since they are based on low-level features.

The main advantage of the direct approach is the early performance of object detection in a scene. In this method, perspective, different object densities, and partial occlusions have no impact on the final count when objects are segmented accurately [26]. However, accurate object segmentation is only applicable in low crowd scenes, and this approach frequently yields inconsistent results in crowded and occluded conditions.

Direct methods can be further classified into two groups: model-based and trajectory-clustering-based approaches.

1. Model-based analysis

In the model-based approach, each object is segmented, detected, and then counted using a model or appearance of object shape. The counting of crowds is performed using this method.

Rittscher et al. [37] introduced a method for crowd segmentation on a video series based on the Expectation Maximization (EM) formulation. For all potential individuals, this method includes shape parameters and managed feature allocations. The likelihood function, which is parameterized on the shape and location of objects is used to divide the image features. Finally, another EM formulation is applied in order to estimate the maximum joint likelihood. This system can be used with a numerous camera configuration and has proven to be resistant to partial occlusion, shadows, and clutter. However, this system is expensive, and has limited extensibility [38]. In a probabilistic top-down segmentation method, Leibe et al. [39] proposed an algorithm for pedestrian detection in congested scenes. To determine the probability of a person being present, they integrated local information from sampling appearance features with global features. They employed

Chamfer matching as global features and a scale-invariant modification of the Implicit Shape Model (ISM) for local features in their method. The findings of their experiment demonstrate that, even in challenging crowded scenes with significant overlapping, the method is able to reliably detect and localize pedestrians. In [40], researchers expanded the initial pedestrian detection method proposed by Viola et al. [41] and described a scanning window type detector using spatiotemporal information. Three different types of filters were used in a batch processing analysis of a moderate number of frames to capture moving objects: an appearance Haar-like filter, an absolute difference Haar-like filter, and a shifted difference filter. In their work, eight distinct pedestrian detectors have been developed using the AdaBoost learning method [42] for eight motion directions. A detection strategy for crowd estimates using wavelet templates and vision-based approaches proposed by Lin et al [43]. The feature areas of the head-like contour are first extracted using the Harr wavelet transform (HWT), and then these features are classified using an SVM. The results are categorized into two groups: head presence or head absence. The perspective transforming method is eventually applied to estimate crowd density.

2. Trajectory clustering-based analysis

In trajectory-based technique, crowd counting is carried out after tracking of each independent motion. For this purpose, tracking is performed using clustering interest points on objects that have been tracked over time.

To identify the movements of individuals in crowds, Brostow and Cipolla [44] developed an unsupervised Bayesian clustering method. In order to perform detection for each frame, the relationships between frames are disregarded. Their algorithm is based on the principle that two moving points are probably a component of one entity. In order to depict moving separate entities, their system tracks and probabilistically clusters low level features. The probabilistic criteria for clustering are based on the trajectory coherence of the image space and the space-time proximity. Their approach uses a one-shot data association, which eliminates the need for a training stage for tracking individuals. This strategy is robust and has the ability of locating people in crowded scenes. However, it frequently fails, for example when a person is camouflaged or in the presence of strong arm movements in situations with rigid motion.

Sidla et al. [45] proposed a video-based counting method based on motion detection to find Region of Interest (ROI) and prediction of movement extracting shape information to count individuals in extremely crowded situations. To this end, their algorithm identifies individuals by detecting human head-shoulder regions that have been masked by ROI filter. The motion of the pedestrian is then examined using the Kalman filter and Kanade Lucas Tomasi (KLT) tracking points to produce the co-occurrence matrix feature vector for active shape models. Ultimately, a virtual gateway and a trajectory-based heuristic are applied to count the number of persons. Although their approach is reliable for tracking people, using trajectories to count individuals in a crowd scenario is challenging. Despite the fact that their method is effective for keeping track of people, using trajectories to count people in a crowd situation is difficult.

An object detection method using coherent motion region detection was presented by Cheriyadat et al. [46] for counting and locating people in occluded scenes. By tracking the low-level features of the objects, they consider that a single moving object coincides with a single coherent motion area. Then, a set of point tracks that represent a group of distinct coherent motion areas is provided by the output. Additionally, in order to address the issue of overlapped moves that can result from camera perspective, greedy approach was applied to pick the optimal disjoint set.

2.1.2 Indirect crowd counting approach

Object counting with indirect approach known as feature-based method or regression method. This approach starts with taking the entire crowd as an object, extracting several local, global, and texture features of the foreground image, and then establishing a mapping to the number of dense crowds to estimate the number of crowds indirectly [47]. To this end, a regression function—such as linear [48], Gaussian [49], or neural networks [50]—has been used to quantify crowds using a variety of foreground pixel features, including foreground area [50], texture features [51], histograms of edge orientation [49], and edge count [52]. Although these approaches have achieved significant progress, most of them demonstrate a nearly linear relationship between foreground area and the number of objects in the image. As a result, they are still inadequate for real-world applications, particularly when occlusions and perspective problems are present.

The occlusions issue has been handled by the usage of additional features, such as histograms of edge orientations, and edge count. The effects of the perspective problem have been addressed by a number of techniques, including the geometric factor [48], which assigns weight to pixels based on their location, the Geometric Correction (GC) [53], which equalizes all scales of objects, the perspective map [49], which gives weight to all extracted features, and the Inverse Perspective Mapping (IPM) [54], which calculates the distance between groups of individuals.

Regression-based counting directly maps from the extracted features of images to the number of objects, and because they ignore the object distribution information within the region, these methods do not have the ability to explicitly detect and localize each object [55]. These methods are not capable to capture semantic information since they rely on low-level features. Additionally, these approaches have limitations in complex scenes such incorrect edge-like feature performance, the difficulty of foreground and background subtraction, and lengthy feature extraction.

1. Pixel-based analysis

To estimate the crowd, pixel-based analysis focuses more on local features. Since most pixel-based methods rely on low-level features, these techniques are used to estimate crowd density rather than recognizing specific objects [26]. In pixel-based processing, the initial step is to eliminate the background using a background subtraction technique on a reference image [56] or an automatic background generator that creates an artificial background image [57]. Following that, using the edge detection approach, features are extracted and fed into the back propagation neural network. This method is more effective when there is a very small or low crowd. However, the method may yield inaccurate results in scenarios with dense crowds due to severe occlusion [58].

Ma et al. [59] introduced a technique for estimating crowd density based on pixel counting. They established a mathematical relationship for geometric correction and demonstrated that it can be done directly on pixels in the foreground, independent of where they are in relation to one another in the image. To this aim, they created a density map that assigned each pixel a weight based on the area it covered on the ground plane. The sum of the weighted foreground pixels is then used to determine crowd counts.

Yang et al. [60] adopted a sensor network from a side and top perspective to segment and estimate the number of individuals in crowded scenes. To produce a top view of visual hull, numerous cameras were used in this technology in order to project the 3D silhouette cones. Dimensions eventually intersected to create two-dimensional information. Then, using the extracted silhouettes, a geometric method is used to calculate bounds on the number and location of the individuals. They also employed thresholding background subtraction to make silhouettes overestimates in order to address the issue of the sensitivity of silhouette intersection to noise. This technique has the disadvantage that some objects could not be seen from all angles in crowded situations, making it difficult to localize specific objects.

2. Texture-based analysis

Texture is one of the attributes of an image that gives detailed information about each region of the image. Texture feature extraction has attracted considerable interest from several studies and is used as a highly effective strategy in many image processing applications. Due to its examination of a coarser grain and requirement for image patch analysis, texture extraction is more reliable than pixel-based approaches [26]. This method is mainly used to estimate the number of objects in a scene rather than counting them.

According to a theory by Marana et al. [61, 62, 63], images of dense crowds typically show fine textures, while those of low-density crowds typically show coarse textures. They employed a Grey Level Dependence Matrices (GLDM) [64] as a statistical technique to extract features related to crowd density from images. With the aim of crowd estimation, four GLCM features were extracted: contrast, homogeneity, entropy, and energy. Then, in order to classify crowds, these features were used as neural network input. They developed five density classifications (very low, low, moderate, high, and very high) using a Kohonens Self-Organizing Mapping (SOM) neural network [65]. However, this method takes more time to classify the crowds.

Multi-scale analysis and SVM were used in the texture extraction approach proposed by Xiaohua et al. [66]. They employed a 2D Discrete Wavelet Transform (DWT) to transform the crowd image into multi-scale formats, which they then map to a multi-dimensional feature space. The crowd density was then classified into low, moderate-low, moderate-high, and high-density levels using a tree-structure SVM-based classifier. In terms of computational complexity, this hybrid feature extraction method outperforms Marana et al., [61]. In addition, their study successfully classified crowds with a modest density at 95% accuracy.

Wu et al. [67] introduced a learning-based method with the ability of assessing textures locally and globally and subsequently identify abnormal crowd density. To improve density estimation in the crowded conditions, their technique generates a series of multi-resolution picture cells using a perspective projection model.

Following that, texture feature-based density vectors are extracted for each input image cell by performing GLDM [61]. An SVM training system uses these vectors, which have been scaled, to connect the 15 textural features to the actual density. Finally, SVM categorizes the density estimation distribution and assigns it to one of two categories: normal or abnormal. Even though the proposed system proved to be effective on real crowd videos, this approach has the problem that each time the initial setup of the system is altered, a new training procedure is required. Zhang and Li [68] proposed Accumulating Mosaic Image Difference (AMID), an improved foreground detection method, to capture complicated random motion patterns. They suggested the idea of intra-crowd motions since random, minute movements are an essential component and one of the fundamental characteristics of high-density crowds. Then, in order to accurately estimate crowd density, they used the AMID feature to adequately describe these local intra-crowd motion patterns. They used a perspective distortion correction model to apply a normalizing method to the acquired foreground in their study. This model was used to determine the crowd density for the measured areas.

3. Feature points analysis

Albiol et al. [69] proposed an alternative indirect strategy rather than segmenting or making an effort to identify persons in each frame. To this aim, they extracted moving corner points as features by Harris algorithm [70] to estimate the number of moving objects. Given that this strategy produced the best results, it has been widely used and developed by numerous researchers in local or global level [26].

Dittrich et al. [71] described a method for crowd counting by merging data from numerous cameras. By using the concept of multiple views and integrating information, they could overcome the occlusion issue and improve the reliability of the counting result. Therefore, to determine the motion vector of the objects in the scene, their system rectified corner points on the ground plane that are associated with them. Then, in order to estimate the counts of the objects in the scene, weights are applied based on distance, and the mean number of points per objects is obtained.

To estimate crowd density, Conte et al. [72] improved the approach of Albiol, et. al. [69] and proposed a robust real-time method to detect corner points, and the number of moving corner points in a scene by considering the varied characteristics of different population densities. To correct the perspective distortion, their strategy entails initially partitioning the entire scene into smaller horizontal zones. In accordance with their distance from the camera, each zone has a unique size. Then, the results of the counting are calculated for each zone individually. In their work, they evaluated the window search, the three-step search, and the local-difference method as three methodologies for classifying points. Window search and three step search use motion estimations as their basis, whereas the local-difference approach concentrates on variations in color intensity. Despite the fact that the three techniques have about identical accuracy for crowd count estimation, their experimental results demonstrate that the local-difference classification algorithm is easier and less computationally intensive.

Adaptive Neuro-Fuzzy Inference Systems (ANFIS) and ϵ-SVR regressor are two trainable estimators that were compared in the study by Acampora et al. [73] as the indirect crowd counting approach. The method first applies a feature detector to identify the interest points, and then it uses an estimated motion vector to filter out the static points. The authors concluded that the neuro-fuzzy based estimator performs more effectively in scenarios with high crowd densities, while the ϵ-SVR based estimator performs better in scenes with low crowd densities.

On the basis of interest point measurements and single feature regression, Fradi and Dugelay [74] developed an indirect crowd counting approach. To improve accuracy, this preliminary study employs perspective normalization at the pixel level rather than allocating one distance value to each set of unique features. Further, Scale-invariant descriptor (SIFT) [75] performed to identify the locations of interest points by measuring the maxima and minima of the difference between Gaussians in scale space. It has been demonstrated that SIFT is resistant to affine, rotational, and scale transformations. Additionally, a density-based clustering algorithm is employed to derive the shape of a group of points by using the shape technique. This method achieved a high accuracy rate, and experiments have demonstrated its capacity to preserve a linear relationship between the proposed feature and the estimated count even in the face of severe occlusion scenes and perspective distortions.

Liang et al. [76] introduced a feature point-based method for tracking and counting crowd flows. A three-frame difference method was used to enhance the SURF point detection procedure. In order to identify the SURF feature points that genuinely belong to the moving crowd, which minimizes time complexity, the binary image of the moving foreground is employed as a mask image. Then, for further improvement, they modified the Density Based Spatial Clustering of Application with Noise (DBSCAN) clustering algorithm [77] to cluster just the motion feature points. Finally, a Support Vector Regression (SVR) machine is employed in conjunction with a Lucas Kanade local optical flow algorithm [78] with Hessian matrix approach to predict the movement orientation and count the crowds in flow.

2.2 Deep learning crowd counting approaches

Deep learning models have outperformed traditional machine learning approaches in recent years in the field of crowd counting. To learn and classify crowd regions of an image, deep learning algorithms rely on deep neural networks to extract semantic invariant features. Therefore, current research has shifted its focus to developing CNN-based techniques, as CNNs provide a more robust feature representation, compared to the hand-crafted features utilized in traditional approaches. The success of CNN in crowd counting and crowd density estimations is also mainly due to its capacity to learn nonlinear relationships between images and the number of objects in images or their associated density maps.

For crowd counting in highly dense crowds, Wang et al. [79] developed the first end-to-end deep CNN regression model. Following CNNs success in image classification, they proposed a CNN model to count objects in the region of interest (ROI) in the image. In their architecture, they modified the original AlexNet network [80] by replacing the final fully connected layer with a single neuron to obtain an object count. Furthermore, training data augmented with additional negative samples were used to eliminate false responses in the backdrop of the images. However, as the crowd grows denser, the occlusion inside the crowd increase, which reduces the ability of basic

CNNs to extract features in these conditions. Therefore, in order to address this issue, Shang et al. [14] developed a CNN architecture in which the final crowd count outputs the entire image rather than partitioning the image into patches. As a result, due to the shared computations on overlapping regions achieved by integrating multiple stages of processing, the complexity is reduced. Although developed CNNs have overcome the occlusion problem, most CNN-based approaches have many parameters and demand a lot of computing resources, which restricts their practical uses. In embedded systems with little memory, this aspect can be very problematic. Therefore, a variety of approaches have been developed to address this issue. For instance, Ding et al. [81], proposed a deep recursive network structure using Recursive Convolutional Network [82]. When the effect is equal, their network structure dictates that the network parameters are less, making them more suitable for use in real-time settings.

Deep learning approaches, in general, require a significant amount of data to be trained properly; otherwise, these techniques will fail to yield high accuracy. Nonetheless, labeled data might be challenging to collect in many situations. Therefore, to address this issue, researchers use an auxiliary task that sorts the unlabeled data in order to enhance the network performance when the quantity of labeled data is sparse.

The CNN-based crowd counting methods are split into three categories based by the type of the training dataset and output of the network: 1) Detection-based CNN approach, 2) Regression-based CNN approach, 3) Fusion of detection and regression-based CNN approaches (see figure 2.2).

