UNIVERSITY OF SOUTHAMPTON

Improving Object Detection Performance By Lightweight Approaches

by

Yingwei Zhou

A thesis submitted in partial fulfillment for the degree of Master of Philosophy

in the
Faculty of Engineering and Physical Sciences
School of Electronics and Computer Science

January 15, 2020

UNIVERSITY OF SOUTHAMPTON

ABSTRACT

Faculty of Engineering and Physical Sciences
SCHOOL OF ELECTRONICS AND COMPUTER SCIENCE

Master of Philosophy

by Yingwei Zhou

Object Detection has been a significant topic in computer vision. As the continuous development of Deep Learning, many advanced academic and industrial outcomes are established on localising and classifying the target objects, such as instance segmentation, video tracking and robotic vision. As the core concept of Deep Learning, Deep Neural Networks (DNNs) and associated training are highly integrated with task-driven modelling, having great effects on accurate detection. The main focus of improving detection performance is proposing DNNs with extra layers and novel topological connections to extract the desired features from input data. However, training these models can be a computational expensive and laborious progress as the complicated model architecture and enormous parameters. Besides, dataset is another reason causing this issue and low detection accuracy, because of insufficient data samples or difficult instances. To address these training difficulties, this thesis presents two different approaches to improve the detection performance in the relatively light-weight way. As the intrinsic feature of data-driven in deep learning, the first approach is "slot-based image augmentation" to enrich the dataset with extra foreground and background combinations. Instead of the commonly used image flipping method, the proposed system achieved similar mAP improvement with less extra images which decrease training time. This proposed augmentation system has extra flexibility adapting to various scenarios and the performance-driven analysis provides an alternative aspect of conducting image augmentation. The "StomaRCNN" is the second approach which is based on a realistic application task to automatically detect, segment and measure the stomata in plant microscope images. The key innovation of StomaRCNN is reorganising DNN pipeline to utilise the detailed features in high-resolution microscope images without damaging the image qualities. Despite the limited related works, StomaRCNN achieved human-level measurement accuracy for open stoma instances and demonstrates a large potential of applying Deep Learning approaches to automatically solve instance measuring problems in plant science. Those presented works propose alternative ways of improving object detection performance and highlight the importance of rethinking object detection in the aspects of data-driven and step-wise architectural design.

Contents

List of Figures

List of Tables

Listings

Nomenclature

w	The weight vector.
$\subseteq$	A subset of.
DL	Deep Learning.
DNN	Deep Neural Networks.
CNN	Convolutional Neural Networks.
AP	Average Precision.
AR	Average Recall.
mAP	mean Average Precision.
mAR	mean Average Recall.

Acknowledgements

Firstly I would like to thank my supervisor Professor Adam Prugel-Bennett for giving me many inspiring suggestions during my whole study and research progress. I am grateful for my parents and sister Hailin Zhou, to help me get through the tough times when I felt lost for my future. My parents and sister Hailin are not only fully supporting my study, but also teaching me the way of love and meaning of a family. I would also like to say my sincere thanks to my friends: Miss Xuefei Wen, Dr Henry W. J. Reeve, Mr Yang Ma, Dr Yiyang Chen and Miss Na Sai. It is my great honour to have you guys in my life and it is always a big pleasure to hang out, chatting and coming up funny memes. More importantly, I can always feel refreshed from unsuccessful experiments, paper rejection or other common college difficulties. Especially I want to thank my good friend and collaborator Miss Sai for spending days and nights helping with proofreading and logical corrections. I am grateful for her patience and significant contribution to our StomaRCNN works.

My past few years are filled with attempts, failures and bouncing back. There are several times I felt over-stressed and desperate for my future. It is these people who cheered me up, highlighted my drawbacks and took me to a further understanding of both my work and life. All these things cannot happen without them and I feel so lucky to have them in my life.

In the end, I want to thank Mr and Mrs Reeve for inviting me to the New Year Eve parties these years and I do enjoy it a lot. Professor Sun-Yuan Kung from Princeton gives me many insightful suggestions on research and life living when I was in a difficult status. I also want to thank the swimming pool and shower facilities in Jubilee Sports Hall for keeping fitness and saving my water fee. The last but not least, there are so many others to appreciate including the catering ladies in the campus restaurant and my peaceful studio flat. Without these people and these experiences, I cannot make it so far, much much appreciated.

To my dear uncle Dr Pi Li, who introduced me to the palace of machine learning . . .

Chapter 1

Introduction

Object detection is a fundamental research area to answer where are the objects and what they are. It consists of two key sub-problems of localisation and classification. Despite its simple conception, many advanced computer vision tasks are based on precisely acquired object location and categories by object detection methods, such as instance segmentation, video tracking and autonomous driving (Chen et al., 2015a; Lee et al., 2018; Schulter et al., 2017). Therefore detection quality has a great impact on these advanced vision tasks. The main contributions of this thesis are introducing two novel aspects to improve the detection performance which are light-weight, flexible to extend and less computational cost. To be specific, an image augmentation system called "slot-based image augmentation" is proposed and experiments have shown its improvements for small object detection and huge potential capacity in augmenting images. Moreover, it is implemented in an image pre-processing fashion, no need for training and easy to extend for various scenarios. This approach highlights the significance of data-source, addressing the very intrinsic of data-driven machine learning approaches. Besides, this work has found the failure of commonly used image augmentation methods that they might damage the mAP of detection DNNs. Details of this system are presented in Chapter 2.

Besides the aspect of data augmentation, another novel aspect is revealed that pipeline re-organisation is effective and important. Addressing a realistic biological scenario, we propose StomaRCNN to automatically detect, segment and measure the small scale stomatal pores. This scenarios limits available data amounts which are quite different from experimental cases and solutions on this scenario are required with high flexibility, less computational cost and more importantly acceptable precision on small scale object measurements. So the scenario has many commonalities with the thesis goals and the proposed StomaRCNN system not only solved these issues but also provides a novel way to fulfil the detection task utilising the images with much less information loss. Furthermore, StomaRCNN works split the commonly used instance segmentation pipeline by completing detection and segmentation with different individual models, instead of using the same one. The split allows using original images instead of sub-sampled one that has degraded resolution and especially harmful for detecting small objects. Additionally, StomaRCNN is organised with an end-to-end architecture enabling the “plugin plug-out” model replacement allowing a wide range of applying transfer learning. More details of StomaRCNN are demonstrated in Chapter 3.

Overall, our proposed methods have achieved important progress towards the research aims and more details can be found in their respective chapters. To evaluate the detection performances, mean average precision (mAP) and mean average recall (mAR) are widely used as the metrics determining the leader board of object detection approaches. mAP and mAR are the principle metrics in this thesis for system evaluations.

Chapter 2

Slot Based Image Augmentation

2.1 Object Detection and Evaluation Metrics

2.1.1 Evaluation Metrics: mAP and mAR

As mentioned above, one of the project goals is to find potential approaches to improve detection performance, in particular, increasing mAP and mAR. Despite the conceptional similarities of precision and recall in other machine learning tasks such as image classification, mAP and mAR are particularly customised for describing object detection performance. mAP is the mean of AP values with different insertion over union (IoU) thresholds.

The IoU threshold determines whether a proposed rectangle is correctly matched with the benchmark. The proposal IoU is calculated from the proposal rectangle and benchmark by

$$IoU = \frac{area\ of\ overlap}{area\ of\ union}. \tag{2.1}$$

The proposed rectangles are recognised as correct prediction only if its IoU is larger than IoU threshold. Furthermore, high IoU threshold leads the model only considers precisely localised predictions for inference or training. In addition, IoU threshold effects recall values as well and a smaller IoU threshold makes the system accepts more predictions contributing to larger recall. A well-trained model is expected with high AP at larger IoU thresholds and not varying too much under different thresholds. Besides the "mean" calculation in mAP/mAR, the average precision/recall (AP/AR) is defined based on the general conception of precision/recall. AP is the average precision over different recall level and AR is the average value of different precision level. Regarding the machine learning confusion matrix in Table 2.1, True Positive (TP) and True Negative (TN) are the correct predictions.

TABLE 2.1: Confusion Matrix in Machine Learning

Benchmark Prediction	True	False
Positive	True Positive (TP)	False Positive (FP)
Negative	False Negative (FN)	True Negative (TN)

Precision measures how accurate the predictions are and given by

$$Precision = \frac{TP}{TP + FP}, \tag{2.2}$$

and disregards the recall rates calculated by

$$Recall = \frac{TP}{TP + FN}. \tag{2.3}$$

As the limitation of using Precision or Recall alone, the conception of Average Precision is proposed to consider model precision under different recall values. To be specific, AP is averaged over several interpolated recalls and was originally proposed in PASCAL VOC challenges by Everingham et al. (2010), which is adapted from a similar idea called *"11-point interpolated average precision"* in document retrieval tasks by Salton and McGill (1986). The AP in MS-COCO metrics is calculated based on original formula in Salton and McGill (1986), as

$$P_{interp}(r) = max(P(r')), \tag{2.4}$$

where r is the recall interpolated value and r' is the recall to maximise P with $r' \geqslant r$. In practice, the equation is implemented as

$$AP = \frac{1}{11} \sum P(r), \tag{2.5}$$

and

$$P(r) = max(P(r')), r' \geqslant r \tag{2.6}$$

with $r \in \{0, 0.1, ..., 1.0\}$ Lin et al. (2014). AR follows similar ideas as mAP with focusing on recall values instead of precision.

AP/AR are the metrics summarising the detection P-R curve as an averaged value which is more straightforward to compare different detection systems. Overall, mAP evaluates model with considering both different recall level and IoU threshold, making it the most commonly used metrics in evaluating model performance and detection based competitions. In MS-COCO metrics, object scales are also considered as mAP calculation and four types mAPs are applied in this thesis: mAP for small objects, medium objects, large objects and overall (averaged across different scales).

2.1.2 Detailed Performance Analysis: "Derek style" P-R curves

Besides the mAP metric, the detailed evaluation of detection performance is required under some scenarios. Hoiem et al. (2012) summarised a series of typical errors and presented in a modified Precision and Recall (P-R) curve called "Derek style P-R curve". Figure 2.3 provides an example of this curve for ResNet-50 on MS-COCO dataset. In Hoiem et al. (2012), four types of errors are summarised: Loc, Sim, Oth, Bg and these errors are reasoned as objects presented in abnormal ways. To be specific, **Loc** stands for poor localisation or insufficient IoU with benchmarks. **Sim** is confusion with similar objects that belong to the same super-category like dog and cat from the animal super-category. **Oth** denotes the confusion with other objects. **Bg** represents confusion with background. In this thesis, model performance and object conditions are analysed in a similar way.

The Derek P-R curve listed detailed model performances and these types of errors as shown in Figure 2.3. Regarding the presented figure, model P-R curves are evaluated with two IoU thresholds: 0.75 for strict one and 0.50 for the easier one as the PASCAL VOC challenge Everingham et al. (2010). Both **C75** and **C50** demonstrates model capability to correct detect objects and marked as white area. Ideally better model has a larger **C75** and **C50** area. In addition, the relative complementary area of **C75** in **C50** are those detects with IoU value in $[0.50, 0.75]$. Four the P-R curves and error areas, **Loc** P-R is acquired after losing IoU threshold from 0.50 to 0.10 basically accepting most of the detection that correctly predicts object category. Hence the relative complementary area of **Loc** and **C50** are those detection results in IoU range from 0.10 to 0.50. **Sim** P-R curve is drawn after accepting incorrect classification within the same super-categories (False Positive within the same super-category) and related results are denoted in red. Despite the removed false positive detection within the same super-category, there are incorrect classifications from different super-categories and accepting those errors delivers **Oth** P-R curve. Incorrect results are presented as the green area. **Oth** P-R curve has analysed classification errors among different categories and super-categories. Beyond **Oth**, many detection results incorrectly recognise sub-area of background as the foreground object images. These incorrect results are denoted as **BG** errors and represented as purple area. **Fn** stands for the rest unspecified false-negative detection results.

Reasons for these errors can be model drawbacks, pipeline design and the abnormal object conditions. As most of the detection models are data-driven, abnormal object conditions are considered as the significant reason causing these errors. Due to a large

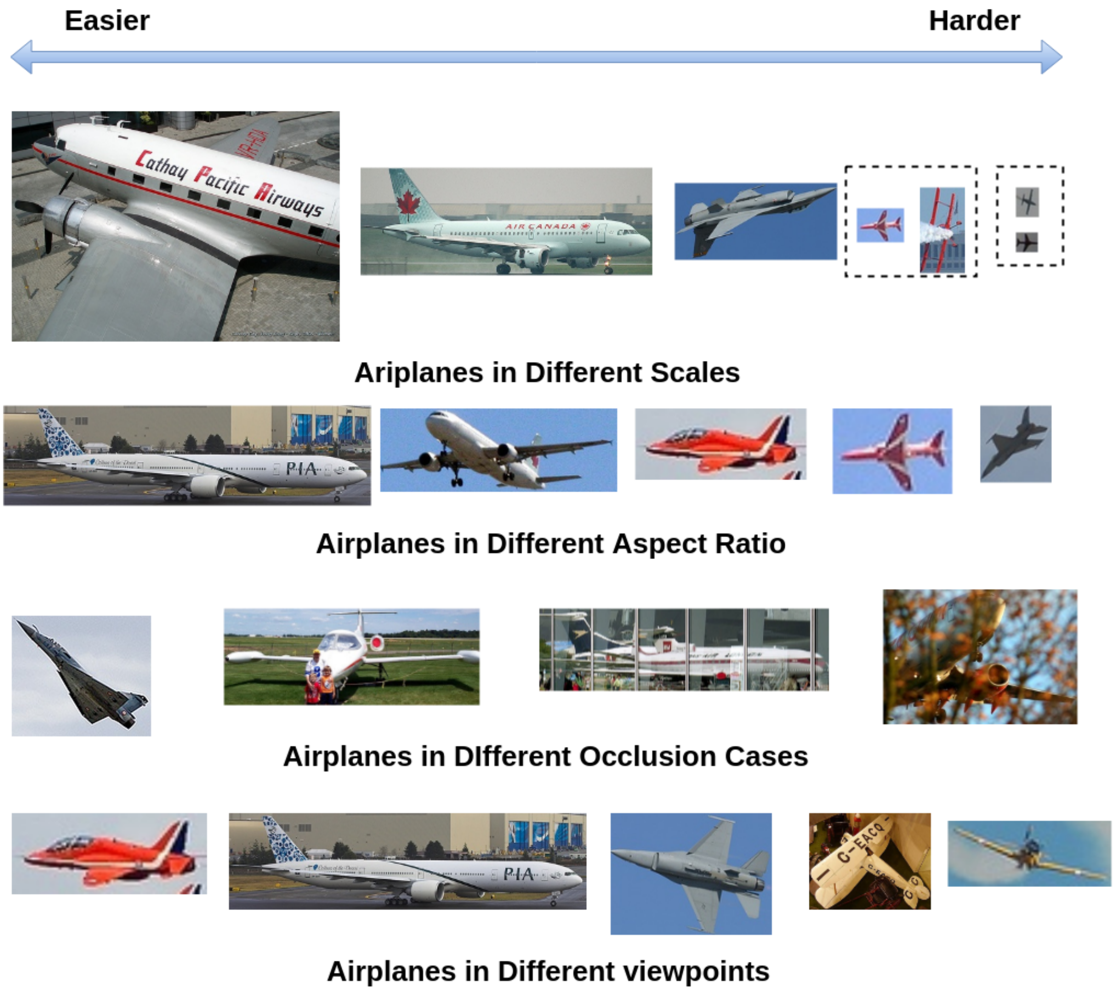
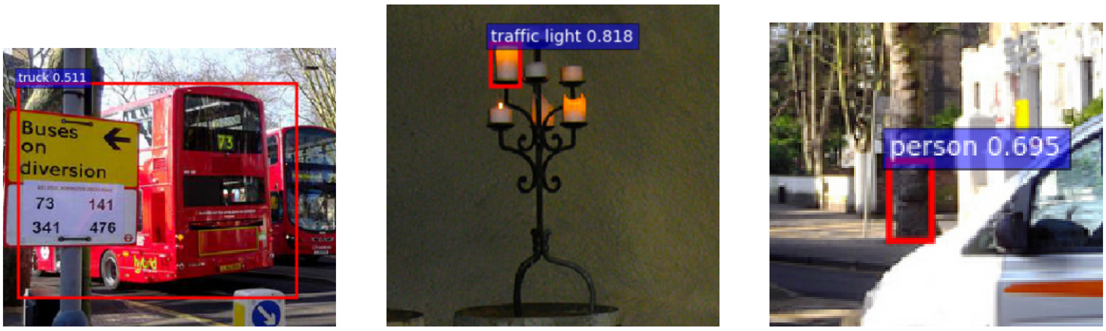

Easier
Harder
Ariplanes in Different Scales
Airplanes in Different Aspect Ratio
Airplanes in DIfferent Occlusion Cases
Airplanes in Different viewpoints

FIGURE 2.1: Difficulties of detecting an object in different conditions summarized from Hoiem et al. (2012). According to the detection mAP, objects at the left side is denoted as "easy" objects while hard ones are on the other side.

number of objects and various categories, it is difficult to group objects into independent and meaningful conditions. Guided by the model performance, Hoiem et al. (2012) has summarised abnormal object conditions as occlusion, truncation, small size and unusual viewpoint. Figure 2.1 gives examples of difficult conditions of airplane objects. These difficult conditions usually cause incorrect detection results as shown in Figure 2.2. In general, object conditions with less small occurrence frequencies are difficult to detect. Moreover, researchers propose different DNN models to improve overall performance or focusing on various scenarios. As computer vision is a task-driven domain, there are not many generalised approaches, especially lightweight and less computation cost methodologies, and the Derek P-R curve illustrates model weakness on specific error types. Besides the idea of error based reinforcement, solutions can be implemented in a pre-processing fashion such as our proposed slot-based image augmentation system, which is lightweight, no need for training and flexible to diverse scenarios.

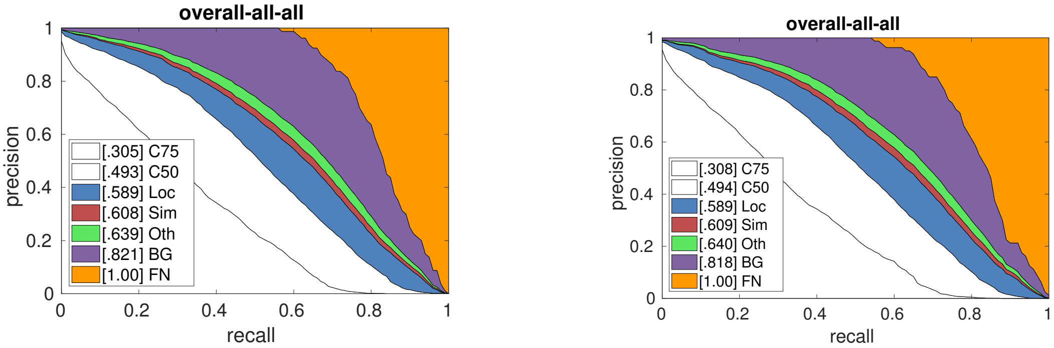


(a) Sim error

(b) Oth error

(c) Bg error

Figure 2.2: Examples of baseline model detection for common incorrect detection cases and the values stand for the prediction confidence. The left demo is a **Sim** error which confused a bus with a truck and they all belong to vehicle super-category. The middle one is a **Oth** case since it detects a candle as traffic lights from a different super-category. The right one is a **Bg** instance which treats part of a background as person.