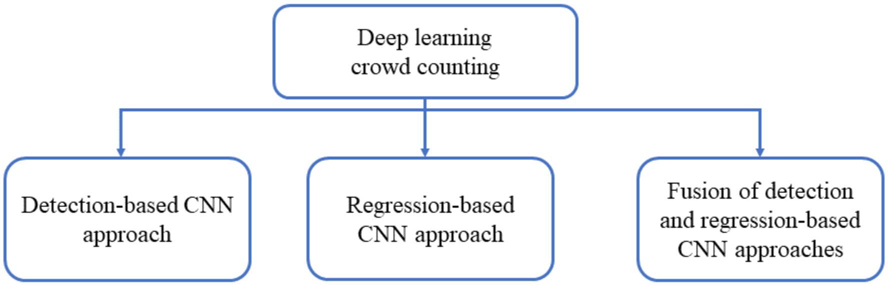

Deep learning crowd counting
Detection-based CNN approach
Regression-based CNN approach
Fusion of detection and regression-based CNN approaches

Figure 2.2: Taxonomy of deep learning crowd counting.
Figure from [26]

2.2.1 Detection-based CNN approach

The detection-based approach initially trains the bounding box-annotated image dataset. Then, the network accurately detects every object in the image throughout the test phase. This approach is capable of precisely locating and counting all of the objects in an image.

An end-to-end training network architecture was proposed by Stewart et al. [83] for the case where the detection object almost completely lacks overlap. They used GooLeNet, which provides substantial location information, to convert each image into 1024-dimensional high-level feature representation. Then, using an Long Short-Term Memory (LSTM) [84] network as a controller, a series of predicted bounding boxes are generated in descending confidence from the representation of these features. The LSTM stops if it is unable to confidently locate any detection box larger than the predetermined threshold. Non-Maximum Suppression (NMS) was not used, because it only processes the bounded box and ignores the image data. They stated that by employing recurrent neural networks to prevent making numerous predictions of the same target, this system produced promising results.

To capture context information and prevent missed detection, Li et al. [85] introduced the HeadNet adaptive relational network. In their research, the features of the input image were extracted using Resnet-101 as the feature extraction network. In order to learn individual stability, local structured feature modules were used, while global adaptive modules were employed to encode the pre-quantified intergroup conflict. The bounding box and confidence are then generated by the network.

Detection-based CNN methods take advantage of the appearance features from still images or motion vectors from videos [86]. Depending on whether the Region Proposal Network (RPN) is used, the two most prevalent types of image object detectors based on deep neural networks described in the literature are: 1) one-stage detector, and 2) two-stage detector.

The architecture of both one-stage and two-stage detectors including three distinct components [87]: a backbone, a neck, and a head. The backbone is responsible for extracting representative features that frequently use CNN-based networks or more recent ones such as ViT-FRCNN [88], ViT-YOLO [89], and Swin transformer [90] that include transformer-based networks and self-attention mechanisms to achieve high performance [91]. The neck, which is almost an essential component, is in charge of extracting multiscale features with the aim of improving the representation. Finally, based on the sort of task, the heads can be classified as dense or sparse. Dense heads in one-stage detectors serve primarily as predictors, whereas they serve as proposal box generators in two-stage detectors. The sparse heads are exclusively employed in two-stage detectors to predict bounding boxes.

1. Two-stage detector

The first type of detectors are two-stage detectors, such as Region-based Convolutional Neural Network (R-CNN), a ground-breaking achievement in the field of object detection proposed by Ross Girshick et al [92]. Two-stage detectors have a longer inference time because of the increased number of regions and additional stages [91]. Basically, the RoI pooling layer acts as a separator between the two stages of this type of object detectors. In the first stage, the RPN filters out the anchor boxes to generate Region of Interests (ROIs) that propose possible object bounding boxes. In the second stage, the RoIPooling (RoIPool) is used to extract features from each candidate box in preparation for the classification and bounding-box regression tasks that follow [93] [87]. The basic architecture of two-stage detectors is shown in figure 2.3.

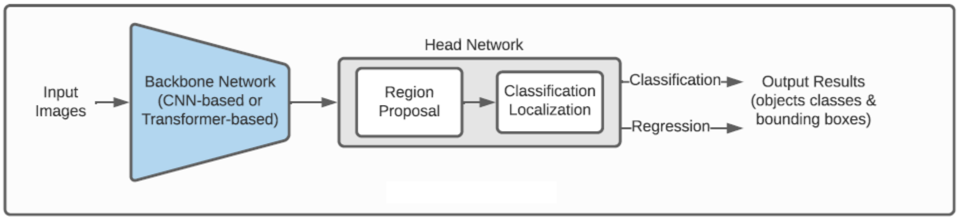

Input Images
Backbone Network (CNN-based or Transformer-based)
Head Network
Region Proposal
Classification Localization
Classification
Regression
Output Results (objects classes & bounding boxes)

Figure 2.3: Two-stage detector.
Figure from [91]

R-CNN [92], SPP-net [94], Fast R-CNN [95], Faster R-CNN [96], R-FCN [97], and Mask-RCNN [11] are some of the most important examples of two-stage object detectors employing region proposal framework.

As previously stated, R-CNN was introduced as a pioneering region-based CNN detector to address the problem of object detection. Apart from taking advantage of the substantial capability of CNN architecture, R-CNN has the drawback of using fixed input image size. Following that, established methods in general object detection were developed, which were accompanied by major advances in deep learning methodologies and gradual improvements in processing capacity. Therefore, in order to overcome the aforementioned issue of R-CNN, SPP-Net as a revolutionary CNN architecture based on the theory of spatial pyramid matching, proposed by Zhang et al. [94]. Base on this theory, SPP-Net adds a spatial pyramid pooling layer between convolutional layers and fully connected layers. Although this approach can overcome the limitations of a fixed-size network, it only modifies the weights of the fully connected layers throughout the process of fine-tuning [98]. Following the limitations of the SPP-net and R-CNN, Ross Girshick et al. [95] proposed Fast R-CNN as an enhanced model that employs a RoI pooling layer to use a single fixed size feature map for all the different-sized region proposals [93]. Later, Ren et al. [96] developed Faster R-CNN to improve Fast R-CNN and R-CNN, both of which were based on traditional Selective Search Algorithm (SRA). Significantly, they used RPN to generate the proposals in order to ameliorate the slow speed and time of the two preceding networks. RFCN [97], enhanced the accuracy even further through the employing of fully convolutional neural networks. Faster RCNN was then expanded to create Mask R-CNN [11], which is mostly an instance segmentation algorithm and a more accurate object detector in terms of detection.

2. One-stage detector

The second type of detectors are one-stage detector, which use a single DNN to perform both localization and classification at the same time, and can predict bounding boxes and their corresponding class labels straight from input images [99, 100]. Specifically, the one-stage detector eliminates the requirement to create candidate boxes and instead relies on the network output object classes and bounding boxes, which are both completed at the same time [98]. Figure 2.4 depicts the basic construction of one-stage detectors.

The initial YOLO (You Only Look Once) proposed by Redmon et al. [101] and its derivatives are the most extensively used one-stage object detectors, followed by Single Shot MultiBox Detector (SSD) [102], Deconvolutional Single Shot Detector (DSSD) [103], and RetinaNet [104].

Historically, OverFeat was first proposed by Sermanet et al. [98] as a pioneered single-object

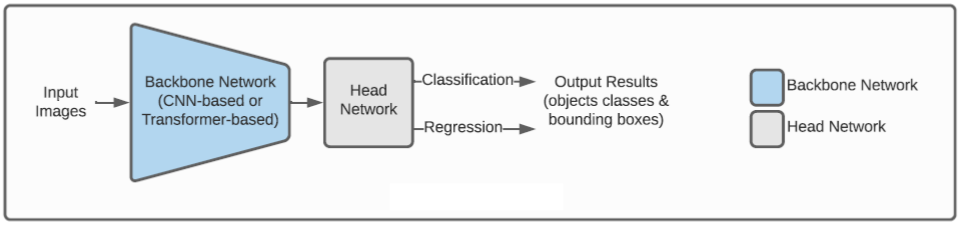


Figure 2.4: One-stage detector.
Figure from [91]

localization approach built on a deep ConvNets-based integrated framework for classification, localization, and detection. Later, YOLOv1 was a one-stage object detector based on a fast feedforward convolutional network proposed by Redmon et al. [101] to predict object classes and locations directly. Although YOLOv1 stands out for its fast-detecting speed, its accuracy is comparatively low and the localization is inaccurate [105]. Consequently, the first version was improved in various areas, and YOLOv2 [106], also known as darknet-19 was proposed to replace it. The accuracy and object localization issue of YOLOv2 improved significantly by adding batch normalization on all of its 19 convolution layers, employing a high-resolution classifier and using convolution layers with anchor boxes to predict bounding boxes instead of the fully connected layers [107]. YOLOv3 [108] is the next sophisticated variant of YOLO that aims to improve on the accuracy of YOLOv2 and it takes advantage of the concept of a feature pyramid to performs detections for large, medium, and small objects at three scales (levels) [105]. After that, Bochkovskiy et al. [109] introduced the 4th version of YOLO, which resulted in significant improvements in overall speed and accuracy. YOLOv4 employs the PANet (Path Aggregation Network) to train and extract image features, which is potentially faster and more accurate than YOLOv3 that uses FPN [110]. Following that, the fifth generation of YOLO released by Jocher [111] which is more flexible and smaller than YOLOv4. However, YOLOv4 and v5 use architectures similar to YOLO v3 and alter modern approaches to make the network more efficient and ideal for single GPU training [112].

In addition to the YOLO series, a Single Shot MultiBox Detector (SSD) is also part of the single-stage detection framework proposed by Liu et al. [102], which uses a single deep neural network to detect objects in images. SSD as a feed-forward convolutional network obtains the feature map and multi-object categories using VGG16 as the backbone network to detect objects efficiently and accurately [113]. The SSD-based method has the ability of extracting multi-level features from images in an automated and efficient manner [114]. However, the major drawback of SSD is that the context information in the low-level feature maps is insufficient, making this method ineffective in detecting small objects [114, 115]. Deconvolutional Single Shot Detector (DSSD) is a modified version of SSD with the addition of the prediction and deconvolution modules, as well as the replacement of the backbone network VGG16 with a more powerful one (i.e., ResNet-101) and an hourglass network structure [103]. Lin et al. [104] proposed RetinaNet that inherits the fast speed of previous detectors while also adding the ability to decrease the impact of extreme class imbalance by employing focal loss as a classification loss function. Despite improvements, the accuracy of one-stage detectors is still inferior to that of two-stage detectors.

2.2.2 Regression-based CNN approach

Regression based methods are trained using a point-annotated image dataset or unsupervised techniques. In the test phase, the network either generates a density map or outputs the total number of objects in the image. Regression-based counting approaches are categorized into two groups:

1. Regression counting methods, where the number of objects is output directly from the image.
2. Regression density map methods, where counting and estimating the density of the crowds is performed using a density map.

In situations with higher densities, regression-based CNN approaches based on generating density maps can provide better prediction outcomes than detection-based CNN methods. Regression-based approaches (see figure 2.5) are further described in this section, according to the network characteristics.

1. Multi-scale model

Due to the increasing complexity of counting datasets, many models may underperform when dealing with significant changes in a high crowd density. As a result, an increasing number of sophisticated scale-resistant models were developed with the primary goal of extracting the information of various scales in the image using a multi-column structure, feature pyramid network,

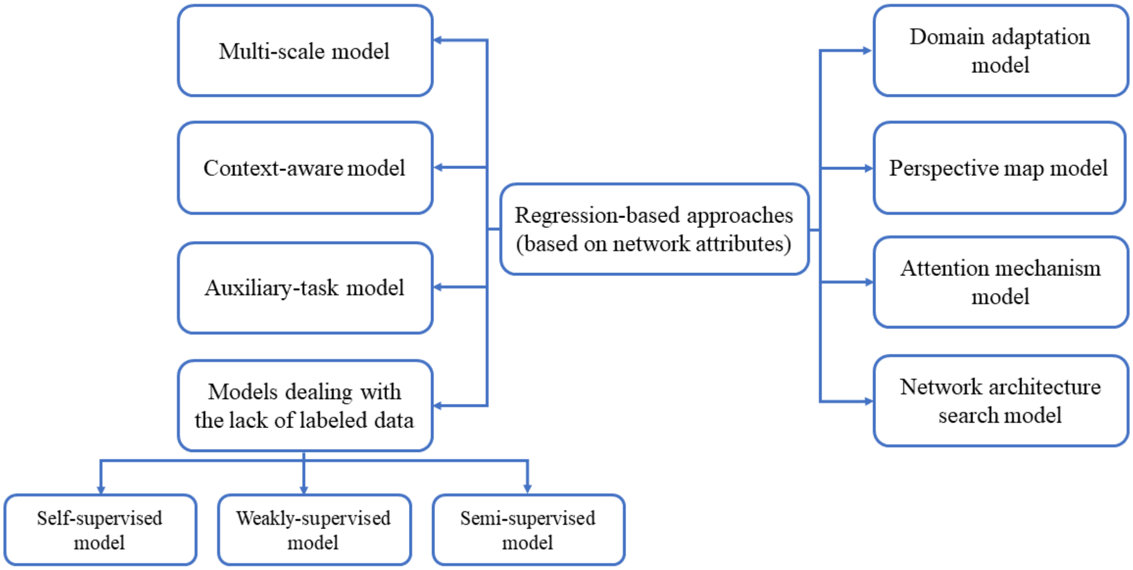


Figure 2.5: Regression-based crowd counting approaches according to network attributes.

or scaling to increase the robustness of scale aware [116]. To capture both high-level semantic information and low-level features, Boominathan et al. [117] combined deep and shallow fully convolutional networks to address crowd scales as well as perspective variations. The deep network, which uses the VGG network, is responsible for acquiring high dimensional information, whereas the shallow network is used to catch smaller objects that are far from the camera. This approach expands on the representational power of the VGG network after the filters for crowd counting are appropriately coordinated. The estimation of each pixel's density is distinct from image classification, though, since the problem of image classification involves assigning a discrete category label to each image. They obtained the density estimation at the pixel level and converted this network to convolutional by removing the fully connected layers from the VGG layout. On the other hand, the shallow network has three convolution layers and a convolution kernel size of 5×5. Then, the shallow and deep feature maps were combined. The density map is ultimately produced by sampling the feature map to the size of the original image.

To capture multi-scale object properties, Zhang et al. [118] presented a multi-column CNN (MCNN). In this structure, three networks are utilized to separately extract various crowd image features, and a convolution layer then combines the features of the three scales. Since different branches have different receptive fields, various sizes of objects can be identified through this framework. However, the inability of this system to predict real-time crowd counting, is one of the limitations of this model.

By utilizing the concept of three subnetworks, Sam et al. [15] proposed a crowd counting model based on switching CNNs. Their architecture consists of multiple MCNN regressors and a switch classifier trained to select the optimal regressor. The input image is first divided into patches, and each patch is then pre-trained into the corresponding subnetwork in each column. By minimizing the training error in that column, the patches are split into three categories. A Switch-CNN is then trained to classify each patch into the appropriate subnetwork using these image categories. The accurate crowd estimation of the original image is finally comprised by the precise prediction of each patch.

Onoro and Sastre [14] developed a scale invariant CNN model (HydraCNN). HydraCNN is capable of estimating the density of objects in different crowded circumstances without explicit scene information and perspective. In their work, the authors firstly introduced the Count CNN (CCNN) architecture that incorporates the perspective information for geometric correction of the input features. Then, they constructed a network of three heads and a body. Each head is a CCNN architecture for learning features of a particular scale. Finally, the outputs of all the heads are concatenated and fed to the body to estimate the final density map.

2. Context-aware model

Context-aware models integrate local and global contextual information into the CNN architecture to enhance the accuracy of detection. Incorporating local and global information for crowd counting is a complex task that has attracted several researchers.

Sheng et al. [119] proposed to integrate semantic information by learning locality aware feature (LAF) sets to perform accurate crowd counting. The proposed architecture comprises three main components. First, a CNN transforms the input raw pixel data into a dense attribute feature map, where each dimension of a pixel feature corresponds to the probability strength of a semantic class. Then, following the idea of spatial pyramids on neighboring patches, the LAF is introduced to explore more spatial context and local information. Finally, the local descriptors from adjacent cells are encoded into image representations using the VLAD encoding method. Sam et al. [120] implemented the idea of a top-down feedback network for crowd counting. Two CNN networks are used simultaneously in this model. The first network uses a bottom-up approach with two columns and various receptive fields to predict the crowd density map. The second network performs a top-down approach to discover how to link low-dimensional CNN regression features with high-dimensional context information.

The current approaches only address changes in crowd composition; they do not address fluctuations in crowd size. To address size and rotation problem for the crowd counting task, Liu et al. [121] suggested a Deep Recurrent Spatial-Aware Network (DRSAN). The learnable spatial change module and the regional optimization approach are initially used to address the issue of rotation variations. Global Feature Extraction (GFE) and Recurrent Spatial-Aware Refinement (RSAR) are the two components of the network. The global features of the input image are extracted using the GFE module. To create high-quality density maps, RSAR is applied. The RSAR module consists of two parts that are performed alternately: 1) Spatial Transformer Network (STN), which locates the regions of interest in the crowd density map, and 2) local refinement network, which uses residual learning to optimize the density map.

3. Auxiliary-task model

One or more crowd counting tasks are provided to the network in this model as auxiliary tasks that can train simultaneously [116]. Marsden [122] employed ResnetCrowd, a Resnet-18 based architecture [123], as an auxiliary task to simultaneously perform crowd counting, violence detection, and crowd density level classification. Before ResnetCrowd, there was no method that could perform all three of these tasks at the same time. The main module of the network is consisting of the first five convolutional layers from Resnet18, the interleaved batch normalization and skip connections. After the first convolutional layer of Resnet18, the authors eliminated the maximum pooling layer but kept the larger feature map for crowd counting at pixel-level. They generated the crowd density map after initially counting pixel-level objects using the counting heatmap convolution layer. The feature maps produced by the feature extraction network were combined and added to the full connection layer of the various task.

Zhao et al. [124] used a two-phase training approach to divide the crowd counting problem into two sub-tasks: crowd density map prediction and crowd speed map. A CNN was applied to learn features from the video stream using line-of-interest (LOI) at the exit and entry. In order to estimate crowd density, Zhao et al. [125] established three heterogeneous attributes: geometric, semantic, and quantitative. These attributes were then used as multiple auxiliary tasks to produce more robust features and enhance the performance of the crowd counting task. Liu et al. [126] introduced the Recurrent Attentive Zooming Network (RAZNet) to simultaneously count and locate crowds. The RAZNet features a dedicated branch for region proposals that can be used to locate dense areas, recursively detect the region of the blurred image, and repeatedly zoom into proposed regions.

4. Models dealing with the lack of labeled data

There are various situations where there is relatively little labelled data when performing crowd counts. Models that address the lack of labelled data problem are described in this section. There are three types of models in this category: semi-supervised, weakly-supervised, and self-supervised.

1. Semi-supervised model

   The semi-supervised Generative Adversarial Networks (GANs) were expanded from the classification problem to regression for dense crowd counting by Olmschenk et al. [127] in their suggested model for crowd counting. In order to determine whether the input sample is real or fake, they adopted a semi-supervised dual objective GAN structure that demands the discriminator to produce two distinct outputs: the expected regression value and a tag. When supervised regression and unsupervised classification are combined, the discriminator is compelled to learn more reliable aspects of the crowd image. Consequently, this approach performs superbly even with limited labelled data.

2. Weakly-supervised model

   Sam et al. [128] proposed an almost unsupervised method for crowd counting that makes use of sparse characteristics. Most of the parameters in their work are trained using a Grid Winner-Take-All (GWTA) autoencoder to extract features from unlabeled images, and only a little portion of the parameters (0.1%) are fine-tuned using a supervision method on location-level annotated data. Lei et al. [129] trained the crowd-counting network using multiple auxiliary tasks. The authors also added a stronger regularization to predict the density of images that were annotated weakly.

3. Self-supervised model

   There are issues with overfitting in many crowd counting algorithms since the size of the available crowd counting dataset is so small. To overcome the aforementioned issue, Liu et al. [130] presented an approach that leverages more unlabeled data during training. To this end, they suggested two methods of gathering datasets: 1) keyword query, which searches images with their corresponding keywords, and 2) query by-example image retrieval, which generates ranked datasets from the existing dataset and filters out the irrelevant images. The generated datasets were then combined with ground-truth labeled crowd scenes through the use of three embedding approaches: ranking plus fine-tuning, alternating-task training, and multi-task training. Three different training strategies were tested in this study, with alternative-task training producing the lowest mean square error and multi-task training producing the lowest average absolute error.

5. Domain adaptation model

Domain adaptation methods can be used to count crowds in any object domain. Crowd counting in high density situations, for instance, is one of these domains due to the varied environments present in real scenes. Wang et al. [131] observed that the data collector in addition to its particular methodology can be used to address the problem of crowd estimation in high density scenarios. Data collector was developed first in order to generate synthesized data. The generated data was then labeled automatically using a data labeler. Along with the data collector, the authors employed two schemes to improve counting performance. First, a large-scale synthetic crowd counting dataset is used to train the network, and the labelled dataset is subsequently used to fine-tune it. Second, a domain adaptation framework is performed for crowd counting without using a lot of annotated data.

6. Perspective map model

Every position in the image has a perspective map that shows the perspective distortion. These methods enable the network to produce density maps by incorporating perspective maps.

In order to enable the network to execute perspective normalization and enhance its robustness to changes in scene scale and perspective, Zhang et al. [132] presented a method that generates density maps based on perspective information. A Perspective-Aware CNN (PACNN) model was suggested by Shi et al. [133] that employed perspective maps to predict density maps. Perspective maps provide scale change information within an image which is crucial for the detection of smaller objects. They initially produced a ground-truth perspective map, which they used to develop perspective-aware weighting layers to adaptively integrate the multi-scale information to predict the density map. To address the issue of scale variation in scenes caused by perspective effects, Yan et al. [134] proposed a Perspective-Guided Convolution Neural Network (PGCNet). The smooth spatial variation of feature maps is guided by the perspective information using PGCNet. PGCNet introduces a perspective estimate branch that can be trained in either supervised or weakly supervised settings.

7. Attention mechanism model

Attention models enhance the network with attention mechanisms to increase the accuracy of crowd counts. In order to select the most pertinent piece of data for visual analysis, these models train an intermediate attention map.

In addition to improving the performance of crowd counting, Sindagi and Patel [135] developed the Hierarchical Attention-based Crowd Counting Network (HA-CCN), which uses attention processes at multiple levels to enhance important network features that have significant effect on the crowd counting task. They initially pretrained CNN-based networks using Global Attention Models

(GAM), which concentrate more on high-level information and selectively boost essential features, and Spatial Attention Models (SAM), which increase low-level characteristics in the network. The density map was then estimated using an image-level labelled dataset that was classified based on degree of crowdedness.

Liu et al. [136] proposed an Attention-injective Deformable Convolutional Network (ADCrowdNet). To address the issue of diminishing the counting accuracy in noisy and high-density situations, they incorporated an attention mechanism and multi-scale deformable convolution to ADCrowdNet. The Attention Map Generator (AMG) and the Density Map Estimator (DME) are two connection networks that are part of the ADCrowdNet framework. In a crowded situation, AMG first determines the parts of the crowd area and then determines the corresponding degree of crowding. The DME creates a high mass density diagram using the crowd area that the AMG has detected.

8. Network architecture search model

Currently, the hand-crafted density estimation network forms the basis of the majority of crowd counting and crowd density estimate approaches using CNN. Therefore, the multi-scale properties in this category are typically employed to handle scale changes and extract the multi-scale features of the fundamental network.

Hu et al. [137] developed an end-to-end automatic search Multi-Scale Network (MAS). In order to automatically construct a crowd counting model, the authors also employed Neural Architecture Search (NAS). A multi-scale encoder-decoder network is effectively searched by NAS-Count. Each unit that makes up the encoder-decoder network has the ability to automatically extract and combine multi-scale features. On four difficult datasets, NAS-Count outperforms time-consuming manual design tasks by automatically developing the multi-scale model in less than one GPU.

2.2.3 Fusion of detection and regression-based CNN approaches

Detection-based methods outperform regression-based methods when there is little to no crowd density. At the same time, regression-based methods outperform detection-based methods when there is a high density of crowds. Therefore, finding a middle path between count by detection and regression is the best method to strike a balance and adopt a model that can perform well in both situations.

Liu et al. [138] proposed the DecideNet model to adaptively leverage detection and regression-based count estimations in order to take advantage of these two methods. DecideNet has the ability to adaptively modify the weights of the detection and regression techniques in response to variations in crowd density. There are three modules in DecideNet: 1) RegNet, which calculates the network density map using the regression-based approach; 2) DetNet, which precisely locates each object in the scene using the detection-based method; and 3) QualityNet, which gives weights to two networks.

Sam et al. [86] introduced the LSC-CNN (Locate, Size and Count) dense detection architecture for counting crowds, which only uses the dataset of point annotation. LSC-CNN has the ability of locating the size and position of each object in the scene. In order to improve object recognition accuracy in the scene and produce correct predictions at various resolutions, LSC-CNN utilizes a multi-column architecture with top-down feature modulation. LSC-CNN has three main modules including feature extraction, Top-down Feature Modulator (TFM), and Grid Winner-Take-All (GWTA). Feature extraction module sends the features retrieved from images of various resolutions into the TFM module. The TFM module combines multi-scale feature graphs, predicts bounding boxes, and uses NMS to choose the most useful detection among several resolutions. The GWTA module is used to handle data imbalance during training phase. The authors concluded that their model outperforms at crowd counting than the current regression method. The experimental results also showed that their framework performs better at placement and has all the benefits of the detection system.

2.3 Discussion

In our application context, the studied plantation blocks may have diverse characteristics and include several objects, such as trees, debris, and water accumulation. These objects can be considered as indicators for planting block conditions, which are generally related to the final mound density. Therefore, we aim to employ object detection algorithms to detect multiple objects in each patch of the planting block, which would be useful to indicate the properties of each block region represented by an image patch. Object detection techniques combine the functions of object localization with image classification. Using object detection, we take an image as input and produce one or more bounding boxes with associated class labels. These methods can handle multi-class localization and classification and objects with multiple occurrences. Even though object detection techniques have been proven to be effective for detecting objects in crowded scenes [91], bounding box detection is generally insufficient when detailed information about object boundaries is needed. In addition, more information on object regions would improve our application's precision in quantifying their presence.

Further development of object detection is image segmentation, which marks the existence of an object using pixel-wise masks created for each object in the image. This approach provides greater granularity than bounding box generation. Instead of drawing bounding boxes, segmentation identifies the pixels of an object. This granularity enables us to obtain detailed information about the objects included in the image at pixel level. In our approach, this information would be useful for calculating the presence ratio of each object type. Thus, we believe that using instance segmentation would be an appropriate choice in our application context.

In this direction, we formulate our first prediction stage as an object detection task by employing instance segmentation prior to counting. More specifically, this is done by applying a two-stage detector. As previously mentioned in the section on detection-based CNN technique (section 2.2.1), one-stage detectors predict the classes and locations of objects by concurrently performing object localization and classification. Therefore, the detection process is fast but less precise. On the other hand, two-stage detectors first propose Regions of Interest (ROIs), and then they employ classification to select the classes of detected objects [99]. As a result, two-stage detectors provide more accurate localization and classification in comparison to one-stage detectors, that give faster real-time detection [93] [99].

Among the different two-stage approaches, Mask-RCNN has the ability of detecting multiple objects, as it adds a branch to predict segmentation masks at the pixel level parallelly to the existing branches in Faster R-CNN for classification and bounding box regression [139]. The mask branch provides extra information for object detection while just slightly increasing the computational cost in conjunction with other tasks.

Following instance segmentation, we perform a patch-level global prediction, which is inspired by the indirect counting approach. More specifically, the second stage of our method will be based on regression analysis as a predictive technique. This analysis uses the information extracted from the previous step by Mask-RCNN to correct the number of mounds that have been detected, and then provide a precise final count. In our work, there is a considerable appearance variability at the scene level, where objects of interest could be unseen due to several perturbation factors, including occlusion by woody debris, water accumulation, mound erosion, and destruction. Thus, applying both models simultaneously would be more efficient than merely employing a visual object detector.

Chapter 3

Methodology

3.1 Motivations

We aim to develop an automated framework to precisely estimate the number of mounds on each planting block represented by an orthomosaic. For each planting block, we capture a batch of images using UAV, to produce a high resolution orthomosaic. We divided each orthomosaic into fixed cell sizes due to the high resolution of images, and used them as the input for our framework, as shown in figure 3.1.

Visual inspection of different blocks (see figure 3.2) shows that the number of mounds is varying from one patch to another. In fact, this variation is due to many factors, such as mechanical site specificities, environmental factors (dry, wet, and snow), and presence of other objects, such as debris and trees (e.g., mound occlusion by debris, and appearance change due to tree shadows).

To handle such difficult factors, a sequential two-step paradigm is adopted in our system design. Firstly, we propose to detect and segment objects using instance segmentation, in order to localize mounds and quantify the presence of other relevant objects in patches. This step allows us to determine the properties of each patch, which may vary among regions within the same block. Secondly, we employ patch-level correction to obtain a final number of mounds. System training is performed according to these two steps, as illustrated in figure 3.3.