(a) The performance of our benchmark model (“Faster RCNN with ResNet50 on MS-COCO dataset) without any image augmentation.

(b) Same as Ren et al. (2015a), the performance of our benchmark model with image flipping augmentation.

Figure 2.3: Our benchmark model performance (“Faster RCNN with ResNet-50) on MS-COCO dataset.

2.1.3 Project Motivation and Proposed Approaches

In summary, MS-COCO mAP measures overall performances regarding different scales and IoU thresholds. The P-R curve reflects model robustness with different detection errors. In this thesis, approaches are proposed to improve the performance guided by the two metrics, which provide systematic evaluations. To be specific, the model performance of small object detection is evaluated by the mAP for small scales and Derek P-R curve demonstrates weakness and potential error types to overcome.

Regarding the motivation of this project, researchers have proposed several approaches addressing various scenarios. Most of these related works conduct architectural modification including feature map fusion, concatenation and stacking extra layers. These approaches have achieved the desired outcome to some extent. However, these modifications usually lead to computational expensive and time-costing. Somehow, the customised architectures are tightly associated with certain scenarios and difficult to be transferred to other conditions, compared with those general systems such as "Faster RCNN. Therefore, we expect lightweight, flexible and easy-to-extend approaches to improve the performance, which can be widely deployed in many hardware cases and capable to transfer into different scenarios. In addition, models are evaluated metrics of mAP and Derek P-R curve.

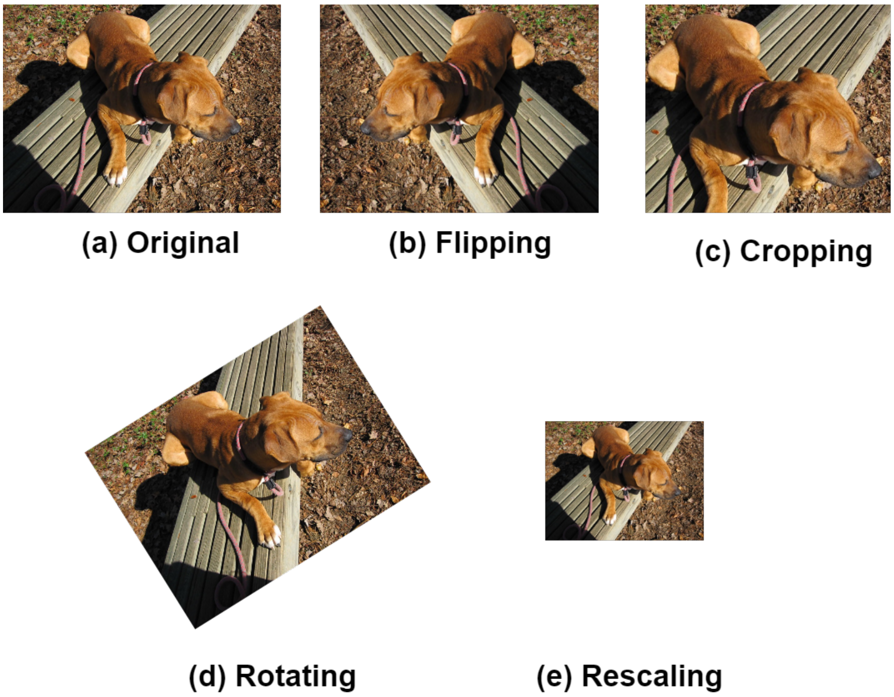
2.2 Research Gap and Motivation

As mentioned in Chapter 1, object detection is the fundamental step for many vision tasks and the detection performance improvement is significantly important. To specify the detection task, the metrics including mAP and Derek P-R curve evaluate the detection system in different aspects. There are many potential ways for improvements. Therefore, potential approaches are able to be found for improvements in detection performance.

In practice, the limitation of available data is a common case and data augmentation methods such as flipping, rotating and tuning brightness, are applied to get through the bottleneck of lacking training data. These methods are conducted as manual image transformation without task-oriented features. Although these methods contribute partial increase for image classification accuracy, they are not that beneficial for object detection. In some cases, augmentation makes the model performance worse and related results are presented in this Chapter. The main reason for the failures is the differences in target learn-able features between image classification and object detection. In addition, object detection requires extra location and contextual features rather than the object feature alone as in classification task. Moreover, the manual image transformation based augmentation is not guaranteed to augment sufficient extra learn-able features, especially for small scale objects.

Therefore, the above discussion draws the motivation that an image augmentation approach is desired especially for object detection task to provide extra classification and localisation features. Regarding the methodological gap between object detection and regular augmentation methods, the slot-based image augmentation is proposed to generate images with more learn-able object detection related features by adding extra combinations of foreground objects and background images.

2.3 Related Works



(a) Original

(b) Flipping

(c) Cropping

(d) Rotating

(e) Rescaling

FIGURE 2.4: **The Examples of Commonly Used Image Augmentation Methods**.

Image augmentation is the new rising stream of improving detection performance. [1] There are two mainstreams of image augmentation: simple image transformation, and adding objects into images by machine-learning based object allocation. Image transformation is the commonly used image augmentation approaches such as flipping, rotating, cropping and re-scaling as shown in Figure 2.4. Because images are represented as matrix in computer vision, these transformations are conducted by linear transformation on matrix and many toolboxes support these transformations such as OpenCV Bradski (2000), Scikit-image Van der Walt et al. (2014) and recently released "Population-Based Augmentation" by Ho et al. (2019). Besides of these listed transformations on image size or positions, another alternative approach is the colour-wise augmentation including brightness, contrast and saturation, which are supported by many deep learning frameworks: TorchVision Paszke et al. (2017), mxNet Chen et al. (2015b), etc. In summary, most of the augmentation works are based on simple image transformation and there are not many other augmentation approaches featured with performance analysis or adaptive augmentation.

Besides these aforementioned augmentation methods, there are some works on inserting objects into background image to generate new images. Dvornik et al. (2018) proposed

[1] This project started from April 2018 when there are not many related on image augmentation especially for object detection. However, as the fast development in this subarea, many related works have been published since the end of 2018.

a machine learning based method to search suitable positions to insert the objects. These positions are ranked by the performance evaluation of the applied object detection models. Among these works, Singh and Davis (2018) highlights the importance of scales. Kisantal et al. (2019) addressed the problem of imbalanced objects in the scale-wise. They augmented extra images by copying and pasting the selected objects into the backgrounds with the detection performance gained. Instead of conducting general augmentation for objects of all categories in the dataset, some of those focus on augmentation of specific categories, such as Kim et al. (2018) proposed a DNN to insert objects and with augmented backgrounds. Despite these attempts on different augmentation approaches, some researchers worked on the strategy of image augmentation. Ratner et al. (2017) proposed a GANs-based pipeline to generate images with domain transformation on the small CIFAR-10 dataset. Zoph et al. (2019) investigated different strategies of image augmentation mainly on MS-COCO dataset. Compared with colour, geometric, and bounding box based approaches, they proposed an improved auto-augmentation method from the previous "AutoAugment Policies" by Cubuk et al. (2018).

Regarding many of these augmentation approaches focus on the objects, it is important to highlight the loss functions on the data balancing problem in the object-wise. One of the most popular is the "Focal-Loss" used in RetinaNet proposed by Lin et al. (2017b). The key idea is weighting the multi-class cross-entropy format loss functions, by decreasing the loss for the frequently occurred categories. Therefore, it helps in reducing overfitting and improved model performance. There are many alternative implementation and extension works on Focal Loss, such as Focal Loss for segmentation by Hossain et al. (2019).

Instead of these augmentations, the mainstream focuses on modifying DNN architecture to improve detection performance by mixing CNN feature maps of different layers. For the small object detection, the contextual information is highlighted thus various DNN architectures are proposed to utilise these features as possible such as original feature pyramid pooling by He et al. (2015), the relation-based DNN architecture by Hu et al. (2018) and contextual refinement methods by Chen et al. (2018).

These architectural modifications methods achieved desired detection improvements however, added extra computation costs and less flexible for transferring the trained models to different scenarios. Inspired by these proposed novel methods and impressive implementation works, the proposed slot-based image augmentation method is lightweight, flexible for transfer learning and highlights a general way of augmenting images for object detection.

2.4 Slot-Based Image Augmentation

2.4.1 System Introduction

The slot-based augmentation system is built on the fundamental element called “slot” which is a generalised conception from the isolated foreground objects. In other words, a slot is a replaceable position which is initially the isolated foreground object in an image and the slot-based image augmentation produces extra images by substituting foregrounds enriching various combinations of foreground and backgrounds. Furthermore detecting objects is the process of recognising foreground objects from its background utilisation the features learnt from training images. Therefore the slot substitution enriches the combinations especially for the unbalanced dataset in which there are not sufficient objects of a certain category. In addition, the slot based augmentation is easy to customise and highly flexible for different scenarios such as substituting objects of the same category, which creates extra learn-able “foreground and background” correlation features for the target category objects. Figure 2.5 example of augmentation. It is also a potential way to change the bounding box style slot substitution to an instance wise substitution as the instances for segmentation tasks are represented as polygons.

2.5 System Implementation

Regarding the conception of slots, the two highlighted features are foregrounds and isolation. On one hand, foregrounds assign slot location as the foregrounds which tightly include objects inside the bounding box and causes limited corruption for other objects or background while substituting them. On the other hand, the isolation selects individual objects that have no insertion with any other objects and no damage to other objects when replacing them with other ones. Hence a slot is a rectangle area of the background image with the same position as an isolated original object and causes minor effects while substituting it.

Despite the simple definition, it takes several steps to conduct the augmentation and the system is designed in an end-to-end fashion making it flexible to modify and extend with more functionalities, which are presented in the following section.

2.5.1 System Design and Pipeline

As introduced above, a slot is the central component of the augmentation system which is required to be isolated and possessing the foreground object locations. Dataset for

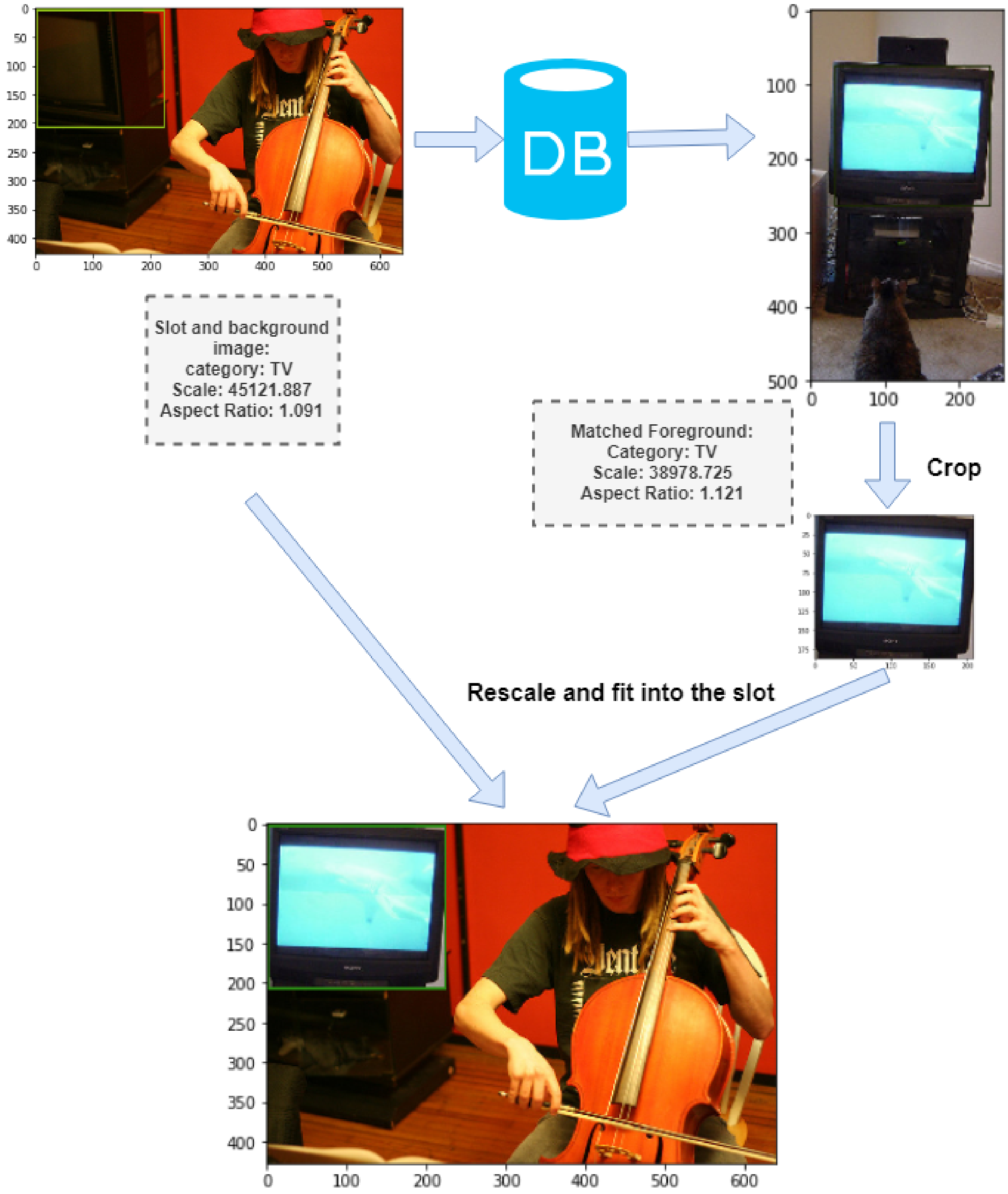


FIGURE 2.5: Generating one image by fitting one TV slot with a valid foreground object which has a similar aspect ratio and scales.

object detection includes images and annotation files that specify object locations as coordinates and object information such as category and scales. Thus foreground object locations are capable to be directly extracted from the annotation files while isolation requires extra steps to extract them. The progress of selecting isolated objects is simplified as finding individual rectangle bounding boxes that have no insertion with any others in the same image. As bounding boxes are represented with coordinates, a heuristic method is applied to solve this problem by comparing a set of coordinates as in Figure

and the rules are shown below:
For two bounding boxes represented by the coordinates of the top-left and bottom-right vertexes as $bbox1$: $[x_{11},\ y_{11},\ x_{12},\ y_{12}]$ and $bbox2$: $[x_{21},\ y_{21},\ x_{22},\ y_{22}]$, where $(x_{11},\ y_{11})$ and $x_{21},\ y_{21}$ are top-left vertices while the bottom-right ones are $x_{12},\ y_{12}$ and $x_{22},\ y_{22}$. So $bbox1$ and $bbox2$ are overlapped if and only if satisfying all the following conditions and the isolated slots are decided by the complementary cases:

(C1): $x_{11} < x_{22}$

(C2): $x_{12} > x_{21}$

(C3): $y_{11} < y_{22}$

(C4): $y_{12} > y_{21}$

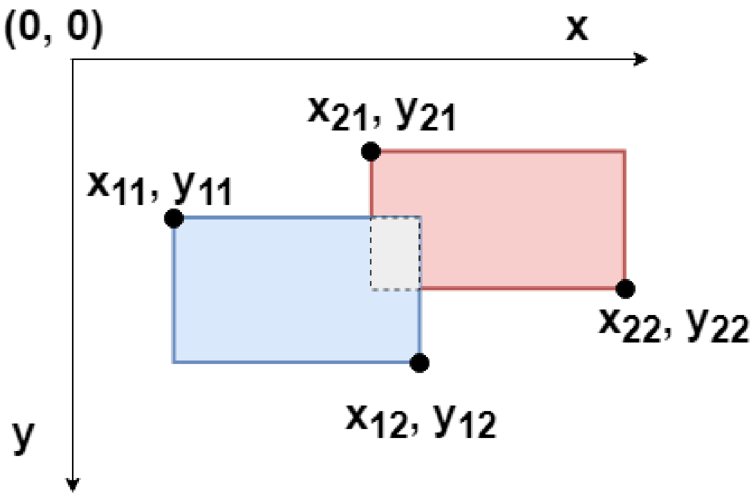


Figure 2.6: Slots are the isolated foregrounds represented as rectangle bounding boxes. Finding isolated bounding boxes can be achieved by analysing coordinates according to a series rules.

The system pipeline contains two main phases: initialisation and image augmentation as shown in Figure 2.7. To be specific, the valid slots selected by matching the complementary conditions proposed above. These picked slots are recorded into a database with detailed information include slot location, original foreground category and other scale related values (width, height, area size and aspect ratio). Generating images is conducted by substituting slots satisfying specific schemes determined by certain scenarios and scenarios are determined based on the model performance evaluated by mAP and Derek P-R curve. For example, if the model behaves low mAP at a certain category, the augmentation is set to produce images containing objects of the target category, Figure 2.5 gives a detailed workflow of augmenting extra images containing objects of the "television" category. Many other scenario-based augmentations are presented in later sections of this chapter.

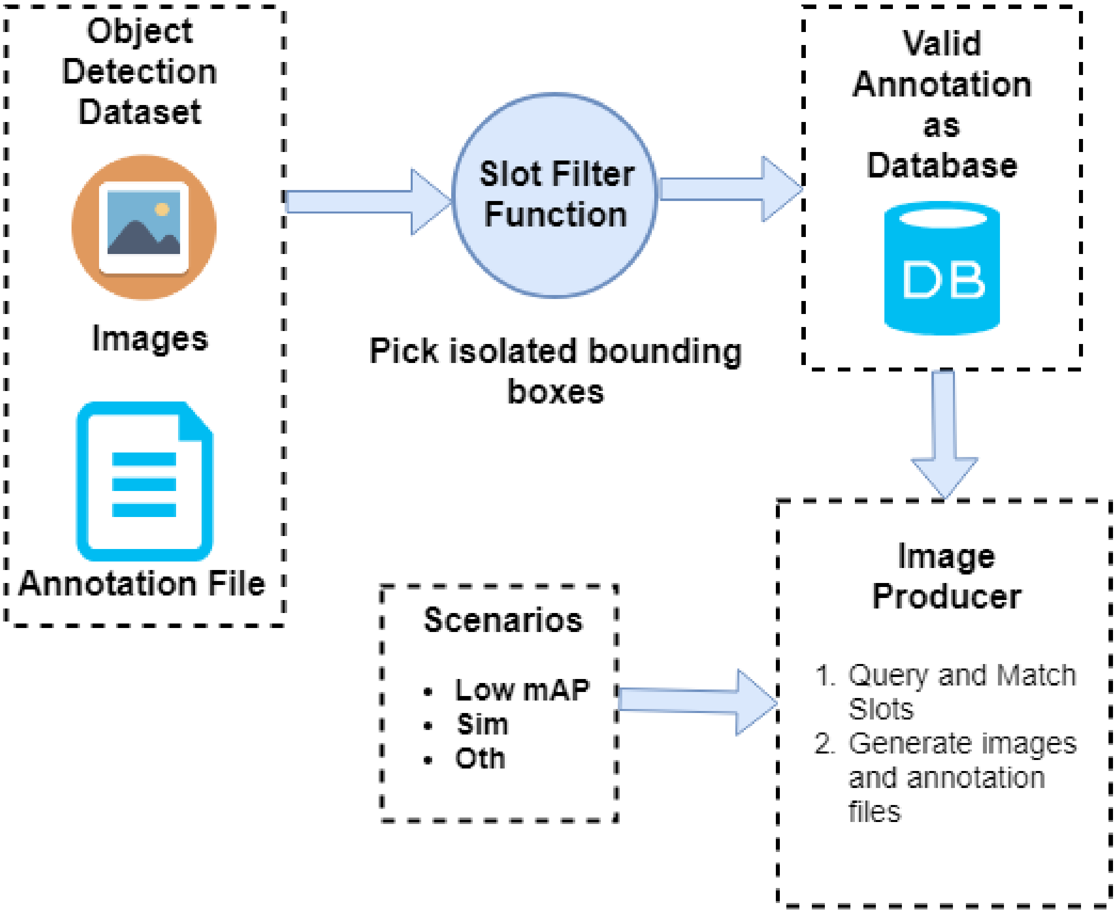


FIGURE 2.7: Components of the Slot-based Image Augmentation System.