Once the entire system is trained, the two models are used to perform mound counting on a new orthomosaic. That is, local image segmentation is applied as a preliminary stage to segment

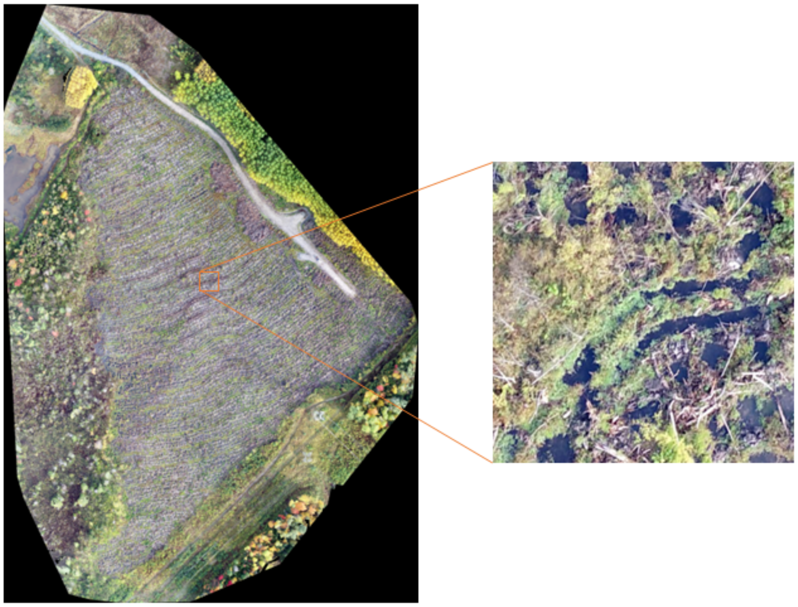
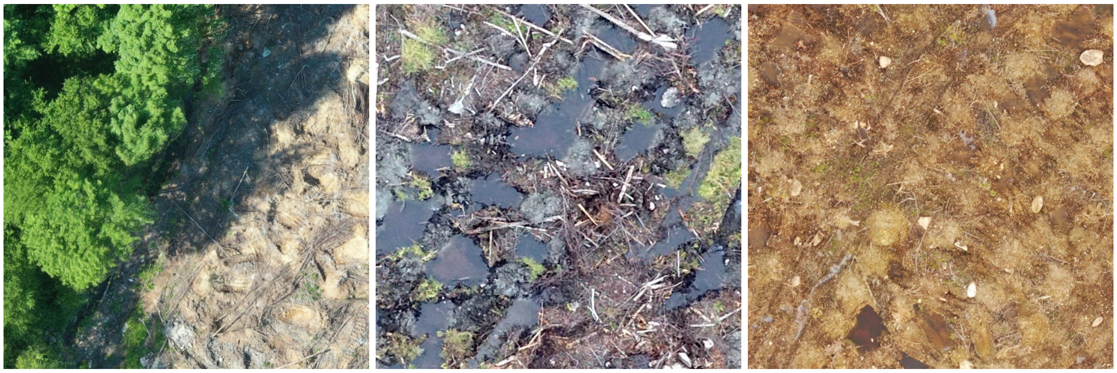


Figure 3.1: Orthomosaic of one planting block captured and reconstructed (left) along with the example of one extracted patch (right).



(a) (b) (c)

Figure 3.2: Examples of challenges for three patches from different orthomosaics.
(a) Presence of tree shadow causing partial occlusion of mounds. (b) Water accumulation due to heavy rain. (c) Mounds with similar texture to the surrounding areas (background) in dry terrain.

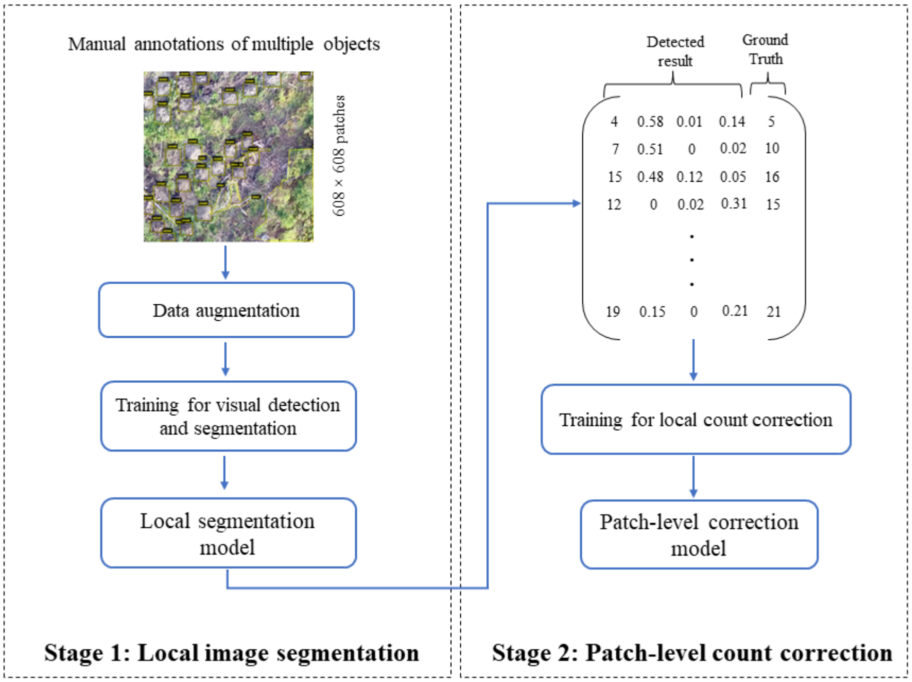

Manual annotations of multiple objects
608 × 608 patches
Data augmentation
Training for visual detection and segmentation
Local segmentation model
Stage 1: Local image segmentation
Detected result
Ground Truth
4 0.58 0.01 0.14 5
7 0.51 0 0.02 10
15 0.48 0.12 0.05 16
12 0 0.02 0.31 15
19 0.15 0 0.21 21
Training for local count correction
Patch-level correction model
Stage 2: Patch-level count correction

Figure 3.3: Pipeline of the system training for two stages.

and recognize objects on each patch of a planting block. The results of this step, which include the number of visually detected mounds and the ratio of other objects (e.g., trees, debris and water), are then fed as input features to the second model for a final patch-level correction.

As stated above and explained further, due to the presence of multiple objects and the limitations of occluded mounds, the efficacy of merely performing local object detection method degrades. Therefore, our two-stage strategy is important for achieving accurate and precise counting under our application constraints. The procedure of analyzing a new patch of an orthomosaic to predict mound counting is depicted in figure 3.4. The details of each stage of our framework are presented in the following sub-sections.

3.2 Local image segmentation

The first step of our method is to perform local instance segmentation to identify and segment multiple objects in each patch. The main motivation for this step is that different plantation blocks have different properties, and mound density highly depends on block characteristics. In this regard,

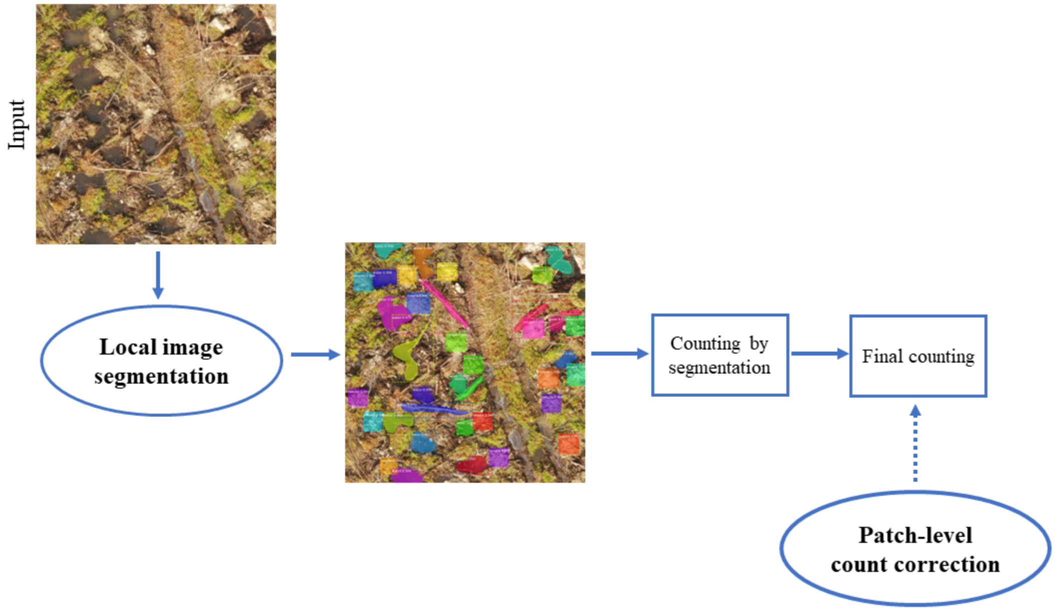


Figure 3.4: The procedure for evaluating a new patch using our framework.

quantifying the presence of mounds and other objects is considered as an indicator of the properties for a given bock. In order to accomplish this, local object instance segmentation is used to detect distinct objects belonging to the same category and to assign a unique instance label to the associated pixels.

Instance object segmentation combines object detection, which outputs bounding box coordinates, and semantic segmentation, which outputs segmentation masks. In this work, we use Mask R-CNN [11] as an instance segmentation to detect mounds and segment all objects in the image for its state-of-the-art (SOTA) performance.

3.2.1 CNN architecture

Mask R-CNN is a cutting-edge instance segmentation technique that adds a segmentation mask generating branch to its predecessor, Faster-RCNN [96], to accomplish proper object detection and pixellevel instance segmentation. We employ Mask R-CNN because it is a two-stage object detector that takes advantage of anchor boxes. Anchor boxes enable this method to detect multiple objects in different scales, which improves the efficiency of the method and provides more accurate localization and classification. Mask R-CNN comprises two stages to produce a final mask segmentation of objects. The first stage scans over the image and generates the proposals, and the second stage predicts the class and box offset and produces a binary mask in parallel [11]. In these two stages, the Mask-RCNN architecture employs three modules: backbone, Region Proposal Network (RPN), and ROI as shown in figure 3.5.

The feature maps are constructed in the first stage by extracting image features of various scales using the backbone network such as ResNet (deep residual networks) [123], which is also known as the feature extraction network. After that, the obtained feature map is sent to the RPN that generates proposals. The RPN has two branches: the first predicts the bounding box, while the second uses SoftMax to do binary classification of foreground and background, with the foreground class implying the presence of a target object in the box. Then non-maximum suppression (NMS) algorithm is used to generate the final proposals (ROIs). Note that the RoIs proposed at this point are obtained from the anchor approach [96]. In the second stage, the corresponding target features of the shared feature maps are extracted by mapping ROIs to feature layers. The RoIAlign is used to modify the feature map to a fixed-size feature map. Finally, the task of mask prediction is completed through FCN branch, and object classification and bounding box regression are completed by two branches of Fully Connected (FC) layers.

3.2.2 Training strategy

In general, deep learning approaches require a significant amount of data to be trained properly; otherwise, these techniques could fail to yield high accuracy. Therefore, due to the lack of training data and the purpose of applying Mask-RCNN as a deep learning-based approach, we apply transfer learning. We trained all of the layers including the RPN, classifier and mask head of our model network using pre-trained weights from the Common Objects in Context (COCO) [140] dataset. In addition to transfer learning, we used data augmentation process to address the issue of the limited number of real-world mounds to improve the recognition rate of our model. In this way, a range of some augmentation techniques were used to increase the diversity of the original training dataset.

Once Mask-RCNN is trained through the adaptation of the preceding methodologies, it is fitted to new datasets using a multi-loss function throughout the learning step. As shown in Eq. (3.1), the goal is to optimize model parameters by minimizing a multi-tasking loss function that incorporates a three-module combination loss: classification, localization, and segmentation.

$$L = L_{cls} + L_{box} + L_{mask}, \tag{3.1}$$

In this equation, L_{cls} represents the loss of classification, L_{box} represents the loss of prediction bounding box, and L_{mask} represents the loss of mask.

Based on the Mask-RCNN network, the mask branch contains a Km^2-dimensional output for each identified ROI that encodes K binary masks with a resolution of $m \times m$, representing K number of classes [11]. Thus, L_{mask} is defined as the average binary cross-entropy loss on the k-th mask, which is calculated using per-pixel sigmoid on $mask_k$, as defined below:

$$L_{mask} = Sigmoid(mask_k). \tag{3.2}$$

Local segmentation is done based on annotated patches of planting blocks for the whole terrain. The number of mounds and the ratio of the other three objects in each patch are then used as input to the local count correction.

3.3 Patch-level correction

The purpose of this stage is to accurately predict the number of mounds in a given orthomosaic representing a planting block. In our first stage, local object detection and segmentation is used to detect visible mounds. However, the number of visible mounds in an orthomosaic rarely corresponds to the actual number of planted seedlings, due to multiple factors, such as occlusion caused by woody debris or tree from neighboring zone (see figure 3.6 (a) and (b)), and destroyed mounds by water flow (see figure 3.6 (c)).

Therefore, the number of detected mounds in a patch is generally underestimated when relying only on detection techniques. To reduce this error, we use regression algorithms based on Mask-RCNN results from the previous stage. The objective of regression analysis in this context is to

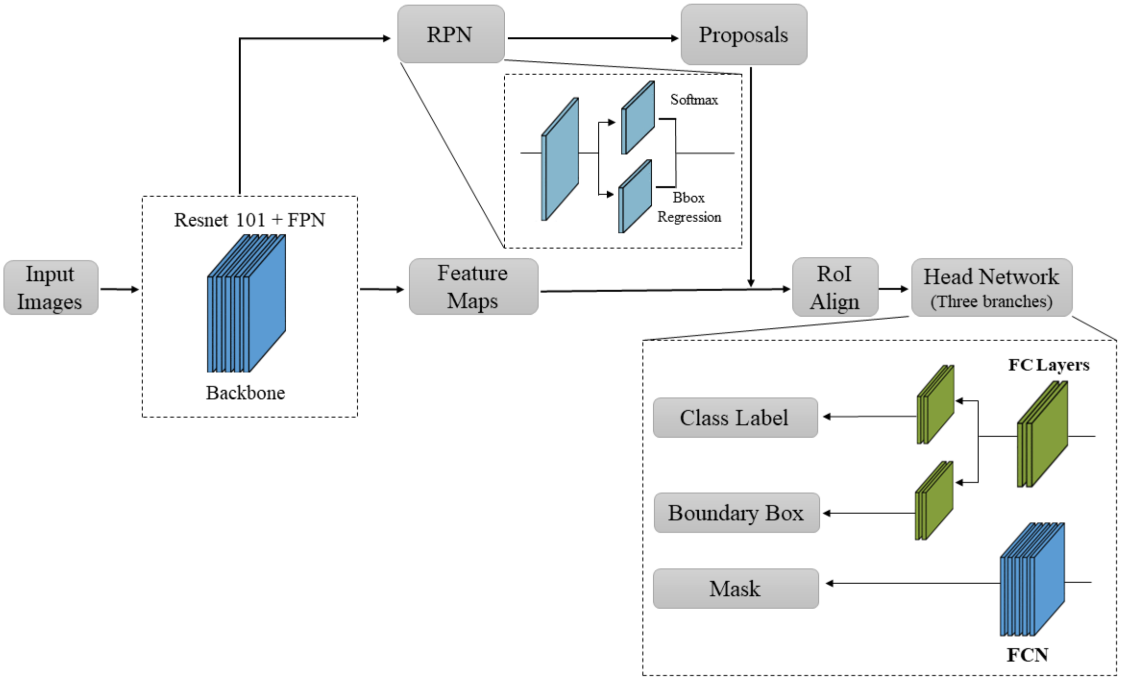


Figure 3.5: Architecture of Mask-RCNN.

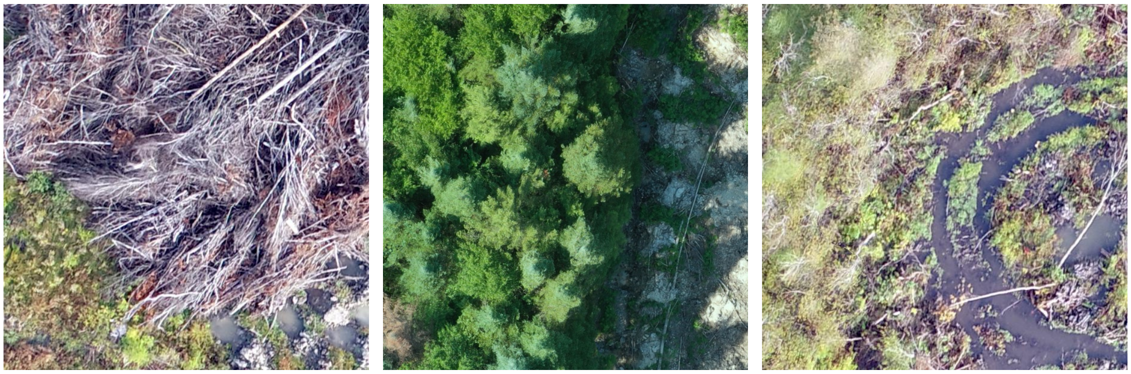


Figure 3.6: Examples of mound occlusion and destruction.
(a) Occlusion due to the presence of woody debris. (b) Presence of tree from neighbor zone. (c) Destroyed mounds by water flow due to heavy rain.

learn a function $X \rightarrow Y$ from N sample images as follows:

$$\begin{aligned} &\mathrm{X} = \left\{x_i^j\right\}, \mathrm{j} \in (1,4), \mathrm{i} = 1,2,\ldots,\mathrm{N} \\ &\mathrm{Y} = \left\{\mathrm{y_i}\right\}, \mathrm{i} = 1,2,\ldots,\mathrm{N} \end{aligned} \tag{3.3}$$

where x^1 represents the number of detected mounds during the first stage and x^2, x^3, and x^4 represent the ratios of trees, water, and debris respectively, and Y is defined as the final number of mounds from the ground-truth.

Regression methods, including linear, Support Vector Regression (SVR), lasso, and Multilayer Perceptron (MLP), were investigated and performed to find the more accurate predictor with a high Relative Counting Precision (RCP). We finally adopted SVR as a predictive regression technique to correct the mounds count on patches of any given block. To obtain the final mounds count, we used the image segmentation results of the first stage, which contains the number of mounds, and the ratio of trees, debris, and water.

SVR is an expanded version of the Support Vector Machine (SVM) model that adds an ϵ-insensitive error function to provide it with the ability to perform regression [141]. It gives the freedom to find the proper hyperplane in higher dimensions to regress the data and customize controlled errors in a reasonable range. SVR is efficient for nonlinear modeling because it reduces model complexity and prediction errors using a kernel function [142]. Additionally, SVR has proven to manage dimensionality effectively in the small dataset [143].

To describe the function of the model, $S = \{(x_1, y_1), (x_2, y_2), \ldots (x_l, y_l)\}$ is taken to be the set of training data in the next n input space, where $x_i \in R^n$. The purpose is to represent the original non-linear data in a high-dimensional space. Thus, a mapping function is applied to regress sample data in the higher-dimensional feature space. The regression function of $f(x)$ is described as:

$$f(x) = \omega\varphi(x) + b \tag{3.4}$$

where ω is the weight vector; φ is the nonlinear mapping function, and b is the offset. The convex optimization problem described in Eq. (3.5 and 3.6) can be used to represent the entire issue:

$$\text{Minimize}\ \frac{1}{2}\|\mathrm{w}\|^2+\mathrm{C}\sum_{\mathrm{i}=0}^{1}\left(\xi+\xi^*\right) \tag{3.5}$$

$$\text{Subjectedto}\begin{cases}\mathrm{y_i}-\langle\mathrm{w},\mathrm{x_i}\rangle-b\leq\epsilon_{\mathrm{i}}+\xi\\ \langle\mathrm{w},\mathrm{x_i}\rangle+b-\mathrm{y_i}\leq\epsilon_{\mathrm{i}}+\xi^*\\ \xi_{\mathrm{i}},\xi_{\mathrm{i}}^*\geq 0i=1,\cdots,l\end{cases} \tag{3.6}$$

where ξ and ξ^* are the slack variables represent the positive and negative errors at the *i*th point, respectively, the constant C represents the penalty coefficient, and ϵ is the deviation between the estimated value and the target value.

To determine the final total number of mounds for each block, we used the outcomes of the algorithms predictions for each patch as follow:

$$\text{Count}_{\text{final}}\left(\text{Block}_i\right)=\sum_{j=1}^{M}\left(p_j\right) \tag{3.7}$$

where p_j represents the j-th patch for a given Block(i).

Chapter 4

Experiments

4.1 Dataset construction

The study area of this work is located in the south of the province of Quebec, Canada, as shown in figure 4.1. The aerial images were collected by flying over private forest sites. Using an excavator with a 45 cm-wide bucket, mounds with heights of about 50 cm and diameters of 80 cm were constructed. In each mound, a fast-growing hybrid poplar clone seedling was intended to be planted.

The aerial RGB images were taken with the DJI Matrice 100 quadcopter equipped with a vertical high-resolution Zenmuse X3 Visual (R, G, B) sensor. Images were captured with a high overlap percentage to maximize orthomosaic reconstruction quality at 120 meters. Using Pix4D software, orthomosaic reconstruction was carried out.

We employed 20 orthomosaics—reconstructed from UAV images of various zones—with varying characteristics as input data. We divided our images into two distinct groups as follows:

- **Group 1**: consists of 4 training orthomosaics manually annotated for the local image segmentation. Three orthomosaics were used for training, and the fourth was used to test the segmentation method.

- **Group 2**: includes 18 testing orthomosaics used to evaluate the performance for the entire framework.

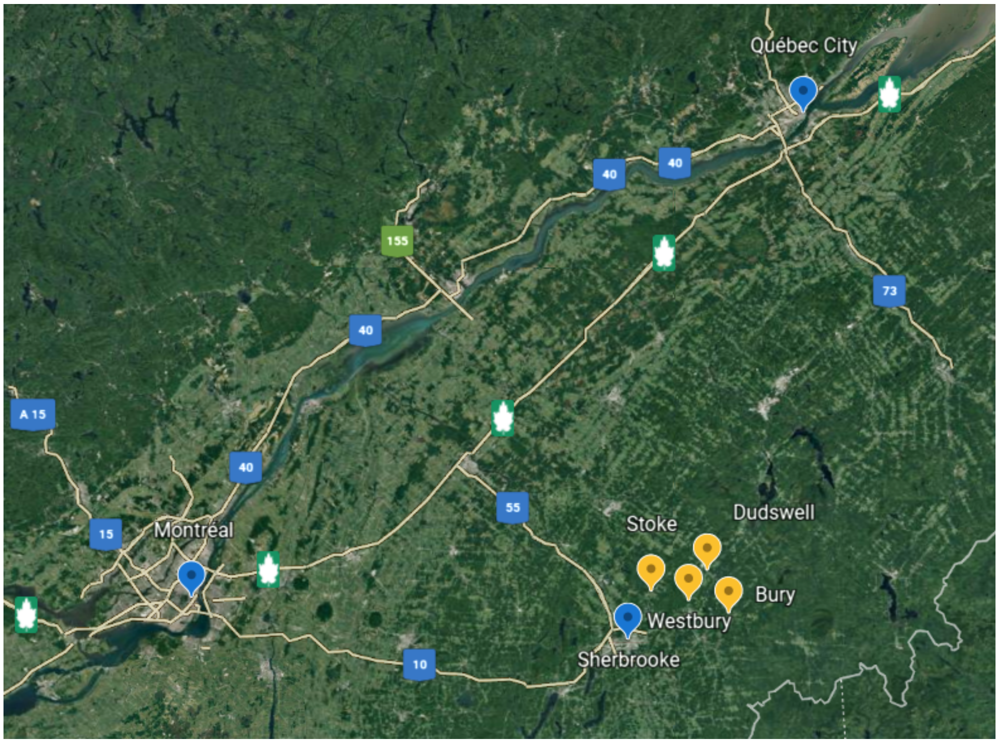
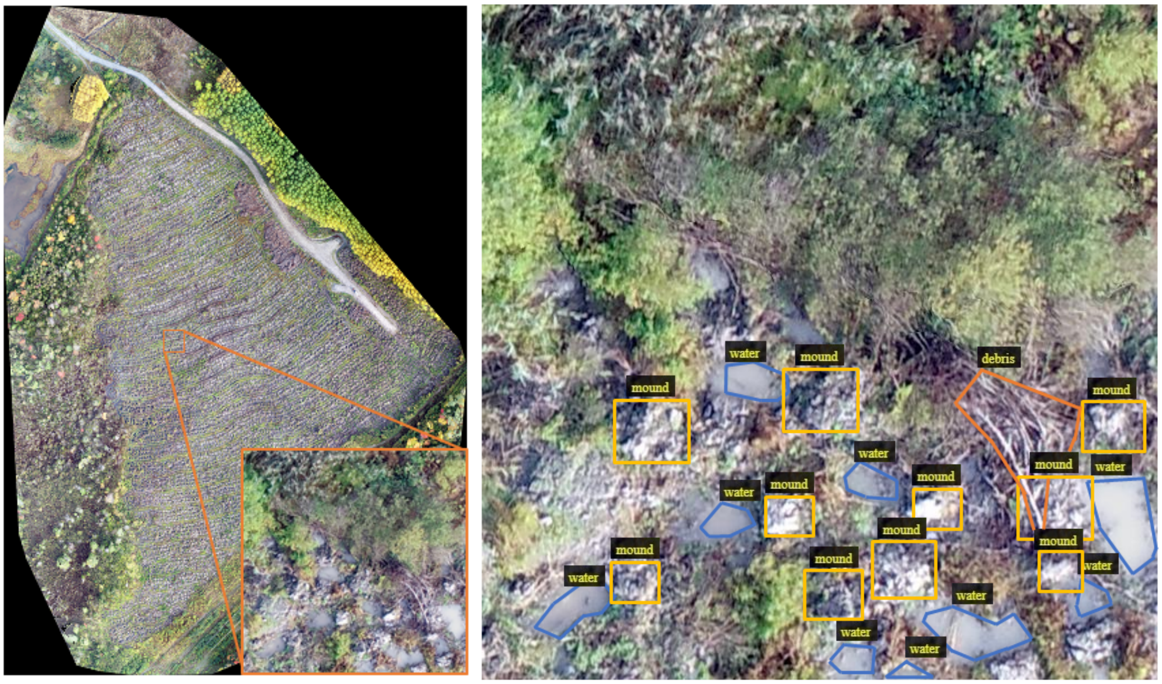


Figure 4.1: Geographic zone of the study. The yellow regions show the study area south of Quebec, Sherbrooke city.

Because the sensor produced images with a high resolution of 23610 × 18151, we adopted a patch-based approach to trim orthomosaic and create non-overlapped patches with regular and stable pixel sizes of 608 × 608. Moreover, since a given plantation block may have various properties for different regions, dividing the orthomosaic into patches is important in order to determine the properties of each region separately. As a result, 1352 patches were employed to train the model. The data was then processed before being fed into the local instance segmentation method for training. The region of interest was carefully investigated. The ground-truth of mounds and other objects, including trees, water, and woody debris, was manually annotated using the open-source VGG Image Annotator (VIA) tool [144]. The JSON file generated by the VIA tool contains information on the masks as a set of rectangles (for mound) and polygon points (for water, debris, and tree). We handled bounding boxes uniformly independent of the source dataset by computing the bounding boxes using masks rather than the bounding box coordinates supplied by the source datasets. Figure 4.2 illustrates a patch with manually annotated objects cropped to fixed cell size.



(a)

(b)

Figure 4.2: Manually annotated objects. (a) Example of one orthomosaic and a sample patch cropped to a fixed dimension. (b) Manually annotated objects (mound, tree, water, debris).

4.2 System training and testing settings

The proposed method was implemented using Python on a PC (CPU i7-12700KF @ 3.61 GHz, 12 cores) equipped with a GPU NVIDIA GeForce RTX 3060. We trained the Mask-RCNN as the segmentation method on each orthomosaic constructed from UAV images. To train the model, we set the batch size to 1 and the learning rate to 0.001, with a decay of 0.001. The momentum is set to 0.9, and the number of epochs is fixed to 150 for pixel-wise detector training. Transfer learning for Mask-RCNN is performed on the weights of the pre-trained model of the COCO dataset [140]. Throughout the segmentation process, we set a confidence threshold of 0.5 to identify possible mounds without increasing the number of false positives. After model training, the results of the Mask-RCNN, which included the number of mounds, the ratio of trees, debris, and water buildup, were supplemented with the ground-truth number of mounds and then fed to the regression prediction method. The linear kernel function was used to train the SVR with the regularization parameter C of 1.0 and the epsilon tube of 0.5.

4.3 Evaluation metrics

The performance of the visual object instance detection and segmentation module is evaluated through mean Average Precision (mAP), which is calculated using the confusion matrix and the following sub metrics of precision (P) and recall (R):

$$P = \frac{TP}{TP + FP} \tag{4.1}$$

$$R = \frac{TP}{TP + FN} \tag{4.2}$$

where TP indicates the number of True Positives, FP indicates the number of False Positives, and FN indicates the number of False Negatives.

The area under the Precision-Recall curve is represented by Average Precision (AP) that is calculated for each class using the Intersection over Union () as a metric for measuring the overlap between ground-truth and the predicted mask as below:

$$AP = \sum_{i=0}^{n-1} \left(R_{i+1} - R_i\right) P_{\text{interp}} \left(R_{i+1}\right), \tag{4.3}$$

where $Pinterp(R)$ is the precision interpolated at a certain recall level.

Given that AP_i^x is the average precision for a given IOU threshold of x and class i, mAP for N number of classes is defined by:

$$\text{mAP} = \frac{\sum_{i=1}^{N} AP_i^x}{N} \tag{4.4}$$

In our forestry application, the main objective is to predict the number of mounds on each plantation block, regardless of their positions or distribution. Therefore, we used the relative counting precision metric, which is the most critical success indicator of our method, to measure the overall system performance by evaluating the regression predictors and obtaining a final counting estimate, as shown below:

$$RCP = 1 - \left| \frac{\#predicted\ mound - \#gt}{\#gt} \right| \tag{4.5}$$

where *#predicted-mound* represents the predicted number of mounds and $\#gt$ represents the number of mounds from the ground-truth.

4.4 Experimental results

The process of counting mounds on a new planting block comprised two steps:

1. Applying the local image segmentation model to detect visible mounds and quantify the presence of trees, debris and water.

2. Performing patch-level correction using a regression function.

Note that we only trained our local image segmentation using three annotated orthomosaics from Group 1. To ensure that the models work properly and verify the performance of our proposed method, we used 18 orthomosaics from Group 2 that had not been utilized in the training processes.