In the reproduction and implementation aspect, the system consists of several central steps such as system initialisation, performance analysis of detection model and image augmentation based on the decided scenario. In experiments, MS-COCO 2014 dataset is selected as the main dataset as it provides helpful API codes and contains a large number of objects especially small objects Lin et al. (2014). MS-COCO 2014 is split as the full training set, a subset of validation called “val minus minival” and the reduced validation dataset called “minival”. The training set and randomly selected 5k “val minus minival” are used for traning while “minival” is used for validation. Furthermore, MS-COCO is a commonly used dataset in many related works as the reference of model capacity. For detection model selection, the PyTorch (Paszke et al. (2017)) version “Faster RCNN is selected as the benchmark detection model implemented by Chen and Gupta (2017) which uses ImageNet pre-trained ResNet-50 as its backend proposed by (Deng et al., 2009; He et al., 2016). “Faster RCNN is a widely used model to compare with related works or transfer pre-trained weights and sufficiently supported by the deep learning community.

As in the pipeline Figure 2.7, system initialisation follows loading dataset into the system, select isolated slots and summarise them into a database. In practice, dataset loading is implemented based on customised MS-COCO API to read annotation (JavaScript), load training images and refine the data structure. The associated Python libraries include Numpy, Scikit-Image, Matplotlib and many others (Van Der Walt et al., 2011; Van der Walt et al., 2014; Hunter, 2007). The isolated slots selection is conducted by implementing the condition matching mentioned above, mainly by the Numpy library. The database is implemented using the data structure of Python list and dictionary encapsulated by PANDAS library (McKinney (2011)), which is feasible for matching SQL (Structured Query Language) style enquiry in a fast and efficient way especially handling a large number of data records. In addition, the slot aspect ratio, area size and other information are calculated using Numpy and Jupyter Notebook is the principal tool for efficient coding.

To analyse model performance with mAP, MS-COCO API provides evaluation tools in Python, Matlab and other languages. In practice, Python version official evaluation tools are integrated into the implementation of the " "Faster RCNN benchmark model. Since Derek P-R curve plotting is not supported by Python API, the Matlab version is used for the curve drawing and extra Linux shell scripts are implemented for the intermediate data structural transformation between Python and Matlab API.

Finally, for the slot substitutions, the slot based sub-image cropping and pasting are conducted by python libraries of OpenCV and Scikit-Image (Bradski, 2000; Van der Walt et al., 2014). Besides the crop and paste step, there are normally multiple available candidates for a certain slot and the rules of candidate selection is another important problem to concern.

2.6 Filtering Slot Candidates

As mentioned in previous sections, slot-based image augmentation is conducted by substituting foreground objects and re-scale the candidate foreground to fit the slot shape. It is a common case that many candidate foregrounds are suitable for the target slot. Hence a candidate selection rule is required to address the "many-to-one" issue. In this section, three different filters are introduced to ensure the augmentation quality with analysis and experiments presented.

In practice, an "attribute matching" strategy is proposed to select valid candidates for the substitution with less damage to the potential learn-able features of the background image and candidate. Regarding the instance information in a large dataset, instances have a wide range of scale related attributes such as area size, length, width and height. Among those attributes, the instance area size, aspect ratio and category are applied as the filters to select candidates for slot substitution, in which the area size and aspect ratio are calculated in the format of rectangle bounding boxes.

As the example in Figure 2.8, candidates are firstly filtered by their aspect ratio to avoid invalid re-scaling. Then candidates with similar scales/resolution are selected by the scale filter. The final step is filter rest of these candidates by their categories. The category filter is a scenario-driven component. For example, candidates of the same category as the original slot foreground are selected when the augmentation system is aimed at augmenting extra images for a certain category. Another scenario is augmenting images to improve the detection performance of one certain super-category instances

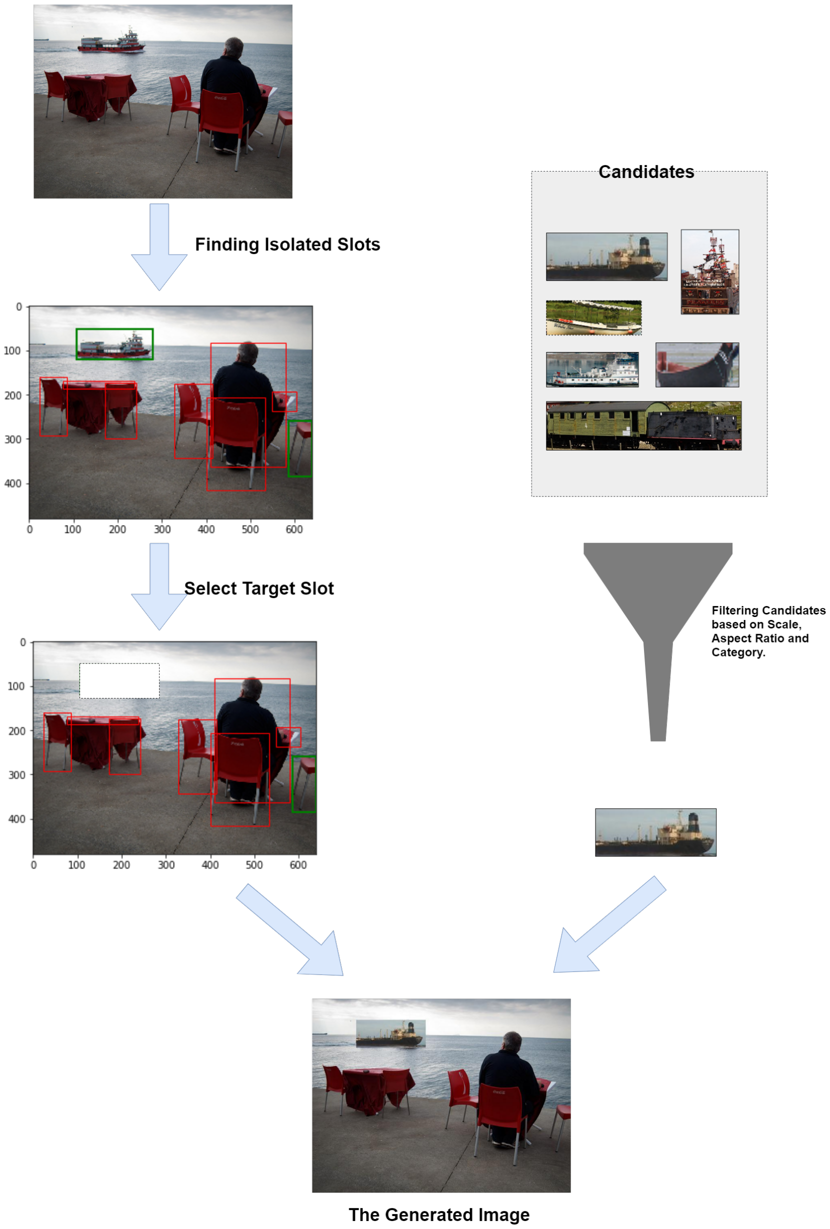


Figure 2.8: System Work-flow

such as substitute cats with dogs under animal super-category. Overall, filters are decided to preserve as much object information as possible without corrupting background images. These three filters are simple, effective and easy for modification. Candidates selected by the three filters are randomly chosen as the final candidate to fit the target slot.

Furthermore, several experiments are conducted to test the validity of selecting these filters and a loss function based candidate matching policy is introduced to compare with the random selection method for the final candidate decision. To transfer the scenario in a more realistic level, a mini-COCO dataset is established for filter validation tests, presented in the following section.

2.6.1 Filter Validation on "mini-COCO" dataset

Table 2.2: Top and Bottom Three Categories of Instance Amount in MS-COCO

Categories	Image Amount	Instance Amount
Person	**64,115**	**262,465**
Chair	12,774	38,491
Car	12,251	43,867
....		
Parking Meter	705	1,285
Toaster	217	225
Hair Drier	**189**	**198**

As previously mentioned, MS-COCO dataset consists of images and instances, providing rich resource for slots and substitution candidates. However, lacking data samples is a common issue due to the limited amounts of slots and candidates. To ensure the selected slot-based filters are functional in general conditions, it is necessary to conduct validation experiments on simulated scenarios. Hence, a mini-COCO dataset is established to run these tests with reduced image and instance amounts, limited slots and preserving the consistency with original MS-COCO. The mini-COCO dataset is required with the followings features:

1. Reduced instance and image amounts to simulate the realistic scenario.
2. All MS-COCO instance categories/super-category are included.
3. Reasonable slot amount.
4. Relatively balanced in an instance level. Table 2.6.1 has listed some of the category amounts in MS-COCO, in which some categories have much more instances than the other.

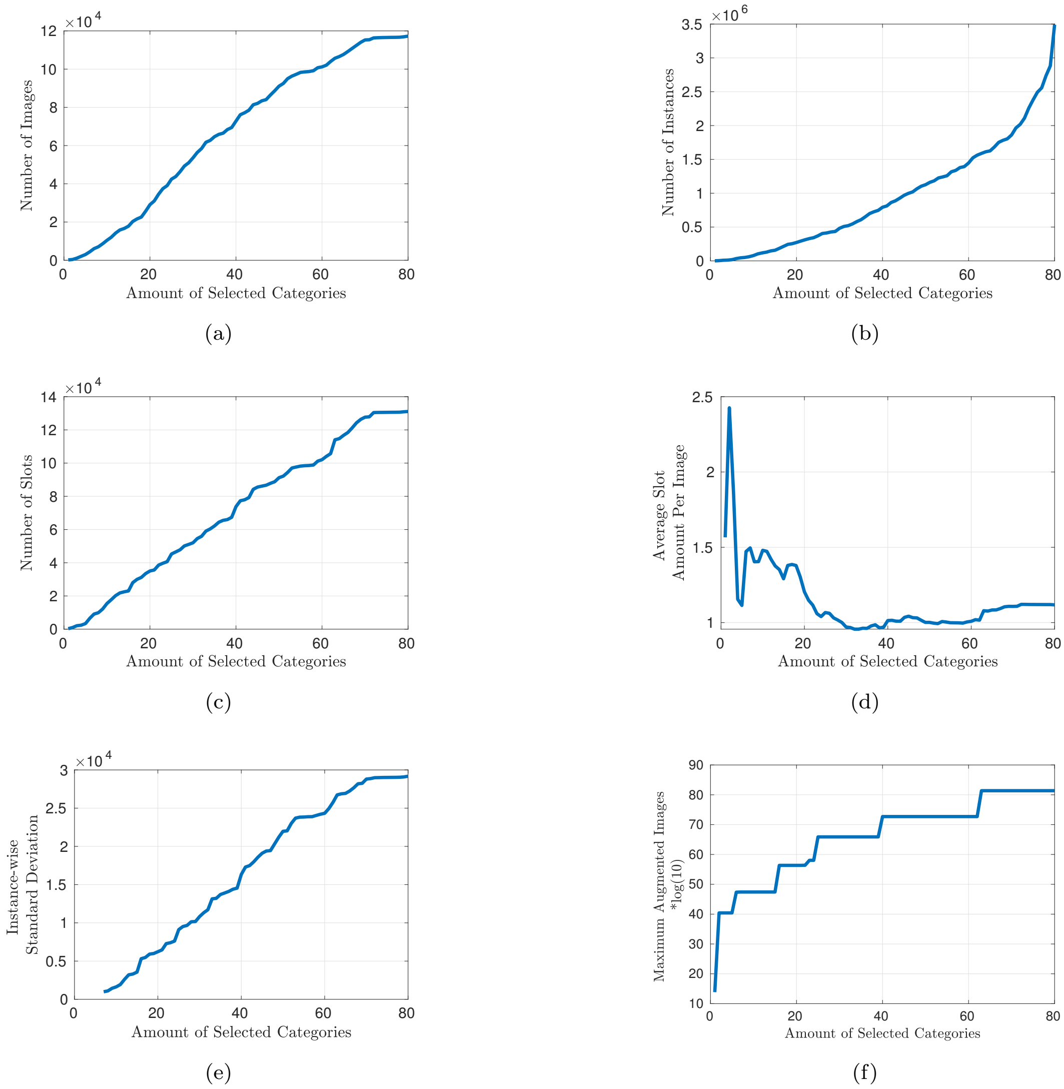
To maintain those desired features, the mini-COCO dataset is established in a heuristic way to iteratively fetch images from MS-COCO as presented in Algorithm 2.6.1. Based on MS-COCO annotations, all 80 categories are sorted in ascending order of instance amount. After some other data structure-wise setup, the desired mini-coco dataset is selected from a set of candidates delivered by an accumulative operation of stacking images containing a certain category instance. In other words, a single category with the smallest instance amount is selected by previous sorting operation and those images containing instances of the selected category, are added into mini-coco as a candidate. As the iterative progress, these mini-coco candidates are evaluated with a series of metrics including standard deviation of instance amounts for dataset balancing concern, slot amount, a summary of instance and image amounts and whether all categories are included. The final mini-coco is selected based on the comparison of those feature-related metrics and in our case, mini-coco dataset is selected at the 20th category is selected containing 73,653 instances in 10,436 images with all instance categories included. In addition, mini-coco contains approximately 9% images (8.8%) and instances (8.6%) comparing with the original MS-COCO which contains 118,287 images and 860,001 instances. Figure 2.9 has presented the related metric values as stacking categories into the mini-coco candidates.

Algorithm 1 A heuristic algorithm to construct mini-COCO dataset

```
procedure CONSTRUCT MINI-COCO
    Initialisation:
    num_cat                                   ▷ category amount in MS-COCO
    sorted_array ← MS-COCO categories sorted by instance amount in an ascending order
    mini_coco ← []
    records ← []          ▷ calculate and store values corresponding to dataset requirements including standard deviation (std), slot amount (slot_amount) and check whether all categories are included (category_included)
    Progress
    while i from 0 to num_cat − 1 do                ▷ iterate items in sorted_array
        mini_coco[i] ← images and associated annotations containing category sorted_array[i]
        CLEANUP          ▷ remove selected images and annotations from MS-COCO
        records[i] ← std, slot_amount and category$_i$ncluded
    Output records and mini_coco
    RETURN mini_coco[a]    ▷ mini_coco[a] is manually chosen based on records and mini_coco.
```

2.6.2 Filter Details and Validation

As discussed above, the chosen filters are aspect ratio, scale and category. In this section experiments are conducted on testing turning on and off these filters using mini-coco



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

FIGURE 2.9: **Row one:** image and instance amount as selecting more categories. **Row two:** overall slot amount and average slot number per image of mini-coco as selecting more categories. **Row three:** instance standard deviation and augmentation capacity.

dataset and baseline “ “Faster RCNN model.

In practice, filters are defined as a selector to pick valid candidates from a certain range w.r.t. the scheme. For example, the scale filter compares candidates’ scales and preserve those within ±20% of the original scales. The aspect ratio filter works in the same way and the category filter. The proportion threshold ±20% is a trade-off between candidate quality and candidate amount. The threshold is preferred to be larger When the dataset includes a large number of instances and smaller with a relatively small dataset. Category filter is applied according to specific scenario. For the task of generating more images with the same category, candidates are preserved only if they belong to the same category as the original slot instances.

Table 2.3: Filter ON/OFF Validations

Category	Scale	Aspect Ratio	mAP
ON	**ON**	**ON**	**0.149**
OFF	ON	ON	0.146
ON	ON	OFF	0.147
ON	OFF	ON	0.148

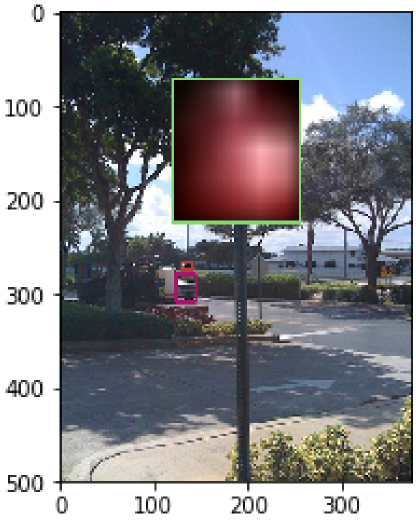


Figure 2.10: A negatively generated image, in which the street sign is too small to fit into the slot.

Figure 2.6.2 shows the experiments on baseline model testing turning on or off those filters on mini-coco dataset. The experimental results demonstrate the necessity of turning on these filters and Figure 2.10 gives an invalid example image without turning on the scale filter and Figure 2.11 presents the case of inappropriate aspect ratio.

2.7 Performance-based Image Augmentation

With the aforementioned filters, slot-based augmentation system can be applied in many different scenarios and this section presents an example of improving detection performance by generating images with the substitution of same category instances.

The performance-based image augmentation is based on the training mAP and P-R details of the baseline model. In this case, the original "Faster RCNN model is selected as the baseline model and trained on MS-COCO dataset with 490k batches without any augmentation (same hyperparameter configuration as Ren et al. (2015a)). The baseline model produces 30.5% overall mAP (C75) that reaches the original benchmark. However, mAP alone is biased to describe model performance and weakness. Hence, detailed

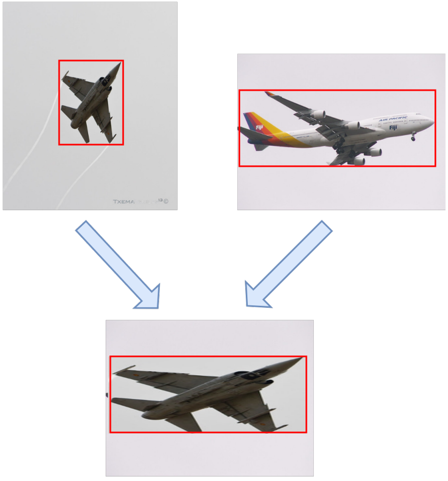
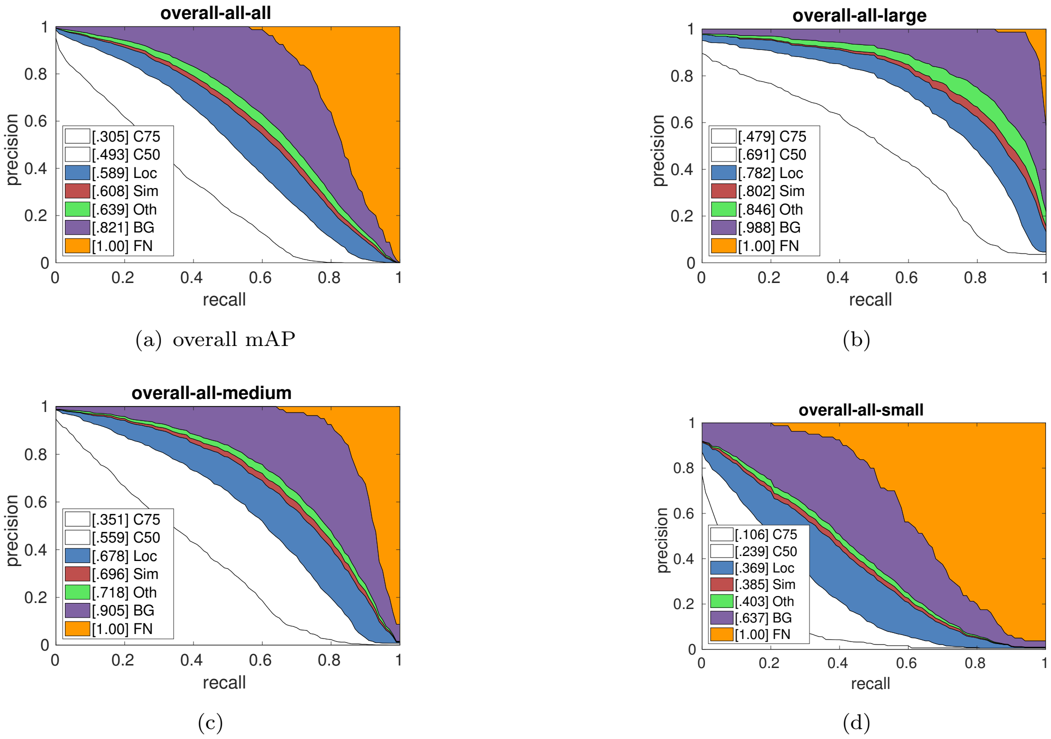


FIGURE 2.11: Object feature loss is caused by substituting the target slot with an instance of different aspect ratio. In this case, the target slot ratio is width larger than height while the substitution is opposite.