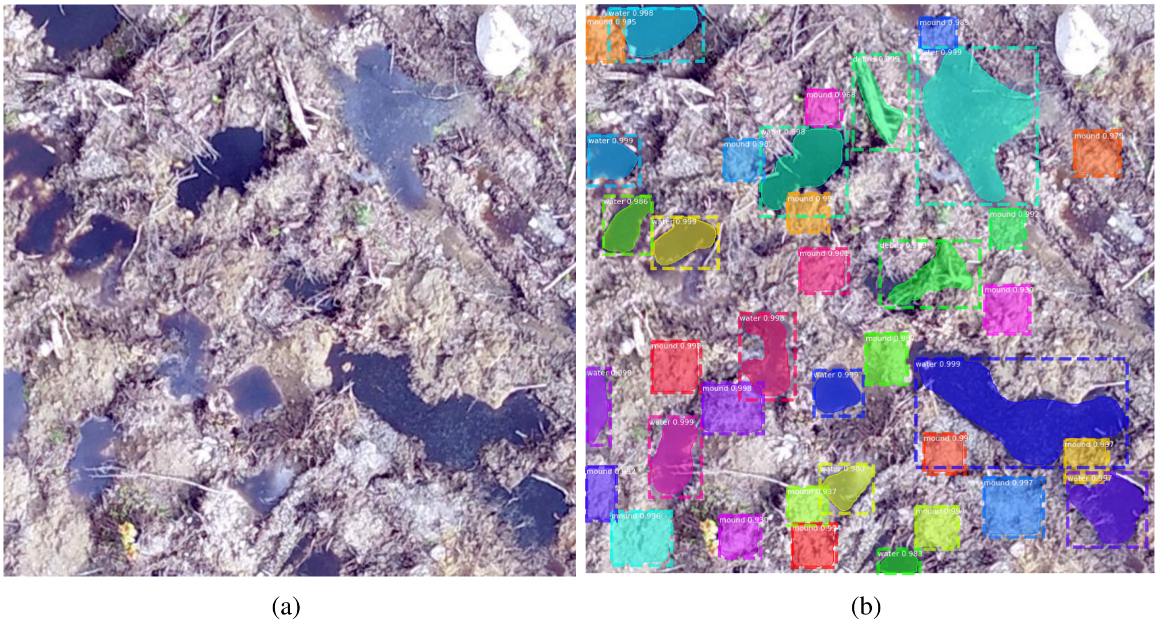
Table 4.1 shows the global quantitative results of testing data for evaluating the segmentation algorithm using mAP metric for all patches of the fourth dataset of group 1. According to the table 4.1, the global mAP at 50% and 75% for all 61 patches are 52% and 16%, respectively.

Table 4.1: Quantitative result of global mAP

Total Number of Patches	**Global mAP 50**	**Global mAP 75**
60	52%	16%

Figure 4.3 depicts one sample patch and its corresponding qualitative result. The number of mounds and the ratio of other detected and segmented objects were then used as input features in the second step of the pre-trained regression algorithm to produce the final mound counting.

Table 4.2 shows the quantitative results of our proposed approach. According to the findings, the RCP of the local image segmentation method is 93%, which indicates that our instance segmentation method has the ability to detect and segment the planting microsites efficiently. However, we could



(a) (b)

Figure 4.3: Qualitative result of one sample patch and its corresponding.
(a) Example of one patch. (b) Corresponding qualitative result.

significantly improve the average detection precision of counting mounds by applying patch-level correction methods. From table 4.2, this improvement reached 96% by performing SVR. Compared with other regression methods, SVR yields the greatest results since it employs a kernel function to concurrently minimize prediction errors and model complexity.

Generally, the total ground-truth number of mounds for all 18 blocks is 125054. The local segmentation-based method proved its capability to correctly detect and segment 115968 mounds with an RCP of 93%. However, the mounds were rectified at the patch level with an RCP of 96% utilizing SVR as a regression analysis approach at the second stage, representing an increase of 3% for the 18 planting blocks examined in this work. The results confirm that regression methods in the second stage could at least improve the estimated number of mounds by 1% compared to simply using Mask-RCNN as the pixel-wise detection approach.

The corrected number of mounds in blocks 02, 05, and 14 is 96%, according to the RCP result of the SVR technique. However, the results of patch-level correction for blocks 03, 09, 10, 12, 13, 15, 16, and 17 outperformed an average precision of 96%.

Table 4.2: Quantitative results of our proposed approach. GroundTruth refers to the final number of plant seedlings planted in a block. Local segmentation-based count is the number of mounds detected and segmented using local image segmentation method. Count is the number of locally corrected mounds and RCP corresponds to the relative counting precision. Average precision measure represents the average over all precision values, but the overall result indicates the counting precision when the entire number of mounds in the dataset is taken into account.

Orthomosaic	**GroundTruth**	**Local segmentation-based count**		**Patch-level corrected count**							
				Linear		***SVR***		***Lasso***		***MLP***	
		Count	***RCP***	***Count***	***RCP***[a]	***Count***	***RCP***	***Count***	***RCP***	***Count***	***RCP***
Block 01	16450	14458	88%	15820	96%	15180	92%	15504	94%	15828	96%
Block 02	2650	2609	98%	2816	94%	2760	96%	2851	92%	2780	95%
Block 03	750	712	95%	724	97%	737	98%	771	97%	728	97%
Block 04	800	784	98%	851	94%	844	94%	891	89%	837	95%
Block 05	2350	2233	95%	2495	94%	2436	96%	2562	91%	2462	95%
Block 06	1700	1513	89%	1623	95%	1600	94%	1683	99%	1621	95%
Block 07	2050	1853	90%	1868	91%	1879	92%	1933	94%	1864	91%
Block 08	3950	3443	87%	3676	93%	3571	90%	3649	92%	3629	92%
Block 09	6847	6632	97%	7041	97%	6915	99%	7091	96%	6923	99%
Block 10	30200	28301	94%	28973	96%	29145	97%	30107	99.7%	28733	95%
Block 11	2950	2742	93%	2778	94%	2797	95%	2894	98%	2765	94%
Block 12	25450	24251	95%	2765	96%	25447	99.99%	25994	98%	25848	98%
Block 13	7400	6658	90%	7825	94%	7551	98%	8079	91%	7824	94%
Block 14	5250	5009	95%	5620	93%	5468	96%	5751	90%	5563	94%
Block 15	3557	3424	96%	3636	98%	3653	97%	3842	92%	3643	98%
Block 16	5150	4320	84%	5362	96%	5032	98%	5418	95%	5331	96%
Block 17	4900	4759	97%	5164	95%	5025	97%	5236	93%	5128	95%
Block 18	2650	2267	86%	2478	94%	2492	94%	2670	99.2%	2471	93%
Overall result	125054	115968	93%	101515	81%	122532	98%	126926	99%	123978	99%
Average precision			93%		95%		96%		94%		95%

[a] Highlighted numbers in red correspond to best precision results, and numbers in blue represent second-best results.

Although the experimental results show the efficiency of our strategy, this thesis is subject to several challenges. Based on the results of local-segmentation mound counting, the RCP for blocks 01, 06, 08, 16, and 18 are less than 90% compared to the other ones, which are equal to or greater than 90%. This can be caused by the fact that the new plantation block may exhibit many unseen properties when we test our object-pixel-wise instance detector. This could include mound shapes and block characteristics that the detector was not exposed to during training. The performance of the segmentation-based count is substantially impacted by occlusion, destruction, and image acquisition in addition to the model being exposed to unseen characteristics. In addition, table 4.2 indicates that while our framework was able to improve the final results by applying SVR, the outcomes for some blocks are lower than others. As can be seen from block 08, our correction method could only correct 128 more mounds than the local approach, bringing the total number of mounds from 3443 up to 3571, which happened due to the challenging situations on the captured images of the related blocks. For instance, because the images of block 08 were taken under dry conditions, the texture of mounds is similar to surrounding regions, which makes object detection very challenging.

Despite the aforementioned difficulties, the overall results demonstrate that using the two stages consecutively results in an average improvement of 3%. Additionally, our framework achieved an average RCP of 96%, and thus outperforms the manual method whose RCP is around 85%.

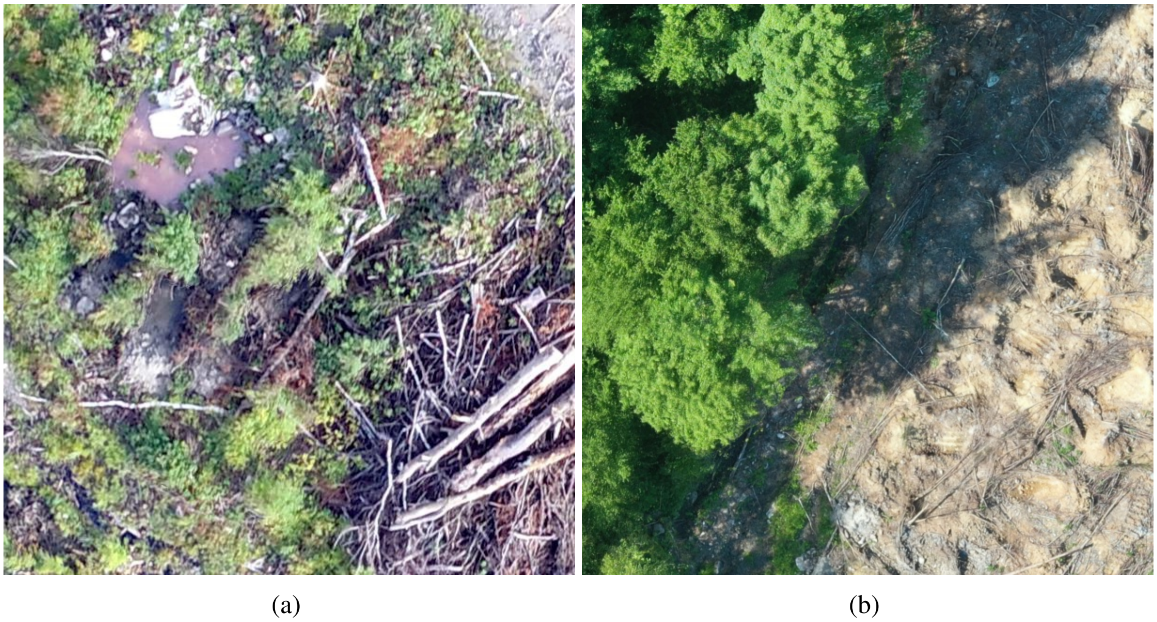
4.5 Limiting factors: discussion

From table 4.2, the RCP results of the segmentation-based count method are less than 90% for certain blocks (01, 06, 08, 16, and 18). As previously mentioned in the experimental results (section 4.4), we employed a new plantation block for the testing phase that was entirely excluded from the training dataset. Thus, the results of the object counting may be underestimated since the new planting block has several characteristics that are different from those in the input dataset used for training. Additionally, other key factors including occlusion, destruction, and image acquisition significantly impact detection performance.

Occlusion: Occlusion is considered as one of the most common issues that reduce the available visual information. In an image, occlusion happens when one object obscures a portion of another. The occluded areas are determined based on the camera's position in relation to the scene. As shown in figure 4.4, the presence of woody debris on the forest floor may cause occlusion. Additionally, trees or shadows cast by nearby zones may cover mounds that are positioned on block borders.

Mound destruction: The tracks of the excavator, which are only employed for scalping on sites with steep slopes, heavy slash, high stumps, or if a range of site preparation treatments is necessary, may induce mound destruction during the mechanical preparation of a planting block as shown in figure 4.5. The constructed mounds may also be deteriorated and eroded during the event of strong rainstorms following mechanical preparation. This happened due to the fact that intense silviculture frequently places planting blocks along hillsides, which favors water flow.

Image capture factors: The image capture process is a crucial phase that can have a substantial impact on the appearance of the object of interest as well as the image quality. This is mostly caused by a number of weather-related flight disruption factors, such as camera movements in windy weather and changes in lighting, as show in figure 4.6. For instance, altering the height or quickly moving a drone in windy conditions can greatly reduce the quality of the images that are captured. Flying at high altitude makes it possible to streamline image acquisition and orthomosaic generation while flying at low altitude increases the level of detail in photos. Consequently, a wise compromise needs to be made. Furthermore, the presence or lack of sun glint reflection results in a poor-quality image as well, which makes it difficult to discern between mounds and background topography, and in some situations, leads in object scale variation.


(a)

(b)

Figure 4.4: Illustrative instances of occlusion.
(a) Occlusion caused by woody debris or rock fragments.
(b) Occlusion caused by the presence of trees and shadows from neighboring zone.

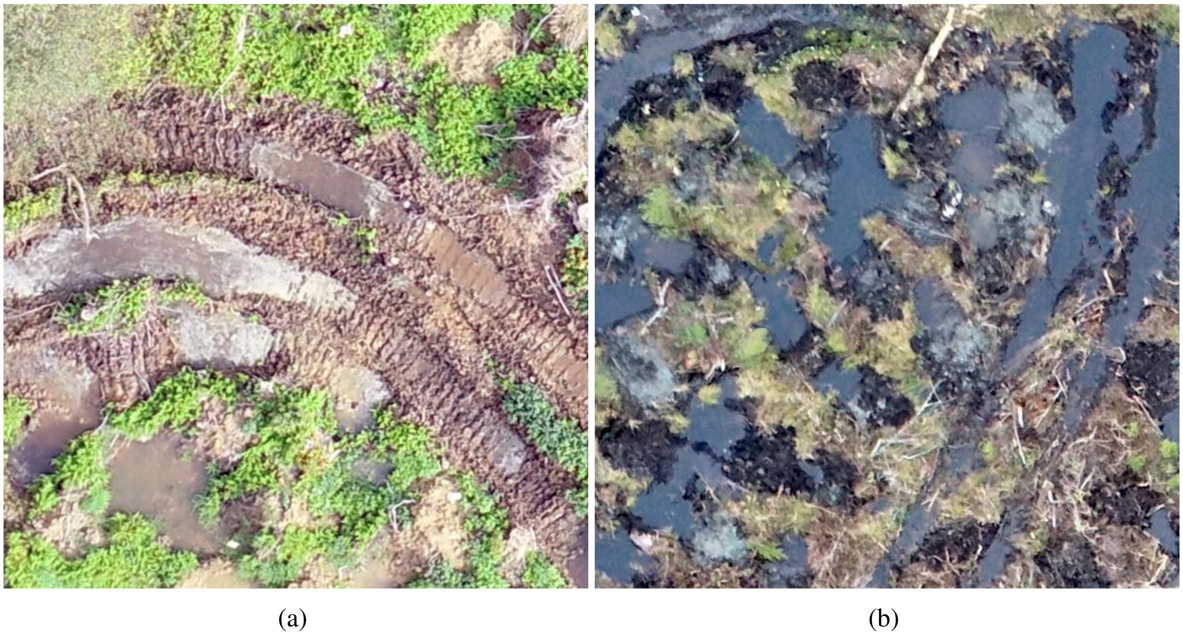

(a) (b)

Figure 4.5: Illustrations of destroyed mounds.
(a) Mounds that were destroyed during mechanical preparation by the excavator. (b) Mound destruction due to heavy rain.

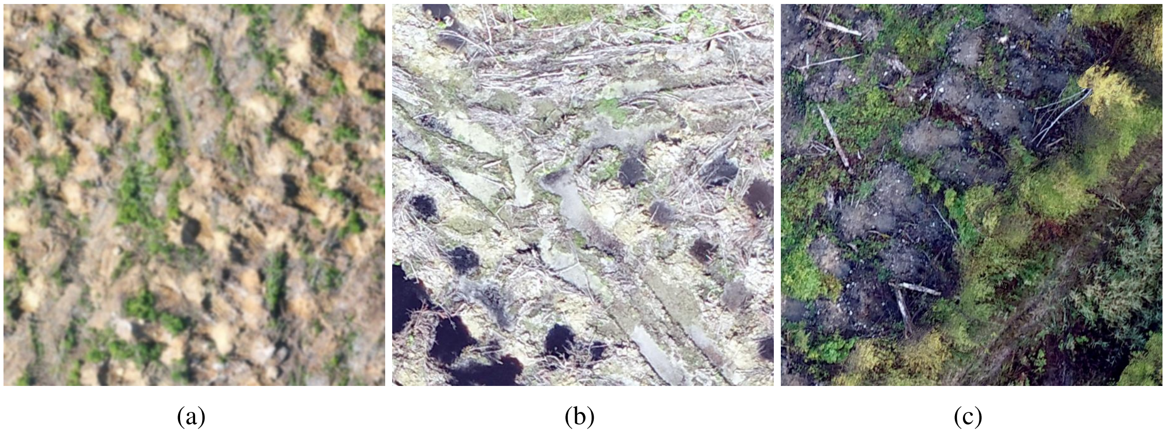

(a) (b) (c)

Figure 4.6: Examples of visual impacts by image acquisition conditions.
(a) Blurred image. (b) Bright image due to the sun glint reflection. (c) Dark image due to the lack of luminosity.