(a) overall mAP

(b)

(c)

(d)

FIGURE 2.12: Derek Style P-R Curves of Baseline "Faster RCNN Model without Image Augmentation.

Table 2.4: Number of Augmented Images with Different Approaches

Augmentation Methods(490k)	Number of Original Images	Number of Augmented Image	Proportion of Original Images
No-Flipping	118,287	0	0%
Flipping (All Categories)	118,287	118,287	50%
Ours Cars	12,251	3,262	78.97%
Ours Boats	3,025	940	76.29%
Ours Traffic Lights	4,139	2,224	64.05%
Flipping and Ours Cars	12,251	15,513	44.13%
Flipping and Ours Boats	3,025	3,956	43.31%
Flipping and Ours Traffic Lights	4,139	6,363	39.41%

P-R curves in Figure 2.12 are introduced to evaluate the scale-wise performance. In addition to the overall performance, category-wise P-R curves are analysed as well. The P-R curve and detailed mAP performance indicate the imbalanced performance that some categories have fairly low mAP due to limited image and instance amount. To address the issue of low mAP of certain categories, augmenting extra images to provide extra learn-able features is expected a performance improvement, hence the motivation of using slot-based augmentation system.

Based on the analysis of the mAP and instance amount, three categories are selected to demonstrate the process of augmenting images to improve the performance: **cars**, **ships** and **traffic lights**. These three categories initially have fewer instance amounts, lower detection mAP and a reasonable number of slot. To augment more images containing these category instances, the slots are substituted with candidates of the same category and the ±20% threshold is applied on the scale and aspect ratio filters. Based on the selected categories, the widely used flipping and slot-based approaches are compared in the aspects of various mAP metrics, the amount of generated images and the effects of combining them together. Table 2.7 displays the augmentation amount with different augmentation methods, in which flipping images is the commonly used approach in many detection systems. In addition, these augmented images also change the probability of training on an original image in other words, the chances of learning original features or artificial features. Details of the augmentation performance are presented in the following sections. **Note** the slot-based method is conducted one epoch so that one slot is only substituted with one candidate.

2.7.1 Augmenting Cars Images

MS-COCO contains 43,867 car instances distributed in 12,251 images. Following the pipeline discussed in Section 2.5.1, isolated slots are selected and replaced with candidates satisfying the requirements of the filters. In other words, the car slots are substituted with those candidates with close aspect ratio, scales and the same category. In the experiments, 3,262 images are generated by replacing slots by one iteration and the sample images are shown in Figure 2.15(a).

Regarding augmentation effects on detection performance, Table 2.7.1 presents baseline model mAP with different augmentation methods of all scales. Table 2.7.1, Table 2.7.1 and Table 2.7.1 describe the same schemes on objects with scales of large, medium and small relatively. To summarise these results, slot-based augmentation system generates around 3000 images (over 12,251 images) and train on these extra images improved the overall mAP 2% comparing with commonly used "**flipping**" and non-augmentation. Regarding the training mAP of different scales, it can be observed that slot-based augmentation has greater effects for medium and small objects, boosting 1.6% mAP for medium instances and 0.9% for small scale (comparing with not applying any augmentation methods, under the **C75** metric). Furthermore, the mixing of slot-based method and flipping, increased 1.4% mAP while 0.5% for flipping only. Despite the mAP improvements of flipping, the overall mAP from Table 2.7.1 shows a 0.7% mAP decreasing which might be the reason for medium scale instances in Table 2.7.1.

TABLE 2.5: Baseline Model mAP on **Cars**

	C75	C50	Loc (C10)
No Flipping	0.268	0.513	0.688
Flipping	0.261	0.510	0.691
Ours	**0.281**	**0.509**	**0.679**
Flipping + Ours	**0.276**	**0.515**	**0.677**

TABLE 2.6: Baseline Model mAP on **Cars (Large Scale)**

	C75	C50	Loc (C10)
No Flipping	0.538	0.776	0.874
Flipping	0.599	0.768	0.872
Ours	**0.580**	**0.778**	**0.883**
Flipping + Ours	**0.538**	**0.802**	**0.883**

TABLE 2.7: Baseline Model mAP on **Cars (Medium Scale)**

	C75	C50	Loc (C10)
No Flipping	0.497	0.727	0.860
Flipping	0.486	0.727	0.842
Ours	**0.513**	**0.736**	**0.848**
Flipping + Ours	**0.501**	**0.729**	**0.841**

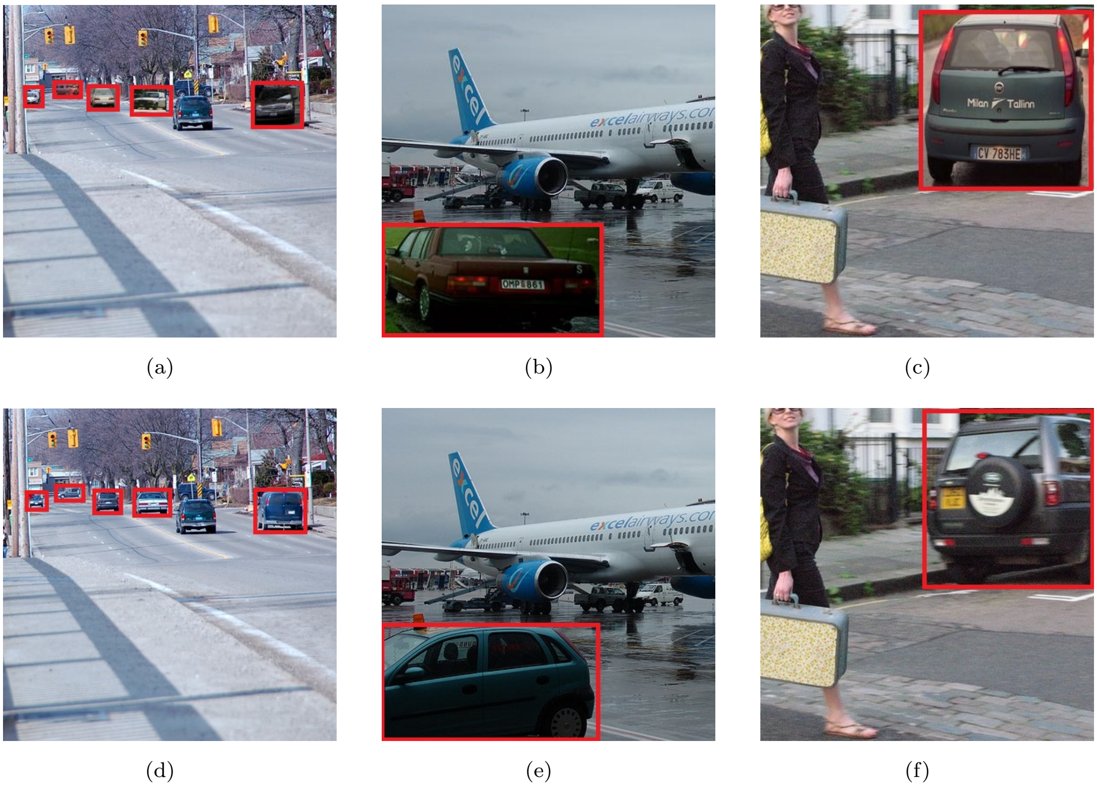


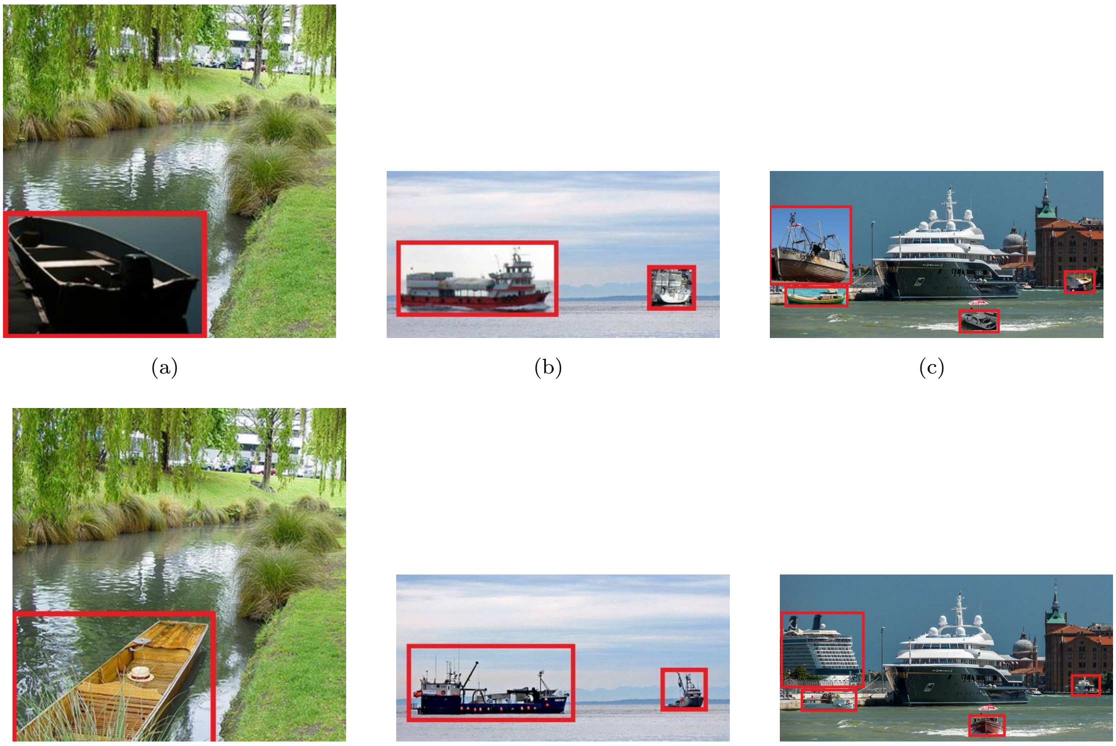
FIGURE 2.13: **Row one:** augmented car images. **Row two:** the original images (some images are cropped for presentation purpose).

TABLE 2.8: Baseline Model mAP on **Cars (Small Scale)**

	C75	C50	Loc (C10)
No Flipping	0.118	0.384	0.620
Flipping	0.123	0.372	0.622
Ours	**0.127**	**0.374**	**0.611**
Flipping + Ours	**0.132**	**0.376**	**0.599**

2.7.2 Augmenting Boats Images

Similar to augment images with car instances, after one iteration, 940 boat images are generated from 3,025 images with 10,759 instances. Figure 2.14 shows three sample images before and after slot substitution. Considering the training mAP, the flipping and slot-based augmentation methods actually decreased the mAP by 0.9 % and 3.1% respectively. While combining these two augmentation methods yields an increasing 2.3% mAP. For different scale objects, the two augmentation methods decreased detection mAP at some extent. But the slot-based method contributes a 2.8% improvement for medium scale objects and flipping method rises a 1.3% improvement on small scale objects. In general, for the boat-related image augmentation, individual augmentation failed to improve training mAP with a decreasing effect instead. However, applying these two methods together improved detection mAP at all scales.



(a) (b) (c)

(d) (e) (f)

FIGURE 2.14: **Row one:** augmented boat images **Row two:** the original images

TABLE 2.9: Baseline Model mAP on **Boats**

	C75	C50	Loc (C10)
No Flipping	0.127	0.37	0.594
Flipping	0.118	0.366	0.600
Ours	**0.096**	**0.351**	**0.592**
Flipping + Ours	**0.140**	**0.385**	**0.603**

TABLE 2.10: Baseline Model mAP on **Boats (Large Scale)**

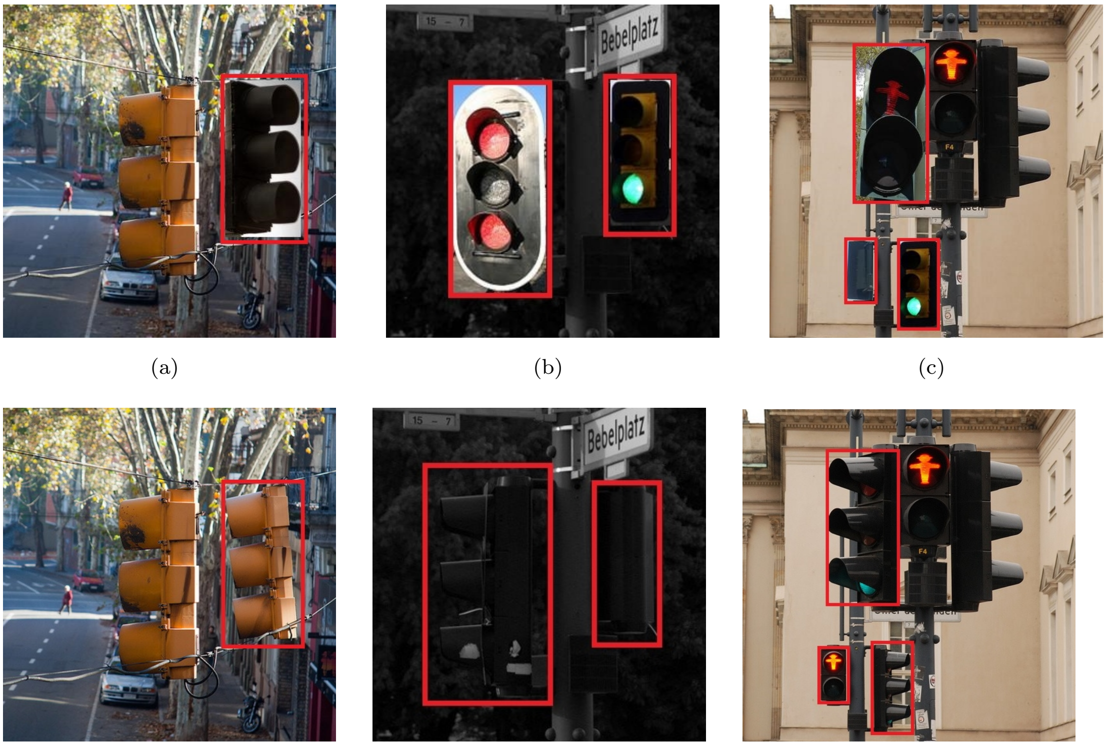
	C75	C50	Loc (C10)
No Flipping	0.342	0.639	0.778
Flipping	0.323	0.591	0.769
Ours	**0.239**	**0.570**	**0.784**
Flipping + Ours	**0.428**	**0.630**	**0.795**

TABLE 2.11: Baseline Model mAP on **Boats (Medium Scale)**

	C75	C50	Loc (C10)
No Flipping	0.125	0.474	0.727
Flipping	0.103	0.460	0.728
Ours	**0.153**	**0.504**	**0.757**
Flipping + Ours	**0.134**	**0.428**	**0.711**

Table 2.12: Baseline Model mAP on **Boats (Small Scale)**

	C75	C50	Loc (C10)
No Flipping	0.043	0.259	0.564
Flipping	0.056	0.272	0.585
Ours	**0.029**	**0.230**	**0.563**
Flipping + Ours	**0.063**	**0.275**	**0.565**



(a) (b) (c)

(d) (e) (f)

Figure 2.15: **Row one:** augmented traffic-light images **Row two:** the original images

2.7.3 Augmenting Traffic Light Images

For the traffic light category, 2,224 images are augmented based on 12,884 instances from 4,139 original images within one epoch. Some sample images are shown in Figure 2.15(a). In the aspect of detection mAP, both flipping and slot-based augmentation have improved the mAP. The improvement for flipping is 0.3% and 2.0% for slot-based method. The combination of these two methods is 2.0% as well. Observing the mAP changes on different scales, the flipping method improved 3.2% and 4.9% for large and medium objects while a decreased 0.4% for small objects. The slot-based method produces 2.9% and 3% for small objects. However, it decreased by 0.7% for medium objects. The combination approach improved all the three scales with 3.5%, 3.3% and 1.5% for large, medium and small objects respectively.

Table 2.13: Baseline Model mAP on **Traffic Lights**

	C75	C50	Loc (C10)
No Flipping	0.096	0.363	0.539
Flipping	0.099	0.366	0.553
Ours	**0.116**	**0.377**	**0.534**
Flipping + Ours	**0.116**	**0.379**	**0.547**

Table 2.14: Baseline Model mAP on **Traffic Lights (Large Scale)**

	C75	C50	Loc (C10)
No Flipping	0.431	0.658	0.865
Flipping	0.463	0.687	0.853
Ours	**0.460**	**0.656**	**0.833**
Flipping + Ours	**0.566**	**0.727**	**0.896**

Table 2.15: Baseline Model mAP on **Traffic Lights (Medium Scale)**

	C75	C50	Loc (C10)
No Flipping	0.265	0.683	0.811
Flipping	0.316	0.682	0.823
Ours	**0.258**	**0.660**	**0.837**
Flipping + Ours	**0.298**	**0.686**	**0.825**

Table 2.16: Baseline Model mAP on **Traffic Lights (Small Scale)**

	C75	C50	Loc (C10)
No Flipping	0.050	0.291	0.478
Flipping	0.046	0.299	0.497
Ours	**0.080**	**0.316**	**0.466**
Flipping + Ours	**0.065**	**0.311**	**0.485**

2.8 Discussion and Analysis

As the main metrics of evaluating object detection models, mAP provides an overview of model performance. However, for different category instances, the mAP per category varies a lot. A normal case is that categories with a large number of instances come with higher mAP such as the "person" category with 44.4% mAP (C75) containing 262,645 instances in 64,115 images. While those with small instance amounts have much lower mAP such as "hair drier" 1% mAP (C75) with only 198 instances in 189 images. The previous section has presented the experiments of applying slot-based augmentation method into three different category instances. These experimental results have shown the benefits of applying augmentation methods. In general, applying image augmentation is expected a higher overall detection mAP. However, the augmentation affects differently over categories and sometimes it even decreased the detection mAP, such as applying the flipping method in boat objects. Compared with flipping images, slot-based augmentation produced closing mAP improvement. For example, the augmentation on car objects behaved even better than flipping with an extra 2% mAP. Besides, the slot-based augmentation added fewer images comparing with flipping which doubled the image amount. For a large dataset such as MS-COCO, flipping actually increased a large number of images causing extra training time and computation resources. While the slot-based augmentation method is relatively more flexible to control the amount of augmentation with less increased computation time. As the performance for some categories is not improved, combining these methods together produced a consistent improvement over all these three categories. For the large boat instance, both flipping and slot-based augmentation failed to improve mAP while the combination provides an 8.6% increase.

In conclusion, image augmentation is frequently applied in object detection based tasks to provide extra learn-able features and improve the detection mAP. One of the commonly used methods is flipping all the images of the dataset. In this Chapter, a slot-based image augmentation method is proposed, in which images are augmented by replacing isolated foregrounds to provide extra combinations of foreground and backgrounds. Additionally, the system components are tested by a series of filter experiments and an augmentation scenario is presented showing the detailed procedure of applying this method. Besides the expected mAP improvement with augmentation methods, the experimental results have highlighted several interesting facts:

- Instance amounts affect category-wise mAPs. Some categories with lower mAP result from lacking data, which highlights the importance of applying image augmentation methods on those specific images without rising extra training time. In addition, detailed mAP and the Derek P-R curve is descriptive on analysing detailed performance.
- Image augmentation is not always bringing benefits to detection performance while decreased the mAP instead such as the experimental results of boat instances.
- According to experimental results, the slot-based augmentation approach performed closed behaviour as image flipping with less increased images. Besides slot-based augmentation has large potential to generate images which is controllable progress by customising slot match ratio and the iteration amount.
- The combination of flipping and slot-based method is relatively more robust and contributes a higher mAP.

Despite those benefits of slot-based augmentation methods, there is a notable effect on contextual features. In this chapter, augmentation methods are discussed on improving detection performance while there are many DNN architectural works aiming at extracting the features between foreground and background to enhance detection performance, such as the relation model proposed by Hu et al. (2018). Slot substitution crops original foreground and replaces it with a new one which changes the foreground-background relational features. However, this type of changes has critical effects. On one hand, breaking the original contextual features makes it difficult for DNN models to detect objects according to their surroundings. On the other hand, breaching original contextual features potentially contributes to a robust feature extraction that recognising objects fully by its features, not by the environments. For example, human beings recognise the flying beer in the sky from advertisement posters.

2.9 Potential Scenarios and Future works

Despite the scenario demonstrated in this chapter, the slot-based augmentation has many potential applicable cases such as the “mask-wise” slots, “external candidates” and “artificial slots”. The conception of “slots” in this chapter is viewed as rectangle bounding boxes and augmentation is conducted by replacing foregrounds inside the slots with others. However, the changes of slots also cause unavoidable distortion of the contextual information between the slot and background images. So the **“mask-wise” slot** is motivated by using polygons instead of rectangles, which has fewer damage effects on the contextual information. Figure 2.16 shows simple comparison of these two types of slots. Furthermore, the candidate matching mechanisms can also be changed to a series of polygon related metrics and a customised triplet loss by Zhao et al. (2018) gives inspiring insights.

Considering the slot substitution, it is a common case to have insufficient candidates especially setting strict filter thresholds. In that case, more available candidates are desired and the external classification images can be viewed as extra resources. Images from classification dataset contain one single object per image and less complicated backgrounds, which is similar to the slot instances such as widely used ImageNet, CIFAR and MNIST datasets (Deng et al., 2009; Krizhevsky and Hinton, 2009; LeCun and Cortes, 2010). Categories of these datasets such as “cat”, “dog” or “cars” have large

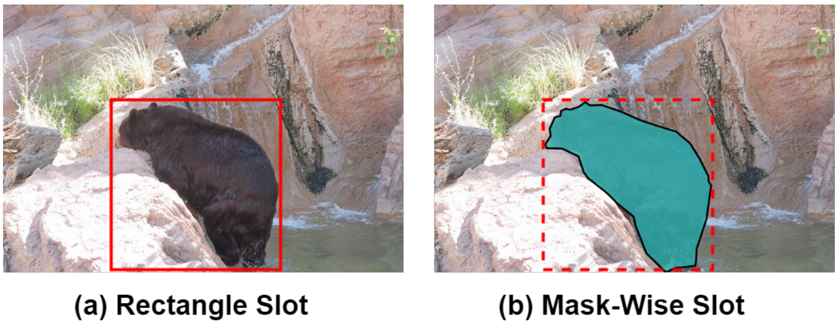


Figure 2.16: A Comparison of Rectangle and Mask-Wise Slot. Represented as polygon, the mask-wise slot has less background information.

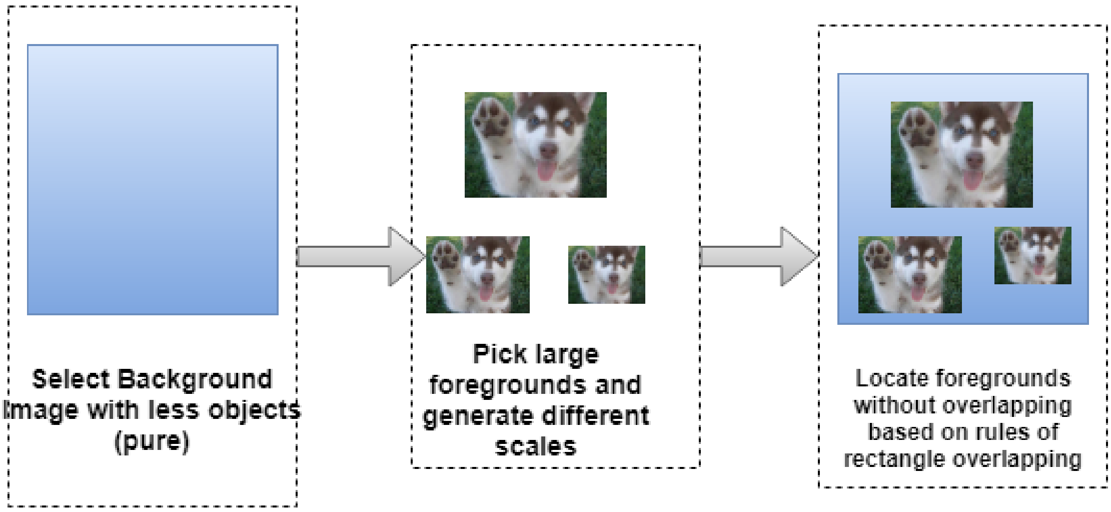

Select Background Image with less objects (pure)
Pick large foregrounds and generate different scales
Locate foregrounds without overlapping based on rules of rectangle overlapping

Figure 2.17: Artificial Slots for Augmenting Images Containing multiple Scale Instances.

potential usability.

Besides, detecting small objects is a challenging task due to limited scales and using available background information is the main focus on this topic. However, as discussed above, these approaches have some unavoidable drawbacks such as high potential dependencies on backgrounds, which potentially leads limits on generalisation capacity. Instead of learning contextual relational features, extracting scale-invariant features is an alternative approach. It conducts detection by the features that are less affected by object scales. Scale-invariant features have less dependency on the background and robust to the abnormal foreground and background combinations. While it is still a difficult problem to extract such features from a large dataset with DNN models. Inspired by the analysis of Singh and Davis (2018), extracting scale-invariant features desires an image to contain the same category objects with different scales and manipulating the loss function to regress the similar detection outcomes on these varied objects is a potential solution. Therefore, the idea of **"artificial slots"** is motivated to created isolated slots with different scales to generate images including various scales of same category instances, as the demo is shown in Figure 2.17. In addition, many compulsory definitions are required such as selecting a suitable background image, the amount and scale of isolated slots in an image, mask or rectangle etc.

All in all, the key idea of slot-based augmentation is regarding the isolated instances as replaceable "slots" and enriching the dataset with various combinations of foreground and background. Furthermore, the point of augmenting images in the DNN context is generating the "desired" image which can be observed by detailed model performance analysis using Derek P-R curve or various mAP values. These "required images" also draw several promising scenarios indicating many potential extensions to addressing different challenges in object detection research areas.

Chapter 3

StomaRCNN

[1] The previous chapter has demonstrated an approach to improve detection performance by augmenting extra images. Regarding the overall process of training DNNs, the architecture pipeline plays an important role in problem modelling and precise outcomes. This chapter is aimed to present an idea of pipeline reorganisation to improve detection accuracy utilising commonly used DNN models. The modified pipeline (denoted as "StomaRCNN") is introduced in a realistic scenario in plant science to detect, segment and measure the "binary-status" objects called "**stoma**". By splitting the aim into detection and segmentation tasks, the detailed object information of the high-resolution microscope images, are preserved for precise segmentation and measuring. The validation results have shown modified pipeline models achieved human-level measurements for open stomata pores. Though the closed pores require further refinement works such as adapting the "key point detection" from facial recognition. Details of StomaRCNN are presented in the following paragraphs.

The opening and closing of plant pores called stomata are essential to life - they control carbon dioxide absorption from the atmosphere and release of oxygen and water, feeding global water and carbon cycles. Biologists frequently measure stomata using microscopy imaging. However, this manual procedure is time-consuming and laborious. The main challenge is to accurately measure stomatal pores represented as multiple small objects in a microscope image. Here, we propose StomaRCNN, an object detection based multistage system, which is composed of Faster RCNN and Mask RCNN, to automatically detect, segment, and measure stomatal pores. The proposed system provides a fast, automatic and flexible solution for biologists. Experiments are conducted on stomatal datasets acquired from barley, the world's fourth-largest food crop and essential for beer

[1] This work collaborates with Na Sai from ARC COE Plant Energy Biology from the University of Adelaide, SA, Australia. During the collaboration, the thesis author is in charged of deep learning related works including architecture design, model training and implementation. Na Sai is responsible for plant science related works and providing raw laboratory image data together with annotation works. Furthermore, the heuristic works are considered as equally contributed.

production. The experimental results showed our proposed approach achieved an average accuracy of 95% in measuring the area and width of open stomatal pores with the overall 0.63 mAP in detecting all types of stomata. Thus, StomaRCNN reached measurement performance in human-level. The proposed system is capable of capturing various visual features automatically for both stomata and pores, assembling the functionalities of detection, segmentation and instance measurements. With the lightweight, highly transferable, fast and flexible system, we achieve state of the art functionality through the implementation of deep learning.

3.1 Introduction

Stomata, derived from the Greek word for "mouth", is the main gate protecting the internal structures of plant leaves from the atmosphere. It plays critical roles in plant physiology such as controlling water loss and carbon gain. Stomatal loss of water (transpiration) drives water distribution through plants and cools leaves Sinha (2004). Photosynthesis uses atmospheric CO_2 to produce carbohydrates as usable energy for plant growth and produces oxygen that is released through stomata to the atmosphere Sinha (2004). Stomata are small pores surrounded by a pair of guard cells, but in some species such as barley, they are flanked by a pair of subsidiary cells as shown in Figure 3.1 Bergmann and Sack (2007); Jarvis and Mansfield (1981). Mostly, stomata are found on the surface of leaves, and are less commonly found on stems, flowers and fruits Hetherington and Woodward (2003); Kirkham (2014). These cells change volume, and hence stomatal aperture, upon daily light and dark cycles, or in response to abiotic and biotic stress to regulate plant productivity McLachlan et al. (2014); Schroeder et al. (2001); Willmer and Fricker (1996); Melotto et al. (2008).

Due to the significance of stomatal regulation to plant physiology, the global water and carbon cycles (and hence climate change), stomata have become an important model system for biologists to investigate the response of plants to environmental stimuli Kim

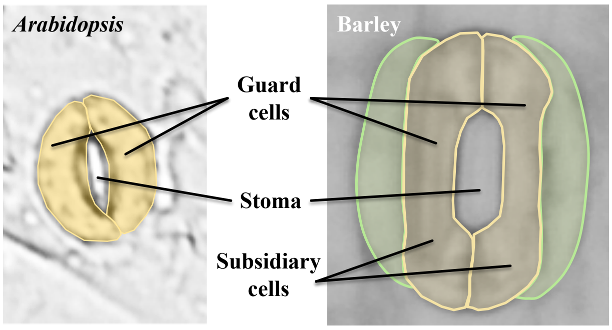


Figure 3.1: **Examples of different stomata morphological structure.** Left: An *Arabidopsis* stoma with kidney-shaped guard cells (in yellow). Right: A barley stoma with dumbbell-shaped guard cells (in yellow) and subsidiary cells (in green). The images are drawn by the collaborator Sai (unpublished).

et al. (2010); Shimazaki et al. (2007). However, using microscopy imaging to examine stomatal behaviour is not simple and straightforward, due to leaf morphological differences between plants and a lack of contrast between stomata and background Ziegler (1987); Chen et al. (2017). In addition, it requires expert knowledge from biologists to identify and measure stomata correctly. Common practice involves manually measuring stomatal width (i.e. stomatal aperture), length or area with image processing software such as Fiji-ImageJ Schindelin et al. (2012). This procedure requires the user to manually specify the points of interest for width or length of stomata or the boundary of stomata pore to specify stomatal area so that the software is able to return relevant results. In order to produce good and accurate results, this common practice is time-consuming, laborious and tedious. Although the ImageJ based plugin developed by Cheng et al. (2014) provides benefits to make stomata related measurement easier and faster, manually tuned parameters are still required to produce the desired performance. Moreover, this is a highly experience-dependent procedure and hard to conduct in an automatic way. Therefore, measurement is commonly a bottleneck as hundreds of images are needed as standard to produce biologically relevant results.

Whilst recently developed tools such as the ImageJ-based Cell Counter Plugin brings the procedure from fully manual to semi-automatic, the efficiency of data retrieval from images in the large scale is still highly desired Cheng et al. (2014). Recent progress in deep learning and computer vision have given rise to an array of impressive semi-automatic procedures. Deep neural networks (DNN) have been developed with various architectures and training mechanisms to address several realistic problems in deep learning and computer vision LeCun et al. (2015). For fundamental object detection tasks, a series of DNNs have been proposed, such as the "speed prior" one stage networks including YOLO Redmon et al. (2016); Redmon and Farhadi (2018), SSD Liu et al. (2016) and RetinaNet Lin et al. (2017b). An alternative approach is the "quality prior" multistage RCNN families such as RCNN Girshick et al. (2014), Fast RCNN Girshick (2015a) and Faster RCNN Ren et al. (2015b). The multistage models use "Region Proposal Network" (RPN) to propose bounding box candidates for final prediction while one stage models directly output predictions by fusing feature maps without an RPN. In general, one stage models are fast and commonly used for real-time detection while RCNNs are used in quality-driven scenarios. Based on the bounding box detection, to achieve an instance level segmentation, Mask RCNN He et al. (2017) and its customised DNNs Dai et al. (2017); Liu et al. (2018) are proposed.

Recently, there have been several reports of using DNNs or Computer Vision techniques to examine stomata related problems (Table 3.1). Among those works, the handcraft feature extractors are commonly used including HOG (Histogram of Oriented Gradients), GIST and DAISY (Dalal and Triggs, 2005; Oliva and Torralba, 2001; Tola et al., 2010). Guided by manual processes, these methods extract features in specific aspects serving related vision tasks. Besides machine learning based models are also used in this area including classifiers (e.g. supporting vector machine (SVM) Cortes and Vapnik (1995), Multilayer Perceptron (MLP) Hornik et al. (1989)), U-Net Ronneberger et al. (2015) based CNN and AlexNet style CNNs Krizhevsky et al. (2012).

To be specific, Laga et al. (2014) achieved stomata detection and segmentation by matching the templates, originally collected from 24 images and these templates are extracted by manual calculation. Similar to manual processing, traditional methodologies played as important components such as HOG based cascade detector in Saponaro et al. (2017). Besides HOG, researchers achieved high accuracy by integrating multiple manual extractors including DAISY, GIST based haralick Texture features and simple classifiers like SVM and MLP Aono et al. (2019). Different from methods mentioned above, U-Net models are specialised at extracting highly abstract features, while preserving the relatively detailed features from previous layers by many deconvolution operations. With appropriate modellings, these U-Net models produced competitive detection results with high recall and precision Saponaro et al. (2017). The DNN based approaches may not behave well alone and combine with HOG contributes an advanced performance for some tasks. For example, deep stomata proposed by Toda et al. (2018) utilised HOG to preprocess raw images making it easier to conduct the desired stomatal measurement with DNNs. Deep learning based methods also make other related works less complicated such as counting stomatal amount as stomatal counter network proposed by Fetter et al. (2018).

Species	Microscope	Method	Function	Reference
Wheat	Laser dissection	Handcraft feature extractors	Detection Measurement	Laga et al. (2014)
Grapevine	Upright fluorescence	Manual HOG	Detection Measurement	Jayakody et al. (2017)
Maize	Optical	HOG, GIST, DAISY, LBP and Haralick	Classication Identification	Aono et al. (2019)
Maize	High-speed confocal	U-Net based DNN	Segmentation	Saponaro et al. (2017)
Dayflower	Stereo	HOG and CNNs	Detection Measurement	Toda et al. (2018)
Maidenhair tree Balsam poplar	Upright fluorescence	AlexNet style CNNs	Detection	Fetter et al. (2018)

Table 3.1: **Summary of related works.**

These proposed methodologies have produced the expected performances on their own specific scenarios. Most of those approaches are highly specialised on specific image data and their approaches are built on handcraft feature descriptors and DNNs with simple architectures. However, for more general scenarios, there are some unavoidable limitations such as special requirements for image capturing equipment or associated feature extractors. Besides, insufficient data (less than 30 images) is another barrier toward more general usability, as intrinsic requirements for those data-driven approaches. In addition, the manual feature descriptors worked well on specific images but less robust compared with the data-driven DNN models. Moreover, it is possible to adapt those feature descriptors for similar species such as barley and wheat, but difficult for distinct species like barley and Arabidopsis as they are form different families. In the aspect of deep learning approaches, classifiers or DNNs used in those proposed methods have simple architecture and fewer parameters. Thus, the limited generalisation capacities cause incapable to handle images acquired with different plant species. Furthermore, considering the idea of using work-flows mixed with data-driven DNNs and manual feature extractors, it is less flexible to reuse the model for a new scenario and difficult to adapt the state of art fine-tuned DNNs.

In practice, an automatic, robust and reusable stomata measurement system is still a challenging problem and highly desired with fewer dependencies on plant species. In addition, the microscope images in plant science consist of fine details with high resolution (2048 by 2880). Processing those images with DNNs requires large GPU memories so re-scaling or cutting them are commonly used methods in many detection systems. Moreover, information loss is unavoidable which makes the problem even harder. To address the existing issues, we proposed StomaRCNN, a multistage stomata detection, segmentation and measurement system with integrated DNN models which are purely data-driven and flexible to be transferred on different datasets Ben-David et al. (2010). The detailed work-flow is presented in Figure 3.2. Data-driven DNNs have ensured the pipeline consistency and robustness to different datasets. Due to DNN architectural commonalities, many pre-trained DNN models (e.g. ImageNet trained CNNs Krizhevsky et al. (2012)) are capable of being transferred to our DNN models. As stomata measurement is a quality-driven task, we use multistage Faster RCNN for stomata detection and Mask RCNN for stomatal pore segmentation instead of one-stage models.