Chapter 5

Conclusion

Mechanical site preparation by mounding is an important step in silviculture and tree regeneration, which promotes the survival and early growth of newly planted seedlings and raises the productivity of agricultural fields. The highest level of agricultural land productivity also boosts the productivity of forestry companies. The precision of planning processes, including methods for gathering and interpreting field data, is crucial to achieve this goal. Despite the increasing usage of new technologies, existing planning techniques still involve human manual operations that are frequently expensive, time-consuming, and prone to errors. These problems stand out when planting activities are planned, and an estimation of the number of mounds on each mechanically prepared block is required. In particular, mound counting is of great significance in forestry, since their number can vary substantially depending on site characteristics.

To automate the process of counting mounds, we proposed a new computer vision framework for supporting forestry managers. Our framework is based on UAV imagery and machine learning approaches. Indeed, we developed an automated approach to detect and segment mounds and then estimate their precise number.

The proposed system adopts an hybrid approach by integrating a local image segmentation method with patch-level count correction. The objective is to predict the number of tree seedlings to be planted. To this aim, local image segmentation is used first to quantify the presence of objects, including the number of visible mounds and the ratio of other objects (e.g., trees, debris, and water). Since the visual detection and segmentation approach is subject to recognition errors under challenging situations, our main objective cannot be accomplished by merely using local image segmentation. Therefore, we apply regression analysis in the second stage to correct the number of the mound at the patch-level by using information extracted at the pixel-level from the first stage.

The experimental results demonstrate that our approach is effective in dealing with a variety of difficult conditions involving environmental factors and the existence of multiple objects on plantation blocks. That is, the local segmentation approach leverages visual mounds from aerial images to predict a preliminary count, which is subsequently corrected by the patch-level correction method. According to our qualitative and quantitative performance assessments, our approach outperforms the manual counting method, commonly used in forestry in terms of precision, while significantly reducing the financial cost of planning planting operations.

During this experiment, we dealt with some limitations related to image data and the mechanism of annotating objects in each patches. As we previously mentioned, certain disturbance factors, such as occlusion, mound destruction, as well as image acquisition problems may impact the detection process. We also observed that the overall performance depends on the amount and the quality of annotations. Therefore, the main direction for future work is to consider situations where images are weakly annotated in order to expand the training dataset.

References

[1] Robert F. Sutton. Mounding site preparation: A review of european and north american experience. *New Forests)*, 7(2):151–192, 1993.

[2] Magnus Löf, Daniel C. Dey, Rafael M. Navarro, and Douglass F. Jacobs. Mechanical site preparation for forest restoration. *New Forests*, 43(5):825-848, 2012.

[3] Robert John Stathers, Trowbridge R, D. L. Spittlehouse, A. Macadam, and J.P. Kimmins. Ecological principles: basic concepts. *In Regenerating British Columbia's Forests. D.P. Lavender et. al.(editors). Univ. B.C. Press, Vancouver, B.C.*, pages 45–54, 1990.

[4] B. J. Sutherland and F. F Foreman. Guide to the use of mechanical site preparation equipment in northwestern ontario. *Great Lakes Forestry Centre*, 1995.

[5] Robert John Stathers and David Leslie Spittlehouse. Forest soil temperature manual. *FRDA Research Program, Research Branch, BC Ministry of Forests and Lands*, (130), 1990.

[6] J.M. Caborn. Microclimates. *Endeavour*, 32(115):30–33, 1973.

[7] Karin Hjelm, Urban Nilsson, Ulf Johansson, and Per Nordin. Effects of mechanical site preparation and slash removal on long-term productivity of conifer plantations in sweden. *Canadian Journal of Forest Research*, 49(10):1311–1319, 2019.

[8] Lav Gupta, Raj Jain, and Gabor Vaszkun. Survey of important issues in uav communication networks. *IEEE Communications Surveys Tutorials*, 18(2):1123–1152, 2015.

[9] Wassim Bouachir, Koffi Eddy Ihou, Houssem-Eddine Gueziri, Nizar Bouguila, and Nicolas Bélanger. Computer vision system for automatic counting of planting microsites using uav imagery. *IEEE Access*, 7:82491-82500, 2019.

[10] Majid Nikougoftar Nategh, Ahmed Zgaren, Wassim Bouachir, and Nizar Bouguila. Automatic counting of mounds on uav images: combining instance segmentation and patch-level correction. *IEEE International Conference on Machine Learning and Applications (ICMLA)*, 2022.

[11] Kaiming He, Georgia Gkioxari, Piotr Dollar, and Ross Girshick. Mask r-cnn. *IEEE International Conference on Computer Vision (ICCV)*, pages 2980–2988, 2017.

[12] Hao Jiang, Shuisen Chen, Dan Li, Chongyang Wang, and Ji Yang. Papaya tree detection with uav images using a gpu-accelerated scale-space filtering method. *Remote Sensing*, 9(7):721, 2017.

[13] Lina Tang and Guofan Shao. Drone remote sensing for forestry research and practices. *Journal of Forestry Research*, 26(4):791–797, 2015.

[14] Luke Wallace, Arko Lucieer, Zbynˇek Malenovský, Darren Turner, and Petr Vopˇenka. Assessment of forest structure using two uav techniques: A comparison of airborne laser scanning and structure from motion (sfm) point clouds. *Forests*, 7(3):62, 2016.

[15] Barreto Abel, Philipp Lottes, Facundo Ramón Ispizua Yamati, Stephen Baumgarten, Nina Anastasia Wolf, Cyrill Stachniss, Anne-Katrin Mahlein, and Stefan Paulus. Automatic uav-based counting of seedlings in sugar-beet field and extension to maize and strawberry. *Computers and Electronics in Agriculture*, 191:106493, 2021.

[16] Ertugrul Bayraktar, Muhammed Enes Basarkan, and Numan Celebi. A low-cost uav framework towards ornamental plant detection and counting in the wild. *ISPRS Journal of Photogrammetry and Remote Sensing*, 167:1–11, 2020.

[17] Alejandro Navarro, Mary Young, Blake Allan, Paul Carnell, Peter Macreadie, and Daniel Ierodiaconou. The application of unmanned aerial vehicles (uavs) to estimate above-ground biomass of mangrove ecosystems. *Remote Sensing of Environment*, 242:111747, 2020.

[18] Dilek Koc-San, Serdar Selim, Nagihan Aslan, and Bekir Taner San. Automatic citrus tree extraction from uav images and digital surface models using circular hough transform. *Computers and electronics in agriculture*, 150:289–301, 2018.

[19] Susana Baena, Justin Moat, Oliver Whaley, and Doreen S. Boyd. Identifying species from the air: Uavs and the very high resolution challenge for plant conservation. *PloS one*, 12(11), e0188714, 2017.

[20] Jung il Shin, Won woo Seo, Taejung Kim, Joowon Park, and Choong shik Woo. Using uav multispectral images for classification of forest burn severity—a case study of the 2019 gangneung forest fire. *Forests*, 10(11):1025, 2019.

[21] Pablo Chamoso, William Raveane, Victor Parra, and Angélica González. Uavs applied to the counting and monitoring of animals. *In Ambient intelligence-software and applications*, pages 71–80, 2014.

[22] Wang Jingying. A survey on crowd counting methods and datasets. *Advances in Computer, Communication and Computational Sciences*, pages 851–863, 2012.

[23] Jian Cheng, Haipeng Xiong, Zhiguo Cao, and Hao Lu. Decoupled two-stage crowd counting and beyond. *IEEE Transactions on Image Processing*, 30:2862–2875, 2021.

[24] Yi Wang, Junhui Hou, Xinyu Hou, and Lap-Pui Chau. A self-training approach for point-supervised object detection and counting in crowds. *IEEE Transactions on Image Processing*, 30:2876–2887, 2021.

[25] Weihang Kong, He Li, Guanglong Xing, and Fengda Zhao. An automatic scale-adaptive approach with attention mechanism-based crowd spatial information for crowd counting. *IEEE Access*, 7:66215–66225, 2019.

[26] Sami Abdulla Mohsen Saleh, Shahrel Azmin Suandi, and Haidi Ibrahim. Recent survey on crowd density estimation and counting for visual surveillance. *Engineering Applications of Artificial Intelligence*, 41:103–114, 2015.

[27] Mahnaz Razavi, Hadi Sadoghi Yazdi, and Amir Hossein Taherinia. Crowd analysis using bayesian risk kernel density estimation. *Engineering Applications of Artificial Intelligence*, 82:282–293, 2019.

[28] Tao Zhang, Jiawei Yuan, Yeh-Cheng Chen, and Wenjing Jia. Self-learning soft computing algorithms for prediction machines of estimating crowd density. *Applied Soft Computing*, 105:107240, 2021.

[29] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, 1:886-893, 2005.

[30] Paul Viola and Michael J. Jones. Robust real-time face detection. 57(2):137-154, 2004.

[31] Bo Wu and Ram Nevatia. Detection and tracking of multiple, partially occluded humans by bayesian combination of edgelet based part detectors. *International Journal of Computer Vision*, 75(2):247-266, 2007.

[32] Yi Wang, Junhui Hou, Xinyu Hou, and Lap-Pui Chau. A self-training approach for point-supervised object detection and counting in crowds. *IEEE Transactions on Image Processing*, 30:2876–2887, 2021.

[33] Payam Sabzmeydani and Greg Mori. Detecting pedestrians by learning shapelet features. *IEEE Conference on Computer Vision and Pattern Recognition*, page 1–8, 2007.

[34] Joel P. Ilao and Macario O. Cordel. Crowd estimation using region-specific hog with svm. *International Joint Conference on Computer Science and Software Engineering (JCSSE)*, page 1–5, 2018.

[35] Bingyin Zhou, Ming Lu, and Yonggang Wang. Counting people using gradient boosted trees. *IEEE Information Technology, Networking, Electronic and Automation Control Conference*, page 391–395, 2016.

[36] Viet-Quoc Pham, Tatsuo Kozakaya, Osamu Yamaguchi, and Ryuzo Okada. Count forest: Co-voting uncertain number of targets using random forest for crowd density estimation. *In Proceedings of IEEE International Conference on Computer Vision (ICCV)*, page 3253–3261, 2015.

[37] Jens Rittscher, Peter H. Tu, and Nils Krahnstoever. Simultaneous estimation of segmentation and shape. *In 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2:486–493, 2005.

[38] Xiaoming Liu, Peter Henry Tu, Jens Rittscher, Amitha Perera, and Nils Krahnstoever. Detecting and counting people in surveillance applications. *In IEEE Conference on Advanced Video and Signal Based Surveillance*, pages 306–311, 2005.

[39] Bastian Leibe, Edgar Seemann, and Bernt Schiele. Pedestrian detection in crowded scenes. *In Proceedings of the IEEE computer society conference on computer vision and pattern recognition (CVPR)*, 1:878–885, 2005.

[40] Michael J. Jones and Daniel Snow. Pedestrian detection using boosted features over many frames. *International Conference on Pattern Recognition*, pages 1–4, 2008.

[41] Paul Viola, Michael J. Jones, and Daniel Snow. Detecting pedestrians using patterns of motion and appearance. *International Journal of Computer Vision*, 63(2):153–161, 2005.

[42] Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55(1):119–139, 1997.

[43] Sheng-Fuu Lin, Jaw-Yeh Chen, and Hung-Xin Chao. Estimation of number of people in crowded scenes using perspective transformation. *EEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 31(6):645–654, 2001.

[44] Gabriel J. Brostow and Roberto Cipolla. Unsupervised bayesian detection of independent motion in crowds. *In Proceedings of the IEEE computer society conference on computer vision and pattern recognition (CVPR)*, 1:594–601, 2006.

[45] Oliver Sidla, Yuriy Lypetskyy, Norbert Brandle, and Stefan Seer. Pedestrian detection and tracking for counting applications in crowded situations. *In Proceedings of the IEEE International Conference on Video and Signal Based Surveillance*, 70, 2006.

[46] Anil M. Cheriyadat, Budhendra L. Bhaduri, and Richard J. Radke. Detecting multiple moving objects in crowded environments with coherent motion regions. *In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 1–8, 2008.

[47] Zizhu Fan, Hong Zhang, Zheng Zhang, Guangming Lu, Yudong Zhang, and Yaowei Wang. A survey of crowd counting and density estimation based on convolutional neural network. *Neurocomputing*, 472:224-251, 2022.

[48] Nikos Paragios and Visvanathan Ramesh. A mrf-based approach for real-time subway monitoring. *In Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR*, 1:I-I, 2001.

[49] Antoni B. Chan, Zhang-Sheng John Liang, and Nuno Vasconcelos. Privacy preserving crowd monitoring: Counting people without people models or tracking. *IEEE Conference on Computer Vision and Pattern Recognition*, page 1–7, 2008.

[50] Ya-Li Hou and Grantham KH Pang. People counting and human detection in a challenging situation. *IEEE transactions on systems, man, and cybernetics-part a: systems and humans*, 41(1):24–33, 2010.

[51] Zhang-Sheng John Liang Chan, Antoni B. and Nuno Vasconcelos. Privacy preserving crowd monitoring: Counting people without people models or tracking. *In Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–7, 2008.

[52] David Ryan, Simon Denman, Clinton Fookes, and Sridha Sridharan. Crowd counting using multiple local features. *In Proceedings of digital image computing: techniques and applications*, pages 81–88, 2009.

[53] Ruihua Ma, Liyuan Liand Weimin Huang, and Qi Tian. On pixel count based crowd density estimation for visual surveillance. *In Proceedings of the IEEE Conference on Cybernetics and Intelligent Systems*, 1:170–173, 2004.

[54] Donatello Conte, Pasquale Foggia, Gennaro Percannella, Francesco Tufano, and Mario Vento. A method for counting people in crowded scenes. *In Proceedings of the IEEE International Conference on Advanced Video and Signal Based Surveillance*, pages 225–232, 2010.

[55] Di Kang, Zheng Ma, and Antoni B. Chan. Beyond counting: Comparisons of density maps for crowd analysis tasks—counting, detection, and tracking. *IEEE Transactions on Circuits and Systems for Video Technology*, 29(5):1408-1422, 2018.

[56] Anthony C. Davies, Jia Hong Yin, and Sergio A. Velastin. Crowd monitoring using image processing. *Electronics Communication Engineering Journal*, 7(1):37–47, 1995.

[57] Jia Hong Yin, Sergio A. Velastin, and Anthony C. Davies. Image processing techniques for crowd density estimation using a reference image. *In Asian conference on computer vision*, pages 489–498, 1996.

[58] Komal R. Ahuja and Nadir N. Charniya. A survey of recent advances in crowd density estimation using image processing. *In 2019 International Conference on Communication and Electronics Systems (ICCES)*, pages 1207–1213, 2019.

[59] Ruihua Ma, Liyuan Li, Weimin Huang, and Qi Tian. On pixel count based crowd density estimation for visual surveillance. *In IEEE Conference on Cybernetics and Intelligent Systems*, 1:170–173, 2004.