3.2 StomaRCNN

3.2.1 Motivation and Multistage Work-flow

The ultimate goal is measuring morphological features of stomatal pores such as area, width or length from microscope images. However, those stomatal pores are normally represented with few pixels leading to the challenging topic of detecting small objects in computer vision. Despite the small scales, these objects are recognisable given detailed context or instance information. As the resource data, high-quality images like microscope images, contain most of the desired information to detect small objects which is ideal to train DNNs directly. Unfortunately, the original images are formatted in high resolutions that consume unaffordable computational resources.

The common practice to handle this issue is using pooling or cropping to re-scale them as small images with lower resolution to make them trainable. However, these approaches are not feasible in our case, because of the risk of losing detailed information. Reducing details brings difficulties to conduct precise measurement, especially for the stomatal pores. As a result, preserving image quality (i.e. resolution) with affordable computing budgets is the key point to resolve this issue. Therefore, one of the potential solutions is segmenting single stoma in small sub-images with original quality. Comparing with the original image, training on sub-images consumes much fewer resources while preserving the same quality. Ideally, the sub-image is completely overlapped with the benchmark bounding box and a well-trained detection model is expected to output tight bounding boxes with one single instance inside (i.e. one single stoma). Considering the key point of retrieving tight bounding boxes from well-trained models, simple object detection task requires fewer details comparing with segmenting objects.

Therefore the re-scaled images (1/5 in our case) by common approaches are applicable to train object detection models.

These insights suggest a solution to detect bounding boxes on re-scaled images and segmenting instances on the associated area from the original image with less computational cost and preserve the detailed image information. Adapting to the goals of measuring, we use Faster RCNN to detect stomata bounding boxes on re-scaled images. Based on the coordinates of bounding boxes, the associated region of interest is cut from the original image through linear projection (i.e. reverse the re-scaling process). The Mask RCNN is applied on the transferred single stoma images to segment boundary of the stomatal pore as polygons (open stoma) or lines (closed stoma). By geometrical calculations, the target measurement of area, length and width are conducted to produce the final results.

3.2.2 Detailed System Workflow

As shown in Figure 3.2, bio-formatted(Nikon .nd2) microscope images are required as the input to this pipeline and all images are converted to png files through an ImageJ Macro script. To decrease the computational budget at the existence detection stage, these converted images are re-scaled as 1/5 of the original width and height. Then, through Faster RCNN, stomata are detected from each re-scaled image with a confidence score attached. Based on the bounding box coordinates, each detected stoma is cut from the original sized images. This strategy avoids computational intensiveness and reduces the noise of the surrounding background. The coordinates of the bounding box detected from re-scaled images are transformed through linear projection and each bounding box contains a single stoma. An opening status is then recognised by Mask RCNN for each sub-image (i.e. a single stoma) with a confidence score attached. For open stomata, the stomatal pore is predicted as a polygon with a confidence score, while a closed stomatal pore is detected as a tight rectangle with a small vertical height. When stomatal pore detection finishes, the area, height and length of the polygon are calculated to represent the stomatal area, width and length, whereas the length of the line is measured to indicate the length of a closed stomatal pore. Measurement data are gathered and displayed with a corresponding width/length ratios. Stomatal density and total number are also provided in a user-readable way. During the procedure, confidence thresholds are set by users before passing to the image segmentation stage and this threshold is applied to the confidence check in the existence detection, opening status detection and measurements detection. Any (sub-)images with a confidence score below the threshold, manual measurement is required and output is assimilated into the final output.

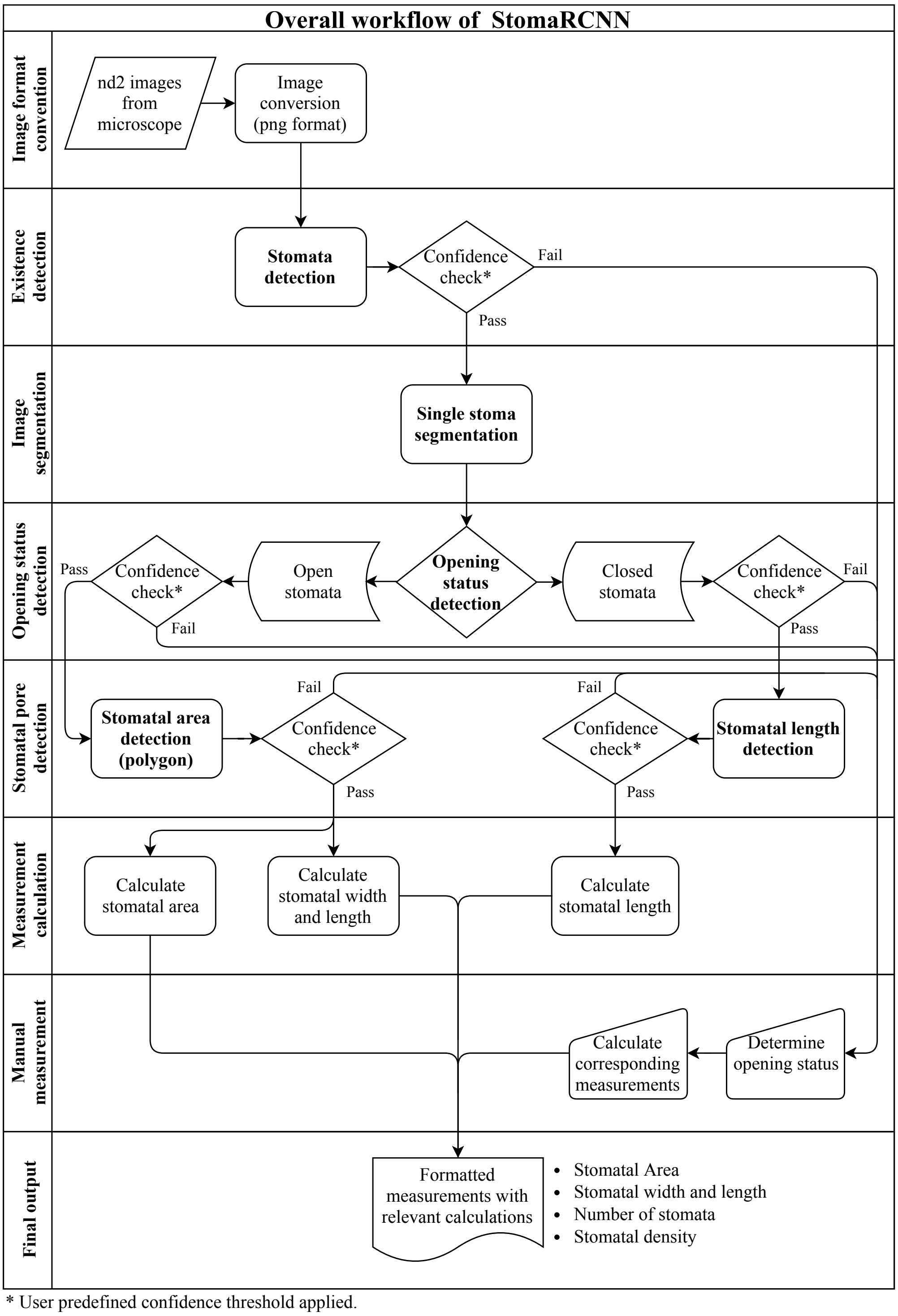


Figure 3.2: **The simplified system work-flow of StomaRCNN.**

3.2.3 Yes or No: Detecting Stomata from Background

One of the prerequisites of measuring stomatal pores is detecting stomata locations. The Faster RCNN is trained to produce tight bounding boxes telling foreground instances from backgrounds. As in Figure 3.3, Faster RCNN consists of backbone CNN, Region

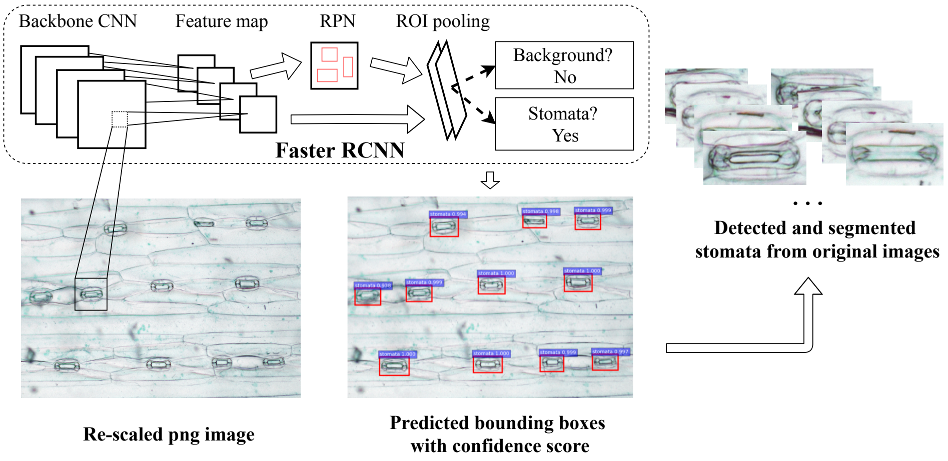


Figure 3.3: **The work-flow of stomata detection.** Sub-images of each single stoma is retrieved from the original high resolution images based on the predicted coordinates by Faster RCNN. Accompanied with transferred annotation, these sub-images are automatically transformed as a dataset for instance segmentation.

Proposal Network(RPN), Region of Interest (ROI) pooling and task head layers (stacks of fully convolutional layers). In detail, the backbone CNN extracts structural features from the input images which are subsequently processed by RPN for proposing anchor boxes. Then, the following ROI pooling layers and the task head produce transferred proposals as final output: object locations and whether it is a stoma or background. In this case, Faster RCNN produced the stomata locations as bounding boxes for the subsequent procedure to form segmentation dataset.

3.2.4 Cutting and Matching: Create Segmentation Dataset

As described in Section 3.2.1, individual dataset consisting of sub-images and annotations, is required for Mask RCNN training. Due to the uncertainty of the bounding boxes predicted by Faster RCNN, the desired dataset cannot be prepared independently. Therefore, we applied strategies to transform the desired dataset automatically by image cutting and annotation matching. The sub-images images are acquired directly by projecting the predicted bounding boxes (from re-scaled images) into the original images. Each sub-image contains only one single stoma with detailed information preserved, contributing a better segmentation performance for Mask RCNN, as in Figure 3.3.

In addition, annotations are acquired by matching sub-images with the benchmarks in original images, which include detailed annotation for the sub-images. Here we simplify this procedure by whether the centroid of sub-images fall into associated benchmark bounding boxes. Comparing with other alternatives of box vertex matching, this approach has significant reduction on algorithm complexity.

3.2.5 Open or Close: Detecting stomatal opening status

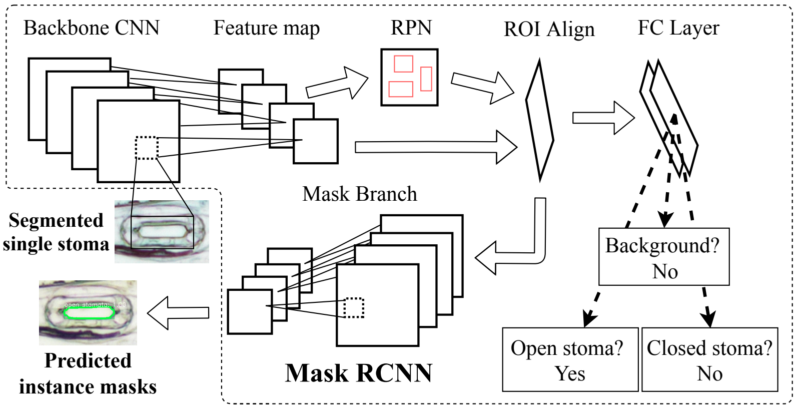


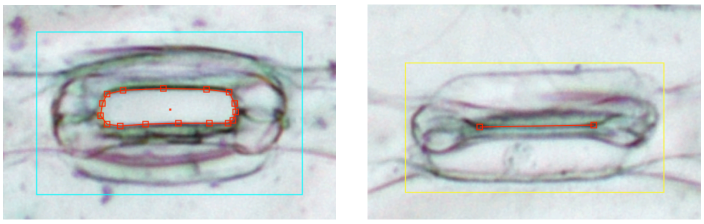
FIGURE 3.4: **The work-flow of stomatal pore detection.** Based on the automatically transferred dataset, the stomatal pores are segmented as the polygons (open stoma) or decorated tight rectangle (closed stoma) by Mask RCNN model. Consequently, those polygons or rectangles are ready for measurement.

With the details preserved, we use Mask RCNN model to segment stomatal pores and predict opening status (i.e. categories). As the architecture is shown in Figure 3.4, Mask RCNN follows similar inference procedures as Faster RCNN but uses extra mask branch to predict the instance outline shapes. In other words, the Mask RCNN model segments the instance category, bounding box location and outline shape as polygons. The desired measurements (stomatal area, width and length) are retrieved by geometric calculations.

3.3 Experiments

3.3.1 Experimental Settings

With the limited image amount, it is challenging to train DNNs on the small dataset as it normally requires more learn-able features and easily gets model overfitting. Several strategies are applied to solve this issue in various aspects: data augmentation and modified transfer learning. To be specific, the dataset is doubled with horizontally flipped images. Besides, our RCNN models are built on ResNet-50 He et al. (2016) pre-trained on ImageNet Krizhevsky et al. (2012) to transfer the knowledge learnt from large dataset to improve the model capacity. We used mean average precision (mAP) to evaluate detection outcomes supported by official COCO-API Lin et al. (2014); Everingham et al. (2010), detailed implementation process is presented in Section 3.3.3. Section 3.4 provides details on data split, model selection, training hyperparameters, avoiding overfitting, and so on. A deployment instruction is provided as an instruction



(a) Open stoma **(b)Closed stoma**

FIGURE 3.5: **Annotation examples for stomata.** Bounding boxes contain a single stoma with opening status, are determined in different colour (open stoma in cyan, closed stoma in yellow). The red polygon and line present defined stomatal pore in each annotation.

for deploying StomaRCNN in a local machine, at Appendix A. Due to the data confidential restrictions, I'm not permitted to upload a full GitHub repository for public access. However, section 3.4 and the deployment introduction have provided sufficient details on implementation and system deployment.

3.3.2 Data Preparation

Leaf samples from two-week-old barley, were peeled as described in Shen et al. (2015) and images were captured by Nikon DS-Fi3 digital camera attached on the Nikon Diaphot 200 inverted microscope. Original-sized png images were annotated using RectLabel (version 2.62). During this annotation procedure, two annotations were given to each stoma (Figure 3.5). First, the bounding box annotation which contains a single stoma (i.e. a pair of guard cell and subsidiary cell) whose opening status is labelled. Second, the measurement annotation which defines the stomatal pore with a polygon (for open stomata) or line (for closed stomata). The process of creating measurement annotation followed exactly the same manual measuring procedure in Fiji-ImageJ and all annotations are organised in a widely used MS-COCO format Lin et al. (2014). The fully annotated dataset contains 1089 instances (recognisable stomata) from 92 images falling into two categories ("open stomata" or "closed stomata"). After filtering invalid instances (e.g. truncated), 841 valid instances from 92 images are obtained. Filtered images are randomly split into training set (82 images with 757 instances) and validation set (10 images with 84 instances).

3.3.3 System Implementation Details

The system is implemented with PyTorch 1.0 Paszke et al. (2017) and CUDA 10.0 on a Linux workstation equipped with four NVIDIA GTX 1080 GPUs. In the stomata detection task, a Pytorch version Faster RCNN model is applied, and the ResNet-50 model pre-trained on ImageNet is transferred as its backbone model.

The Faster RCNN is assembled with ImageNet pre-trained ResNet-50 as the backbone CNNs Deng et al. (2009); He et al. (2016), adapted from Chen and Gupta (2017). Under the consideration of lightweight, the Mask RCNN was implemented based on Massa and Girshick (2018) and used similar backbone ResNet-50 CNN as Faster RCNN. Other hyperparameters are set according to the dataset instance details such as mean pixel value for data normalisation, RPN anchor scales and ratios. Moreover, various Python libraries are used for other mentioned procedures including PANDAS, Numpy, scikit-image, OpenCV and Shapely for stomata pore measurements (McKinney, 2011; Van Der Walt et al., 2011; Van der Walt et al., 2014; Bradski, 2000; Gillies et al., 2007–).

3.3.4 Training Faster RCNN

The Faster RCNN model is trained with 10K image batches with obtaining an mAP of 0.603. The confidence threshold of stomatal detection is set as 50% to acquire more predictions, surprisingly most of the inference gain confidence scores of over 90% as in Figure 3.3. 99.52% of predicted bounding boxes detected single stoma (837 over 841 valid instances) and transformed into the segmentation dataset. These results have shown that our “light-backbone” Faster RCNN model had a strong capacity to extract nearly all the stomata with high metric values.

As mentioned in section 3.2.3, Faster RCNN is trained to produce stomata bounding boxes for cutting sub-images as in Figure 3.3. Since Faster RCNN model transferred ResNet-50 as its backbone, it is a common practice that the first several layers of the pre-trained model are frozen, to utilise the features they learnt which widely fits many vision scenarios. However, it does not work well on our dataset because microscope images have significantly different dataset distributions comparing to ImageNet. Therefore, we defrost the first several layers as the strategy to cope with the issue and it contributes an extra 2% mAP for Faster RCNN training.

To improve the model performance, tuning hyper-parameters is a key step. Among various hyper-parameters, we found RPN anchor settings and recalculated pixel means boost the Faster RCNN significantly from 0.40 to 0.60 mAP. RPN anchor scales are set as [30, 40, 50] and [0.4, 0.6, 0.8, 1.0] for aspect ratios, based on the distributions of instances. Those settings provide a closing initial status for bounding box regression in both RPN and task head. Hence, the decreased regression difficulty contributes the mAP improvement.

3.3.5 Training Mask RCNN with Decorated Annotations

The core part of pore measuring is to segment the stoma pore outline as a polygon (open pore) or a line (closed pore), which is conducted by the Mask RCNN (Figure 3.4) on the transformed single stoma images. The transformed dataset contains 837 single stoma images split as the training set (738 images/instances) and validation set (99 images/instances). The length and width of open stomatal pore have a large variance, leading to many different aspect ratios. To be specific, the length (131.31 pixels in average) vary from 68 to 181 and the width are ranged from 1.673 to 169.0 with the mean value of 30.04. The normal operation of directly segmenting those small objects failed even with fine-tuned RPN anchor settings and produced very low mAP, with few barely predicted segmentation for closed stomatal pores with low confidence (less than 10%). Similarly, the overfitting model reaches 0.821 mAP, resulting in the unbalanced detection that most of the closed stomatal pore detection failed. Comparing with the architecture modifications in related works Singh and Davis (2018); Lin et al. (2017a); Liu et al. (2016); Yi et al. (2018), we attempted to address this small object detection issue in a different way by "decorating annotations".

3.3.5.1 Decorating Annotations

Decorating annotations (closed pores) is a procedure adding extra vertices into annotations of closed pores to transfer the lines to rectangles. The transformation is implemented by vertically enlarging the shape by a small value as the "width" of the closed stomatal pores and keeping the length unchanged. This value is given as the mean of the first quartile of all open stomata pore widths, which is 8.3116 pixels in our case. The assigned width is much smaller compared with the mean open stomatal width (30.04 pixels), which has no significant effects on deciding the pore category. Therefore, these decorations make detecting closed stomatal pore detection an easier problem while preserving the target length features.

3.3.5.2 Overall Training

The Mask RCNN is trained for 20K iterations with the default settings and produced acceptable results as in Figure 3.6. The inference time with CPU is averaged around 8.0 seconds per image, while GPU decreases the time cost to less than 2.0 seconds. Regarding the mAP metrics, the performances for the small and medium-size instances are considered, especially for closed pores. For small objects (mainly the closed stomatal pores), our Mask RCNN model achieved 0.437 mAP for pore outline segmentation and 0.457mAP for the pore bounding box detection. For the medium objects (mainly the open stomatal pores), the performance is 0.551 mAP on segmentation and 0.521 mAP on bounding box detection. As those metrics alone are not sufficient to reflect the

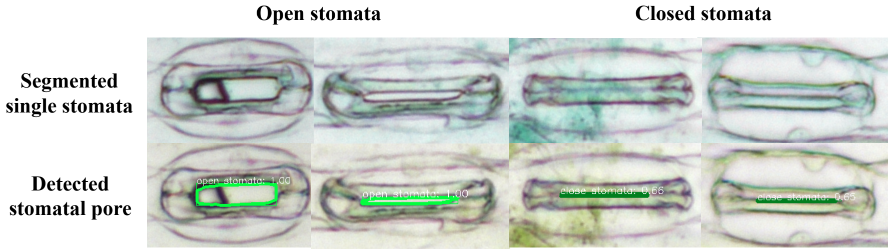


FIGURE 3.6: **The examples of segmented single stoma and predicted stomatal pores.** *Top:* segmented single stoma based on Faster RCNN predictions. *Bottom:* predicted polygon segmentation by Mask RCNN (light green for open stomata and dark green for closed stomata). These predictions are processed for further measurement.

performance, our model contributes to the precise measurements of open pores with high inference confidence (as in Figure 3.6) and more details are presented in Section 3.3.6.

3.3.6 Area and Length: Measuring Stomata

By performing the measuring algorithms on the predicted open stomatal pores, our system has shown accurate results with a high confidence score for the area and width, shown in Figure 3.7. The area and width present 95.03% and 95.05% of average precision respectively. To be specific, with over the 601 open pores, StomaRCNN has detected 596, 549, 533 filtered with 50%, 75% and 90% confidence respectively. Regarding measurement precision, linear regression represented with R square of 0.9889 for width and 0.9915 for area measurement. In conclusion, the lightweight StomaRCNN achieved manual-level precision automatically. For closed stomata, the morphological structure of closed stomata, makes it challenging even for manual measurement as it is visually difficult to distinguish. Therefore, the prediction results are produced with less confidence and relatively lower accuracy. As a consequence, the measurement is less competitive than manual processing.

Since training large models on the small dataset can easily lead to overfitting, our backbone ResNet uses several regularisation methods such as batch normalisation and residual connections. For Mask RCNN training, mAP is not sufficient to decide the trained model for measurement as the high mAP (>0.80) normally leads to the failure of segmenting closed stomatal pores. The final model is selected with a balanced trade-off between mAP and the visualised segmentation results. Additionally, the single stoma images for Mask RCNN training are in small scales and we tuned the minimal input image size as 800 pixels (against 160, 320 and 1000) to enlarge the small images. Besides, lower starting learning rates (0.001) and decay step size are important.

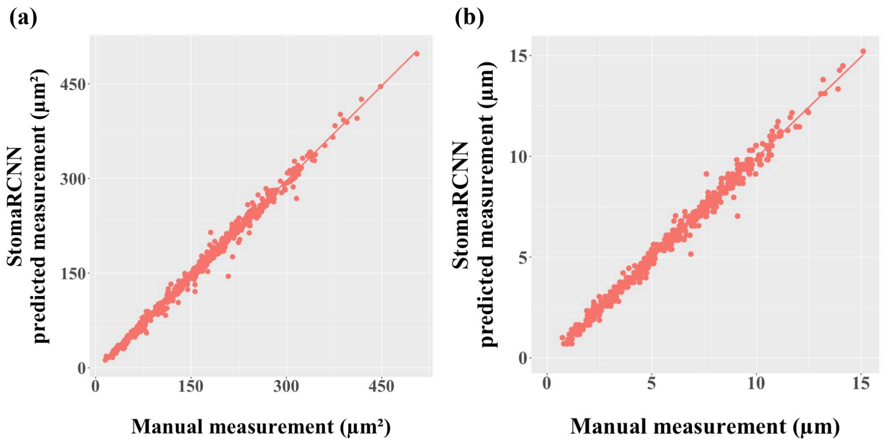


FIGURE 3.7: **Manual Measurement vs StomaRCNN Predicted Measurement with over 90% confidence.** Each data point represents an open stoma and fit in linear regression. The StomaRCNN produced precise measurement for area **(a)** and width **(b)** of stomatal pore, with R^2 of 0.9889 and 0.9915 respectively.

3.4 Reproduction Notes

This section summarises the key components to reproduce the reported results in this chapter. A brief technical introduction on deploying StomaRCNN is provided. For the deployment concerns, I've summarised an instruction in Appendix A, which is initially used to guide my collaborator deploying the system on their server at Adelaide University, Australia.

3.4.1 Data Split

The collected dataset contains **92** images with **841** stomata objects. Among these 841 objects, **604** ($\approx$ 71.8% of total objects) are the open stomata and **237** ($\approx$ 28.2% of total objects) are the closed ones. Considering the balance between learning and evaluation, around 90% data are randomly split as the training set, leaving the rest 10% as the testing set, which is not used for training. The "9:1" split is determined by the total image amount and the practical concerns. The details of splitting dataset are shown below.

The training set contains **82** images, which is $\approx$ 89.1% of total images. It includes **757** objects, which is $\approx$ 90.0% of total objects. Among those objects, there are **549** open stomata and **192** closed ones. For the testing set, it includes **10** images, which is $\approx$ 10.9% of total images. Those testing images contains **84** objects which is $\approx$ 9.9% of total objects. Among those objects, **55** are open and **45** are closed stomata.

Comparing with many large datasets, our Stomata dataset is small. The key reason is the time-consuming procedures for plant scientists to conduct one experiment. Their key steps include growing the target from the seed (normally one or two weeks), test with different chemical approaches, metabolic activity observation, select valid leaves and microscopy imaging.

Considering the procedure of model selection, the general idea is using small models or handcraft feature extractors such as HOG-based filters. The main reason for avoiding deep learning models is that DNNs are easy to be overfitting when the dataset is small. However, this work selects RCNN models not only by the reasons aforementioned in previous sections but also by the stomata data. Unlike general public datasets such as MS-COCO and ImageNet, our dataset contains only stomata images with open and closed pores. More importantly, from the aspect of plant science, stomata have the biological consistencies. In other words, there is a very limited variety of the stomata shapes (square, rectangle but never a triangle), especially for the barley species. Therefore, DNN models are selected for stomata measurement with their deep learning features hold, such as transfer learning, model reusing, end-to-end, and so on. These advantages make DNNs fit better than the traditional handcraft feature extractors such as HOG-based models.

3.4.2 Anchor-based Model Selection

As aforementioned above, this project selects Faster RCNN and Mask RCNN models for detection and segmentation. The key reason for selecting RCNN models is their anchor mechanisms, which ensured a better performance than those classic one-stage anchor-free models. Actually, the anchors in Faster and Mask RCNN are fixed with location, scales, and aspect ratios. Instead of these anchor-fixed models, there is a great raising trend of models using “trainable anchors” after we finished this project. The key idea is changing the fixed anchors to a “learnable” and “differentiable” mechanism, in order to guide the detection network to produce better performance. It is necessary to discuss the pros and cons of replacing Faster and Mask RCNN models with these newly proposed anchor-free models.

Fixed-Anchors: As the architectures of Faster and Mask RCNN is demonstrated in the previous sections, the detection and segmentation relay on “anchors, which are the fixed locations in the “Region Proposal Network” (see Figure 3.3). The detection proposals are firstly generated based on the fixed anchor at the fixed location, then adjusted to a different location at different scales and aspect ratios. The proposal adjusting procedures are handled by the RCNN model, trained in a bounding box regression fashion. In other words, Faster RCNN and Mask RCNN models are trained to learn the differences between the fixed anchors and the ground truth. Given different input feature maps, the models output their predictions by adjusting the pre-fixed bounding box proposals

Table 3.2: Stomatal aspect ratio and scales

Stomatal Aspect Ratios	Max: 2.6364 (width: 58, height: 22) Min: 1.1136 (width: 49, height: 44) Mean: 1.6706
Stomatal Scales (in pixels)	Max: 2730.0 (width: 65, height: 42) Min: 950.0 (width: 50, height: 19) Mean: 1757.8942

to a different location, scale, and aspect ratios. To summarise, anchors in Faster and Mask RCNN have the fixed bounding box locations, scales, and aspect ratios. These pre-fixed anchors are fine-tuned by the model to produce the predictions.

Trainable-Anchors: the key drawback of the fixed-anchor mechanism is manually tuning the configures for the fixed anchors, which makes it less adaptive to different tasks and difficult to regress with inappropriate hyper-parameters. To address this issue, researchers have proposed several novel works on “trainable” anchors. Yang et al. (2018) proposed a “MetaAnchor Algorithm” to learn optimal anchor location and settings by minimising the customised “anchor box prior” loss. Their work is based on RetinaNet Lin et al. (2017b) and it outperformed the baseline RetinaNet on MS-COCO dataset. Wang et al. (2019) proposed a novel region proposal design combining the advantages of dynamic anchors and recent anchor-free models. It first determines the object centroid to avoid redundant anchors in irrelevant locations. Then the anchors are trained with the guidance of the object centroid. Their mechanism outperformed many baselines such as Faster RCNN and RetinaNet, on MS-COCO dataset. Overall, “trainable anchor” mechanisms avoid manually setting anchors, which are more robust and adaptive when the dataset contains large variations on contexts and instances.

Considering this recent progress with this stomata measurement project, it is important to highlight that the stomata dataset is less complicated than MS-COCO, which is the main dataset that “trainable anchors” succeed. Another drawback is that training anchors take extra time and extra resources, especially when the fixed anchor settings can be well-determined. In this case, stomata are biologically consistent to similar species, which means anchor location, scale, and aspect ratios can be well initialised by precomputation even the dataset is small. Table 3.2 has summarised stomata scale (the annotated bounding box size) and aspect ratio from the dataset, which is given by

$$Aspect\ Ratio = \frac{Horizontal\ Width}{Vertical\ Height}. \tag{3.1}$$

TABLE 3.3: Faster RCNN anchor setting tests. The chosen candidates covers most stomatal scales and aspect ratios, as in Table 3.2.

Aspect Ratio	Scales	mAP
[0.2, 0.5, 1.0]	[32, 64]	0.496
[0.4, 0.6, 0.8, 1.0]	[30, 40, 50]	**0.596**

TABLE 3.4: Anchor Settings for Stomata Detection (Faster RCNN) and Stomatal Pore Segmentation (Mask RCNN)

	Anchor Aspect Ratios	Anchor Scales (pixels)	Sensitive to the Settings?
Faster RCNN	[0.4, 0.6, 0.8, 1.0]	[30, 40, 50]	Sensitive
Mask RCNN	[0.5, 1.0, 2.0]	[0.5, 1.0, 2.0]	Less sensitive

Anchor scales and aspect ratios are the two key settings for the fixed anchors, which are guided by the summarised information in Table 3.2. In order to speed up anchor box regression and improve the performance, the anchors should be initialised close to the majority of stomata. Besides, the anchor amount in each location is given by the multiplication of the number of configured scales and aspect ratios. More anchor scales and aspect ratios fit the data well but gain more parameters with increasing resources at the same time. While few settings make the detection more difficult, even zero valid proposals in RPN. Therefore, tuning anchor hyper-parameters requires a trade-off between performance and the associated budget. After testing two candidate settings as presented in Table 3.3, the anchors for Faster RCNN and Mask RCNN are shown in Table 3.4. Mask RCNN is less sensitive to the settings due to the input images are the cropped sub-images, which are re-scaled before feeding into Mask RCNN. Besides, these sub-images are relatively simpler as each one contains only one stoma with one stomatal pore inside it.

In conclude, recent progress on "trainable anchors" has raised promising potentials to improve the model performance and current fixed-anchor RCNN models are still competitive, as long as the anchor hyper-parameters can be appropriately determined, given by the stomata dataset.

3.4.3 Optimisation and Hyper-parameter Settings

Overall Settings: both Faster and Mask RCNN are trained with the Stochastic Gradient Descent (SGD) optimiser. Extra regularisation methods are applied to avoid overfitting, including batch normalisation Ioffe and Szegedy (2015) in the backbone CNN model (ResNet 50), weight decay and learning rate decay during backpropagation. The original images are normalised before feeding into RCNN models, by subtracting the mean pixel values in each of the RGB channel (calculated from the training set). The detailed settings are presented in Table 3.5. The rest of this section demonstrates the

TABLE 3.5: Hyper-parameter settings for Faster and Mask RCNN

	Faster RCNN	**Mask RCNN**
Optimiser	SGD	SGD
Momentum	0.9	0.9
Weight Decay	0.0001	0.0001
Initial Learning Rate	0.0001	0.001
Learning Rate Decay (every certain epochs)	5,000	3,000 and 7,000
Total Training Epochs	10,000	10,000
Batch Norm Settings	σ: 0.997 ϵ: 0.00001	σ: 0.997 ϵ: 0.00001
Image Normalisation (mean RBG pixel values)	[214.1339, 212.7212, 204.2712]	[193.1126,190.7024,183.7082]

TABLE 3.6: Backbone Model Settings

Model Pre-trained	Weight Frozen	mAP (Faster RCNN)
✗	N/A	0.496
✓	✗	0.575
✓	✓	**0.597**

motivation and parameter tuning process for these hyper-parameters.

Backbone Model Settings: for the backbone ResNet50 model, there is a concern of freezing weights (the first two of four ResNet blocks) which assumes the pre-trained model is capable for capturing positional features of the stomata dataset. The potential benefit of freezing backbone models is reducing training time (i.e. fewer parameters to tune during backpropagation) and maintain the performance. While the negative concern is the large domain shift. The pre-trained models come from ImageNet which is quite different the stomata dataset so that the pre-training model may not differ much with the untrained ones. To verify this potential bonus, several experiments are conducted to investigate whether the pre-trained model better than randomly initialised one, and whether the weight frozen helps the detection performance, as shown in Table 3.6. As indicated in the experimental results, the pre-trained model with "weight frozen" outperformed other variations. Therefore, the ImageNet pre-trained backbone model (ResNet 50) is applied with the first two residual blocks frozen.

Optimisation and Regularisation Settings: As aforementioned, StomaRCNN uses Stochastic Gradient Descent for training both Faster and Mask RCNN, which is firstly proposed by Robbins and Monro (1951). However, there are some other alternatives such as the popular Adam optimiser proposed by Kingma and Ba (2014). The principle advantage for Adam is its adaptivity to the data in different magnitudes. Adam also warps the mechanism of momentum and dimension-wise learning rate from AdaDelta

(Zeiler (2012)). However, machine learning is a data-driven area that algorithmic performances vary from different tasks (i.e. dataset). Besides, SGD with momentum and weight decay provides great improvements comparing with the solely SGD, making such combination widely used in object detection, and segmentation tasks. In practice, like many other models, the SGD-based optimiser produces higher mAP than Adam. The key reason is the different dataset making the adaptive optimisation approach less effective. Therefore, StomaRCNN uses the SGD-based optimiser with detailed settings in Table 3.5.

Avoiding Overfitting: As discussed above, stomata data naturally maintain biological consistencies. For training the RCNN models, many methods have been applied to avoid overfitting issue, such as batch normalisation applied in the backbone CNN models. Besides, weight decay is applied to make the trained weights sparse so that less overfitting. Image flipping is also used to augment extra images, which contributes less overfitting as well. For regarding the RCNN models, they are naturally applied the “smooth l1” loss (Girshick, 2015b; Ren et al., 2015a; He et al., 2017), which is a soft version for the classic L1 norm loss. The smooth l1 loss replaced the linear with the exponential form, as in the equation below:

$$Smooth\ L_1 = \begin{cases} 0.5x^2 & if\ \ |x| < 1 \\ |x| - 0.5 & otherwise \end{cases} \tag{3.2}$$

Comparing with L2 norm, L1 norm is robust to outliers, such as the abnormal bounding box locations. Because the gradients of the L1 norm are scalar, which is independent of x's value. The smooth L1 loss actually changed the gradients when x value is small ($|x| < 1$). Therefore, these decreased gradients in small x make the system more robust, especially RCNNs normally calculate losses for thousands of bounding box proposals.

Learning Rate Decay Settings: learning rate decay is the commonly used method to fine-tune models. It decreases the learning rate when the training reaches the configured epochs. In other words, the model is firstly trained with a larger learning rate to reach the local optimal in the hyper-space. Then learning rate is decreased to a smaller value which fine-tunes the model to the local optimal. In practice, the learning rate decay value is conducted by multiplying a small number (0.0001 in this project). This scalar is given by RCNN default settings and works well in many computer vision tasks, and this project too. However, the epoch setting for conducting learning rate decay is not that straightforward. The epoch setting actually balances the length between the rough training and the fine-tuning phases. Table 3.7 shows a group of candidates for the epoch settings. Therefore, the final weight decay epoch for Faster RCNN is 5,000 given total epoch 10,000, which equally splits the rough training and fine-tuning phases. Mask RCNN follows the default 3,000 and 7,000 epoch setting, since its input

Table 3.7: Learning Rate Decay Setting Selection for Faster RCNN

Total Epochs	Decay Epoch	mAP
10,000	5,000	**0.597**
10,000	3,000	0.565
10,000	2,000	0.584
10,000	500	0.525
50,000	35,000	0.534
20,000	15,000	0.518
1,000	700	0.407
500	300	0.229

images are relatively simple. The reason for the two-step learning rate decay in Mask RCNN is that segmentation requires more fine-tune epochs, as it conducts precise pixel-wise recognition. In other words, not only the Region Proposal Network (as in Faster RCNN) requires fine-tune, but also the segmentation CNN branches (Figure 3.4).

Initial Learning Rate Settings: the initial learning rate is set according to input data magnitudes, by which it avoids exceptional gradients and associated NaN errors during training. Since the data normalisation is applied to reduce the pixel values, three key learning rates are tested: 0.01, 0.001, 0.0001, and 0.00001. For StomaRCNN, 0.01 and 0.001 directly failed after a few epochs with NaN errors. Training with the smallest 0.00001 is much slower than 0.0001. Therefore, Faster RCNN learning rate is initialised as 0.0001, and Mask RCNN is set as 0.001 in a similar way.