[60] Danny B. Yang, Hector H. Gonzalez-Banos, and Leonidas J. Guibas. Counting people in crowds with a real-time network of simple image sensors. *In ICCV*, 3:122, 2003.

[61] Aparecido Nilceu Marana, S. A. Velastin, L. F. Costa, and R. A. Lotufo. Estimation of crowd density using image processing. *IEE Colloquium on Image Processing for Security Applications*, pages 11–11, 1997.

[62] Aparecido Nilceu Marana, S. A. Velastin, L. F. Costa, and R. A. Lotufo. Automatic estimation of crowd density using texture. *Safety Science*, 28(3):165–175, 1998.

[63] Aparecido Nilceu Marana, Marcos Antonio Cavenaghi, Roberta Spolon Ulson, and F. L. Drumond. Real-time crowd density estimation using images. *In International Symposium on Visual Computing*, pages 355–362, 2005.

[64] Robert M. Haralick. Statistical and structural approaches to texture. *Proceedings of the IEEE*, 67(5):786–804, 1979.

[65] Teuvo Kohonen. The self-organizing map. *Proceedings of the IEEE*, 78(9):1464–1480, 1990.

[66] Li Xiaohua, Shen Lansun, and Li Huanqin. Estimation of crowd density based on wavelet and support vector machine. *Transactions of the Institute of Measurement and Control*, 28(3):299–308, 2006.

[67] Xinyu Wu, Guoyuan Liang, Ka Keung Lee, and Yangsheng Xu. Crowd density estimation using texture analysis and learning. *In Proceedings of the IEEE international conference on robotics and biomimetics*, pages 214–219, 2006.

[68] Zhaoxiang Zhang and Min Li. Crowd density estimation based on statistical analysis of local intra-crowd motions for public area surveillance. *Optical Engineering*, 51(4):047204, 2012.

[69] Antonio Albiol, Maria Julia Silla, Alberto Albiol, and Jose Manuel Mossi. Video analysis using corner motion statistics. *In IEEE International Workshop on Performance Evaluation of Tracking and Surveillance*, pages 31–38, 2009.

[70] Chris Harris and Mike Stephens. A combined corner and edge detector. *In Alvey vision conference*, 15(50):10-5244, 1988.

[71] F. Dittrich, A. L. Koerich, and L. E. S. Oliveira. People counting in crowded scenes using multiple cameras. *In 2012 19th International Conference on Systems, Signals and Image Processing (IWSSIP)*, pages 138–141, 2012.

[72] Donatello Conte, Pasquale Foggia, Gennaro Percannella, and Mario Vento. Counting moving persons in crowded scenes. *Machine vision and applications*, 24(5):1029–1042, 2013.

[73] Giovanni Acampora, Vincenzo Loia, Gennaro Percannella, and Mario Vento. Trainable estimators for indirect people counting: A comparative study. *In 2011 IEEE International Conference on Fuzzy Systems (FUZZ)*, pages 139–145, 2011.

[74] Hajer Fradi and Jean-Luc Dugelay. People counting system in crowded scenes based on feature regression. *In Proceedings of the 20th European Signal Processing Conference (EUSIPCO)*, pages 136–140, 2012.

[75] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[76] Ronghua Liang, Yuge Zhu, and Haixia Wang. Counting crowd flow based on feature points. *Neurocomputing*, 133:377–384, 2014.

[77] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. *In Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, 96(34):226–231, 1996.

[78] Bruce D. Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision. *In proceedings of the 7th International Joint Conference on Artificial Intelligence, IJCAI'81, Morgan Kaufmann Publishers Inc., San Francisco*, 2:674–679, 1981.

[79] Chuan Wang, Hua Zhang, Liang Yang, Si Liu, and Xiaochun Cao. Deep people counting in extremely dense crowds. *Proceedings of the 23rd ACM international conference on Multimedia*, pages 1299–1302, 2015.

[80] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, 25:1097-1105, 2012.

[81] Xinghao Ding, Zhirui Lin, Fujin He, Yu Wang, and Yue Huang. A deeply-recursive convolutional network for crowd counting. *In proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1942–1946, 2018.

[82] David Eigen, Jason Rolfe, Rob Fergus, and Yann LeCun. Understanding deep architectures using a recursive convolutional network. *Proceedings of the International Conference on Learning Representations*, 2013.

[83] Russell Stewart, Mykhaylo Andriluka, and Andrew Y. Ng. End-to-end people detection in crowded scenes. *In Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2325–2333, 2016.

[84] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[85] Wei Li, Hongliang Li, Qingbo Wu, Fanman Meng, Linfeng Xu, and King Ngi Ngan. Headnet: An end-to-end adaptive relational network for head detection. *IEEE Transactions on Circuits and Systems for Video Technology*, 30(2):482–494, 2019.

[86] Deepak Babu Sam, Skand Vishwanath Periand Mukuntha Narayanan Sundararaman, Amogh Kamath, and R. Venkatesh Babu. Locate, size, and count: accurately resolving people in dense crowds via detection. *IEEE transactions on pattern analysis and machine intelligence*, 43(8):2739–2751.

[87] Linxiang Zhu, Feifei Lee, Jiawei Cai, Hongliu Yu, and Qiu Chen. An improved feature pyramid network for object detection. *Neurocomputing*, 483:127–139, 2022.

[88] Josh Beal, Eric Kim, Eric Tzeng, Dong Huk Park, Andrew Zhai, and Dmitry Kislyuk. Toward transformer-based object detection. *arXiv preprint arXiv:2012.09958*, 2020.

[89] Zixiao Zhang, Xiaoqiang Lu, Guojin Cao, Yuting Yang, Licheng Jiao, and Fang Liu. Vit-yolo: Transformer-based yolo for object detection. *In Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2799–2808, 2021.

[90] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. *In Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021.

[91] Junhyung Kang, Shahroz Tariq, Han Oh, and Simon S. Woo. A survey of deep learning-based object detection methods and datasets for overhead imagery. *IEEE Access*, 10:20118–20134, 2022.

[92] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. *In Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

[93] Licheng Jiao, Fan Zhang, Fang Liu, Shuyuan Yang, Lingling Li, Zhixi Feng, and Rong Qu. A survey of deep learning-based object detection. *IEEE access*, 7:128837–128868, 2019.

[94] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, 37(9):1904–1916, 2015.

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

[96] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015.

[97] Jifeng Dai, Yi Li, Kaiming He, and Jian Sun. R-fcn: Object detection via region-based fully convolutional networks. *Advances in neural information processing systems*, 29, 2016.

[98] Xiang Li, Yuchen Jiang, Chenglin Liu, Shaochong Liu, Hao Luo, and Shen Yin. Playing against deep-neural-network-based object detectors: A novel bidirectional adversarial attack approach. *IEEE Transactions on Artificial Intelligence*, 3(1):20–28, 2021.

[99] M. F. Ansari and K. A. Lodi. A survey of recent trends in two-stage object detection methods. *Renewable Power for Sustainable Growth*, pages 669–677, 2021.

[100] Peng Sun, Guang Chen, and Yi Shang. Adaptive saliency biased loss for object detection in aerial images. *IEEE Transactions on Geoscience and Remote Sensing*, 58(10):7154–7165, 2020.

[101] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. *In Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[102] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. Ssd: Single shot multibox detector. *In European conference on computer vision*, pages 21–37, 2016.

[103] Cheng-Yang Fu, Wei Liu, Ananth Ranga, Ambrish Tyagi, and Alexander C. Berg. Dssd: Deconvolutional single shot detector. *arXiv preprint arXiv:1701.06659*, 2017.

[104] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. *In Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.

[105] Peiyuan Jiang, Daji Ergu, Fangyao Liu, Ying Cai, and Bo Ma. A review of yolo algorithm developments. *Procedia Computer Science*, 199:1066–1073, 2022.

[106] Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. *In Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7263–7271, 2017.

[107] Shifeng Zhang, Longyin Wen, Xiao Bian, Zhen Lei, and Stan Z. Li. Single-shot refinement neural network for object detection. *In Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4203–4212, 2018.

[108] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

[109] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, 2020.

[110] Hamzeh Mirhaji, Mohsen Soleymani, Abbas Asakereh, and Saman Abdanan Mehdizadeh. Fruit detection and load estimation of an orange orchard using the yolo models through simple approaches in different imaging and illumination conditions. *Computers and Electronics in Agriculture*, 191:106533, 2021.

[111] Glenn Jocher, Alex Stoken, Jirka Borovec, Liu Changyu, Adam Hogan, L. Diaconu, F. Ingham, J. Poznanski, J. Fang, and L. Yu. ultralytics/yolov5: v3.1. *Zenodo, 10.5281/zenodo.4154370*, 2020.

[112] Kun Han and Xiangdong Zeng. Deep learning-based workers safety helmet wearing detection on construction sites using multi-scale features. *IEEE Access*, 10:718–729, 2021.

[113] Yuhao Bai, Yunxiang Guo, Qian Zhang, Boyuan Cao, and Baohua Zhang. Multi-network fusion algorithm with transfer learning for green cucumber segmentation and recognition under complex natural environment. *Computers and Electronics in Agriculture*, 194:106789, 2022.

[114] Yuan Dai, Weiming Liu, Haiyu Li, and Lan Liu. Efficient foreign object detection between psds and metro doors via deep neural networks. *IEEE Access*, 8:46723–46734, 2020.

[115] Zongwang Lyu, Huifang Jin, Tong Zhen, Fuyan Sun, and Hui Xu. Small object recognition algorithm of grain pests based on ssd feature fusion. *IEEE Access*, 9:43202–43213, 2021.

[116] Zizhu Fan, Hong Zhang, Zheng Zhang, Guangming Lu, Yudong Zhang, and Yaowei Wang. A survey of crowd counting and density estimation based on convolutional neural network. *Neurocomputing*, 472:224–251, 2022.

[117] Elad Walach and Lior Wolf. Learning to count with cnn boosting. *In European conference on computer vision*, pages 660–676, 2016.

[118] Yingying Zhang, Desen Zhou, Siqin Chen, Shenghua Gao, and Yi Ma. Single-image crowd counting via multi-column convolutional neural network. *In Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 589–597, 2016.

[119] Biyun Sheng, Chunhua Shen, Guosheng Lin, Jun Li, Wankou Yang, and Changyin Sun. Crowd counting via weighted vlad on a dense attribute feature map. *IEEE Transactions on Circuits and Systems for Video Technology*, 28(8):1788-1797, 2016.

[120] Deepak Babu Sam and R. Venkatesh Babu. Top-down feedback for crowd counting convolutional neural network. *In Thirty-second AAAI conference on artificial intelligence*, pages 7323–7330, 2018.

[121] Lingbo Liu, Hongjun Wang, Guanbin Li, Wanli Ouyang, and Liang Lin. Crowd counting using deep recurrent spatial-aware network. *In Proceedings of the International Joint Conference on Artificial Intelligence*, pages 849–855, 2018.

[122] Mark Marsden, Kevin McGuinness, Suzanne Little, and Noel E. O'Connor. Resnetcrowd: A residual deep learning architecture for crowd counting, violent behaviour detection and crowd density level classification. *In Proceedings of the IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 1–7, 2017.

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

[124] Zhuoyi Zhao, Hongsheng Li, Rui Zhao, and Xiaogang Wang. Crossing-line crowd counting with two-phase deep neural networks. *In Proceedings of the European Conference on Computer Vision*, pages 712–726, 2016.

[125] Muming Zhao, Jian Zhang andChongyang Zhang, and Wenjun Zhang. Leveraging heterogeneous auxiliary tasks to assist crowd counting. *In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12736–12745, 2019.

[126] Chenchen Liu, Xinyu Weng, and Yadong Mu. Recurrent attentive zooming for joint crowd counting and precise localization. *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1217–1226, 2019.

[127] Greg Olmschenk, Jin Chen, Hao Tang, and Zhigang Zhu. Dense crowd counting convolutional neural networks with minimal data using semi-supervised dual-goal generative adversarial networks. *In IEEE Conference on Computer Vision and Pattern Recognition: Learning with Imperfect Data Workshop*, 2019.

[128] Deepak Babu Sam, Neeraj N. Sajjan, Himanshu Maurya, and R. Venkatesh Babu. Almost unsupervised learning for dense crowd counting. *In Proceedings of the AAAI Conference on Artificial Intelligence*, 13(1):8868–8875, 2019.

[129] Yinjie Lei, Yan Liu, Pingping Zhang, and Lingqiao Liu. Towards using count-level weak supervision for crowd counting. *Pattern Recognition*, 109:107616, 2021.

[130] Xialei Liu, Joost Van De Weijer, and Andrew D. Bagdanov. Leveraging unlabeled data for crowd counting by learning to rank. *In Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7661–7669, 2018.

[131] Qi Wang, Junyu Gao, Wei Lin, and Yuan Yuan. Learning from synthetic data for crowd counting in the wild. *In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8198–8207, 2019.

[132] Cong Zhang, Hongsheng Li, Xiaogang Wang, and Xiaokang Yang. Cross-scene crowd counting via deep convolutional neural networks. *In Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 833–841, 2015.

[133] Miaojing Shiand Zhaohui Yang, Chao Xu, and Qijun Chen. Revisiting perspective information for efficient crowd counting. *In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7279–7288, 2019.

[134] Zhaoyi Yan, Yuchen Yuan, Wangmeng Zuo, Xiao Tan, Yezhen Wang, Shilei Wen, and Errui Ding. Perspective-guided convolution networks for crowd counting. *In Proceedings of the IEEE/CVF international conference on computer vision*, pages 952–961, 2019.

[135] Vishwanath A. Sindagi and Vishal M. Patel. Ha-ccn: Hierarchical attention-based crowd counting network. *IEEE Transactions on Image Processing*, 29:323–335, 2019.

[136] Ning Liu, Yongchao Long, Changqing Zou, Qun Niu, Li Pan, and Hefeng Wu. Adcrowdnet: An attention-injective deformable convolutional network for crowd understanding. *In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3225–3234, 2019.

[137] Yutao Hu, Xiaolong Jiang, Xuhui Liu, Baochang Zhang, Jungong Han, Xianbin Cao, and David Doermann. Nas-count: Counting-by-density with neural architecture search. *In European Conference on Computer Vision*, pages 747–766, 2020.

[138] Jiang Liu, Chenqiang Gao, Deyu Meng, and Alexander G. Hauptmann. Decidenet: Counting varying density crowds through attention guided detection and density estimation. *In Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5197–5206, 2018.

[139] Wenchao Gu, Shuang Bai, and Lingxing Kong. A review on 2d instance segmentation based on deep neural networks. *Image and Vision Computing*, 104401, 2022.

[140] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. *In European conference on computer vision*, pages 740–755, 2014.

[141] Vladimir Vapnik, Steven Golowich, and Alex Smola. Support vector method for function approximation, regression estimation and signal processing. *Advances in neural information processing systems*, 9, 1996.

[142] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. The elements of statistical learning: data mining, inference, and prediction. *Springer New York, NY*, 2, 2009.

[143] Debasish Basak, Srimanta Pal, and Dipak Chandra Patranabis. Support vector regression. *Neural Information Processing–Letters and Reviews*, 11(10):203–224, 2007.

[144] Abhishek Dutta and Andrew Zisserman. The via annotation software for images, audio and video. *Proceedings of the 27th ACM International Conference on Multimedia*, pages 2276–2279, 2019.