Image Normalisation: as in many computer vision tasks, input images are usually normalised by subtracting the mean pixel values in each of the RGB channels. The main reason is reducing input scales to decrease the difficulties of hyper-parameter tuning. Many default hyper-parameters are empirical set on the dataset which have the similar magnitudes, so that these settings can be directly transferred with after normalising the images. Another reason is to avoid unnecessary errors such as NaN gradients caused by too large loss value. Because pixel values (ranged from 0 to 255) are much larger than ground truth labels (normally less than 100). In StomaRCNN, the mean pixel values are calculated from the training set and directly used for normalising the test data. Despite the small dataset, the stomatal biological consistencies and similar microscope equipment ensure the calculated values good suitability for test images. In practice, mean pixel values [214.1339, 212.7212, 204.2712] are used for Faster RCNN, and [193.1126,190.7024,183.7082] are applied in the Mask RCNN model. The normalisation values are also tested with freshly produced raw microscope images, and it is correctly functioning.

The instruction of system deployment is presented in Appendix A.

3.5 Conclusion and Future Works

We propose StomaRCNN to automatically measure the stomatal pores with precise results and high confidence. Especially for open stomatal pores, our proposed system achieved human-level accurate measurement with less inference time. Therefore, StomaRCNN is expected to dramatically shorten the measuring time for biologists. Comparing to common approaches in Section 3.1, the proposed system reduced computational costs while preserving the detailed information.

Even with a small amount of data, our system achieved the desired behaviours in both mAP and measurement without overfitting. StomaRCNN has the expected generalised flexibility among different species since the morphological structure of stomata are highly reserved in cross-species and the image resolution is fixed because of the bounded microscope lenses. To address the small instance segmentation, our proposed method of decorating annotations indicates potential approaches to decrease the difficulties while preserving the target features in a simple way. Besides, StomaRCNN utilises the plugin plugin-out pipeline enabling various model compositions not limited to DNNs. Adapting impressive approaches in face recognition Wu et al. (2018); Hu and Ramanan (2017); Dong et al. (2018) is another focus of the future work to improve the closed pore measurement.

Chapter 4

Conclusions

This thesis has presented two different approaches of improving object detection performances in the aspects of image augmentation and DNN pipeline modification. The slot-based image augmentation system demonstrates the positive effects of generating extra images and large potential possibilities of extending it for many other object detection related challenges such as extracting scale-invariant features for small object detection. Moreover, it observes detailed performance in category-wise and highlights the importance of "requirement oriented" augmentation. StomaRCNN demonstrates a "task-splitting" pipeline of measuring instances in high-resolution images. In this work, detection and segmentation are distributed to two individual models due to the different requirements on instance details. For the architecture design, the end-to-end pipeline enables replacements on different models. Furthermore, details of original images are preserved and utilised for precise measurement tasks, avoiding unnecessary image detail losses.

To summarise, object detection is a fundamental area for many advanced computer vision tasks and the improvements in detection performance have great effects on related domains. Instead of DNN architectural modification, this thesis provides two lightweight approaches to address the performance issue. All in all, the deep learning based approaches are data-driven systems extracting features from training data and guided by loss function and optimisation algorithms. Besides the DNN architectures, it is worthy to recall the significance of data itself and more efforts are required to rethink the relations among images, instances and related features, since the commonly used augmentation method does not always helps detection performance. In addition, the pipeline modification draws insights on reviewing the object detection related to DNN architecture designs.

Appendix A

StomaRCNN: Deployment Guide

A.1 Overview

The procedures of deploying and running StomaRCNN following general, basic and simple deep learning pipelines, highlighted in the following key steps:

1. System Working Procedures
2. Pre-requirements Installation
3. Deploying and local-building codes
4. Testing installation by running the prediction demo
5. Training with our Current Data
6. New Data Preparation, Formating, Integrating with previous ones
7. Train with own/updated Data
8. Evaluate and Understand the Performance
9. Tricks and Values specifically for training StomaRCNN

A.2 System Working Procedures

StomaRCNN is designed to detect stomata location, cropping stomata from original images, segment stomatal pores and calculate weights, heights and areas by geometric calculations.

1. Image Preparation: Scripts and RectLabel[1]
2. Stomata Detection: Faster RCNN (Pytorch)
3. Stomata Cropping: Python Scripts
4. Stomata Data re-Formatting: Python Scripts based on PANDAS
5. Stomatal Pore Segmentation: Mask RCNN (Pytorch)

[1] https://rectlabel.com/

6. Stomatal Pore Calculation: Python Scripts

A.3 Data Preparation and Formatting

- Annotation and Images are organised in an MS-COCO[2] fashion.
  - MS-COCO style follows a certain architecture: ”dictionary-and-lists”. Please refer to sample annotation files for details.
- The data consist of .png Images and .json Annotations.
- Annotation files are named as ”COCO”+ ”train/val” to reduce unnecessary workload.
  - It’s recommended to give them meaningful names afterwards. (remained works)

A.4 Stomata Detection with Faster RCNN (Pytorch)

There are many popular repositories on the Internet. The author used Ruotian Luo’s version when there were not any good other choices. It’s now not maintained and feel free to install other recommended versions, in Python 3. I’ve prepared part of the code analysis UML style graph here.

The key procedures are summarised as the followings[3]:

1. **Install Prerequisites:** follow this link
2. Download and Compile: *git clone* and *Make* the repository, link.
3. **Install Data API:** Install Python COCO API, link.
4. **Deploy StomaRCNN Detection Codes and Demo Data:** copy and replace StomaRCNN version faster RCNN library files, configuration files and others, link.
5. **Visualise Detection Results:** for visualisation, refer the Jupyter Notebook link. There is also an completed script to run detection models on small images, crop from original large images and automatically generate annotation files, link. It’s recommended to use GPU to do this job, see this link.
6. **Run the codes:** you can run the demo with Jupyter Notebook to test your installation, such as *customised_visualisation*. To train the network, run *./experiments/scripts/train_faster_rcnn.sh* in terminal. You can configure the training details inside the shell script. Refer this link, and change the dataset as I pre-defined ”stomata” and the trained model is res50. It looks like: *./train_faster_rcnn.sh 0 stomata res50*

[2] https://github.com/cocodataset/cocoapi

[3] part of the links cannot be presented, due to the confidential reasons.

Note: Double-check the file path and folder path in the code. The relative path is suggested.

A.5 Stomatal Pore Segmentation with Mask RCNN (Pytorch)

Mask RCNN is used to segment cropped sub-images which contains and only contains one single stoma (and pore). Associated with the sub-images, MaskRCNN is trained with generated annotation files, automatically transformed from the previous step (see *overall_integrated_pipeline.ipynb*). I've prepared part of the code analysis graph, here.

The MaskRCNN model is based on Facebook's Benchmark version. You can validate your installation by running the scripts, such as this one.

Again, there are many more efficient segmentation models coming up. Therefore, feel free to replace MaskRCNN with a more advanced model. As usual, you can always optimise the deployment using libraries to prune/trim and quantize DNNs, which are popular in Tensorflow and Tensorflow Lite on Google Cloud ecosystem. I'm not experienced in the Pytorh one, but do check this Quantization.

Here are the principle pipelines to deploy and run the codes. **Again, I recommend optimising the deployment...**

1. **Benchmark Installation in a benchmark way:** a simple attempt is using *conda* as instructed here. It also provides tips on Docker.
2. Replace files with our customised codes.
3. Run the demo to visualise the pre-trained model.
4. Then you are ready to train your own model by updating the configure files.

A.6 Stomatal Pores Measuring and Output

The previous step has produced the outputs and ready for measurement. Please use an integrated script to run the segmentation, measurement and saving results as .csv files. Please check this link.

Bibliography

Alexandre Aono, James Nagai, Gabriella Dickel, Rafaela Marinho, Paulo Oliveira, and Fabio Faria. A stomata classification and detection system in microscope images of maize cultivars. bioRxiv, 2019.

Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. Machine learning, 79(1-2):151–175, 2010.

Dominique C Bergmann and Fred D Sack. Stomatal development. Annu. Rev. Plant Biol., 58:163–181, 2007.

G. Bradski. The OpenCV Library. Dr. Dobb's Journal of Software Tools, 2000.

Chenyi Chen, Ari Seff, Alain Kornhauser, and Jianxiong Xiao. Deepdriving: Learning affordance for direct perception in autonomous driving. In The IEEE International Conference on Computer Vision (ICCV), December 2015a.

Tianqi Chen, Mu Li, Yutian Li, Min Lin, Naiyan Wang, Minjie Wang, Tianjun Xiao, Bing Xu, Chiyuan Zhang, and Zheng Zhang. Mxnet: A flexible and efficient machine learning library for heterogeneous distributed systems. arXiv preprint arXiv:1512.01274, 2015b.

Xinlei Chen and Abhinav Gupta. An implementation of faster rcnn with study for region sampling. arXiv preprint arXiv:1702.02138, 2017.

Zhe Chen, Shaoli Huang, and Dacheng Tao. Context refinement for object detection. In The European Conference on Computer Vision (ECCV), September 2018.

Zhong-Hua Chen, Guang Chen, Fei Dai, Yizhou Wang, Adrian Hills, Yong-Ling Ruan, Guoping Zhang, Peter J Franks, Eviatar Nevo, and Michael R Blatt. Molecular evolution of grass stomata. Trends in plant science, 22(2):124–139, 2017.

Yan Cheng, Ling Cao, Sheng Wang, Yongpeng Li, Hong Wang, and Yongming Zhou. Analyses of plant leaf cell size, density and number, as well as trichome number using cell counter plugin. Bio-protocol, 4:13, 2014.

Corinna Cortes and Vladimir Vapnik. Support-vector networks. Machine learning, 20 (3):273–297, 1995.

Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation policies from data. arXiv preprint arXiv:1805.09501, 2018.

Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In Proceedings of the IEEE international conference on computer vision, pages 764–773, 2017.

Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In international Conference on computer vision & Pattern Recognition (CVPR'05), volume 1, pages 886–893. IEEE Computer Society, 2005.

J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In CVPR09, 2009.

Xuanyi Dong, Yan Yan, Wanli Ouyang, and Yi Yang. Style aggregated network for facial landmark detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 379–388, 2018.

Nikita Dvornik, Julien Mairal, and Cordelia Schmid. Modeling visual context is key to augmenting object detection datasets. In Proceedings of the European Conference on Computer Vision (ECCV), pages 364–380, 2018.

Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. International journal of computer vision, 88(2):303–338, 2010.

Karl Fetter, Sven Eberhardt, Rich S Barclay, Scott Wing, and Stephen R Keller. Stomatacounter: a deep learning method applied to automatic stomatal identification and counting. BioRxiv, page 327494, 2018.

Sean Gillies et al. Shapely: manipulation and analysis of geometric objects, 2007–.

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

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

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.

Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In Proceedings of the IEEE international conference on computer vision, pages 2961–2969, 2017.

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.

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

Alistair M Hetherington and F Ian Woodward. The role of stomata in sensing and driving environmental change. Nature, 424(6951):901, 2003.

Daniel Ho, Eric Liang, Ion Stoica, Pieter Abbeel, and Xi Chen. Population based augmentation: Efficient learning of augmentation policy schedules. arXiv preprint arXiv:1905.05393, 2019.

Derek Hoiem, Yodsawalai Chodpathumwan, and Qieyun Dai. Diagnosing error in object detectors. In European conference on computer vision, pages 340–353. Springer, 2012.

Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. Neural networks, 2(5):359–366, 1989.

Md Sazzad Hossain, Andrew P Paplinski, and John M Betts. Adaptive class weight based dual focal loss for improved semantic segmentation. arXiv preprint arXiv:1909.11932, 2019.

Han Hu, Jiayuan Gu, Zheng Zhang, Jifeng Dai, and Yichen Wei. Relation networks for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 3588–3597, 2018.

Peiyun Hu and Deva Ramanan. Finding tiny faces. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), July 2017.

John D Hunter. Matplotlib: A 2d graphics environment. Computing in science & engineering, 9(3):90, 2007.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167, 2015.

Paul Gordon Jarvis and Terence Arthur Mansfield. Stomatal physiology, volume 8. Cambridge University Press, 1981.

Hiranya Jayakody, Scarlett Liu, Mark Whitty, and Paul Petrie. Microscope image based fully automated stomata detection and pore measurement method for grapevines. Plant methods, 13(1):94, 2017.

Jeesoo Kim, Jangho Kim, Jaeyoung Yoo, Daesik Kim, and Nojun Kwak. Vehicle image generation going well with the surroundings. arXiv preprint arXiv:1807.02925, 2018.

Tae-Houn Kim, Maik Böhmer, Honghong Hu, Noriyuki Nishimura, and Julian I. Schroeder. Guard cell signal transduction network: Advances in understanding abscisic acid, co2, and ca2+ signaling. Annual Review of Plant Biology, 61(1):561–591, 2010.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980, 2014.

M.B. Kirkham. Chapter 24 - stomatal anatomy and stomatal resistance. In M.B. Kirkham, editor, Principles of Soil and Plant Water Relations (Second Edition), pages 431 – 451. Academic Press, Boston, second edition edition, 2014.

Mate Kisantal, Zbigniew Wojna, Jakub Murawski, Jacek Naruniec, and Kyunghyun Cho. Augmentation for small object detection. CoRR, abs/1902.07296, 2019.

Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems, pages 1097–1105, 2012.

Hamid Laga, Fahimeh Shahinnia, and Delphine Fleury. Image-based plant stomata phenotyping. 12 2014.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. nature, 521(7553): 436, 2015.

Yann LeCun and Corinna Cortes. MNIST handwritten digit database. 2010.

Hankyeol Lee, SEOK EON CHOI, and Changick Kim. A memory model based on the siamese network for long-term tracking. In European Conference on Computer Vision Workshop. Springer, 2018.

Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 2117–2125, 2017a.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollar. Focal loss for dense object detection. In 2017 IEEE International Conference on Computer Vision (ICCV), pages 2999–3007. IEEE, 2017b.

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. Springer, 2014.

Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia. Path aggregation network for instance segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 8759–8768, 2018.

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. Springer, 2016.

Francisco Massa and Ross Girshick. maskrcnn-benchmark: Fast, modular reference implementation of Instance Segmentation and Object Detection algorithms in PyTorch. https://github.com/facebookresearch/maskrcnn-benchmark, 2018. Accessed: 28/04/2019.

Wes McKinney. pandas: a foundational python library for data analysis and statistics. Python for High Performance and Scientific Computing, 14, 2011.

Deirdre H McLachlan, Michaela Kopischke, and Silke Robatzek. Gate control: guard cell regulation by microbial stress. New Phytologist, 203(4):1049–1063, 2014.

Maeli Melotto, William Underwood, and Sheng Yang He. Role of stomata in plant innate immunity and foliar bacterial diseases. Annu. Rev. Phytopathol., 46:101–122, 2008.

Aude Oliva and Antonio Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. International journal of computer vision, 42(3): 145–175, 2001.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. In NIPS-W, 2017.

Alexander J Ratner, Henry Ehrenberg, Zeshan Hussain, Jared Dunnmon, and Christopher Ré. Learning to compose domain-specific transformations for data augmentation. In Advances in neural information processing systems, pages 3236–3246, 2017.

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.

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

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In Advances in neural information processing systems, pages 91–99, 2015a.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In Advances in neural information processing systems, pages 91–99, 2015b.

Herbert Robbins and Sutton Monro. A stochastic approximation method. The annals of mathematical statistics, pages 400–407, 1951.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In International Conference on Medical image computing and computer-assisted intervention, pages 234–241. Springer, 2015.

Gerard Salton and Michael J McGill. Introduction to modern information retrieval. 1986.

Philip Saponaro, Wayne Treible, Abhishek Kolagunda, Timothy Chaya, Jeffrey Caplan, Chandra Kambhamettu, and Randall Wisser. Deepxscope: segmenting microscopy images with a deep neural network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pages 91–98, 2017.

Johannes Schindelin, Ignacio Arganda-Carreras, Erwin Frise, Verena Kaynig, Mark Longair, Tobias Pietzsch, Stephan Preibisch, Curtis Rueden, Stephan Saalfeld, Benjamin Schmid, et al. Fiji: an open-source platform for biological-image analysis. Nature methods, 9(7):676, 2012.

Julian I Schroeder, Gethyn J Allen, Veronique Hugouvieux, June M Kwak, and David Waner. Guard cell signal transduction. Annual Review of Plant Physiology and Plant Molecular Biology, 52(1):627–658, 2001.

Samuel Schulter, Paul Vernaza, Wongun Choi, and Manmohan Chandraker. Deep network flow for multi-object tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 6951–6960, 2017.

Lei Shen, Peng Sun, Verity C. Bonnell, Keith J. Edwards, Alistair M. Hetherington, Martin R. McAinsh, and Michael R. Roberts. Measuring stress signaling responses of stomata in isolated epidermis of graminaceous species. Frontiers in Plant Science, 6: 533, 2015.

Ken-ichiro Shimazaki, Michio Doi, Sarah M Assmann, and Toshinori Kinoshita. Light regulation of stomatal movement. Annu. Rev. Plant Biol., 58:219–247, 2007.

Bharat Singh and Larry S Davis. An analysis of scale invariance in object detection snip. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 3578–3587, 2018.

Rajiv Kumar Sinha. Modern plant physiology. CRC Press, 2004.

Yosuke Toda, Shigeo Toh, Gildas Bourdais, Silke Robatzek, Dan Maclean, and Toshinori Kinoshita. Deepstomata: Facial recognition technology for automated stomatal aperture measurement. bioRxiv, page 365098, 2018.

Engin Tola, Vincent Lepetit, and Pascal Fua. Daisy: An efficient dense descriptor applied to wide-baseline stereo. IEEE transactions on pattern analysis and machine intelligence, 32(5):815–830, 2010.

Stefan Van Der Walt, S Chris Colbert, and Gael Varoquaux. The numpy array: a structure for efficient numerical computation. Computing in Science & Engineering, 13(2):22, 2011.

Stefan Van der Walt, Johannes L Schönberger, Juan Nunez-Iglesias, François Boulogne, Joshua D Warner, Neil Yager, Emmanuelle Gouillart, and Tony Yu. scikit-image: image processing in python. PeerJ, 2:e453, 2014.

Jiaqi Wang, Kai Chen, Shuo Yang, Chen Change Loy, and Dahua Lin. Region proposal by guided anchoring. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 2965–2974, 2019.

Colin Willmer and Mark Fricker. Stomata, volume 2. Springer Science & Business Media, 1996.

Wayne Wu, Chen Qian, Shuo Yang, Quan Wang, Yici Cai, and Qiang Zhou. Look at boundary: A boundary-aware face alignment algorithm. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 2129–2138, 2018.

Tong Yang, Xiangyu Zhang, Zeming Li, Wenqiang Zhang, and Jian Sun. Metaanchor: Learning to detect objects with customized anchors. In Advances in Neural Information Processing Systems, pages 320–330, 2018.

Jingru Yi, Pengxiang Wu, Menglin Jiang, Daniel J. Hoeppner, and Dimitris N. Metaxas. Instance segmentation of neural cells. In The European Conference on Computer Vision (ECCV) Workshops, September 2018.

Matthew D Zeiler. Adadelta: an adaptive learning rate method. arXiv preprint arXiv:1212.5701, 2012.

Hengshuang Zhao, Xiaohui Shen, Zhe Lin, Kalyan Sunkavalli, Brian Price, and Jiaya Jia. Compositing-aware image search. In The European Conference on Computer Vision (ECCV), September 2018.

Hubert Ziegler. The evolution of stomata. Stomatal function, 29:57, 1987.

Barret Zoph, Ekin D Cubuk, Golnaz Ghiasi, Tsung-Yi Lin, Jonathon Shlens, and Quoc V Le. Learning data augmentation strategies for object detection. arXiv preprint arXiv:1906.11172, 2019.