An Automated System for 3D Modelling and Feature Extraction of Small-Scale Objects, Combining Computer Vision and Deep Learning

Hengjia Li
U5629478

Supervised by Dr Chuong Nguyen and Dr Marnie Shaw

June 2019

A thesis submitted in part fulfilment of the degree of
Bachelor of Engineering
Department of Engineering
Australian National University

This thesis contains no material which has been accepted for the award of any other degree or diploma in any university. To the best of the author's knowledge, it contains no material previously published or written by another person, except where due reference is made in the text.

Hengjia Li

Hengjia Li
30 May 2019

Acknowledgements

This thesis has been the work of the whole year and throughout the journey, I have learnt how to establish individual research and to think about fundamental questions in Computer Vision and Deep Learning. I still remember the first time I met with Dr Chuong Nguyen, my mentor and supervisor, and he provided me with the choices of two projects, which were a 3D modelling project and a CNN-based feature detection project. At that time, I had never imagined that I could carry on both projects and make progress in each. Now, when I look back into the journey, I am grateful because this work will never be achieved without the help and encouragement of many people, whom I would like to thank and acknowledge here.

First, I would like to thank both of my supervisors, Dr Chuong Nguyen and Dr Marnie Shaw. They not only offer me the chance to work on these projects but also provide me with their patience to support my research. We have frequent meetings and they are always ready and patient for my questions. I would like to thank their encouragement on my current research and my future research opportunities,

To my fellow honours students, Minh Doan, Yilin Geng, Hongjian He, José Quiroga and Andrea Bedón Pineda, a huge thanks for their support and being a part of my university life. A special thanks to Yuetong Chen for her taking care of me and bringing me a wonderful time at Canberra.

I am deeply grateful to my family and beloved parents for their sacrifices. As an international student, I am studying away from home. I am sorry for making my parents worry but finally, I am about to finish my degree and it will be the time that I can go home.

It has been a joy to study computer vison and deep learning. I hope that you also can feel its beauty in reading this paper.

Abstract

Digitisation and 3D modelling of natural history collections are important for indexing and archiving biodiversity (Hudsn et al., 2015). Existed 3D modelling devices require user input to provide semantic-level information about the target specimen. Here we established a multi-view imaging system with a Bayesian inference framework to automate this process. The system aims to capture multi-view all-in-focus images by applying an extended-depth-of-field technique and the captured images can be reconstructed into 3D models by 3D reconstruction software. The Bayesian inference framework enables the system to detect the target features of the images automatically. We use the CUB-200-2011 dataset as a proof-of-concept dataset to validate the framework. Example outputs from image blending and feature detection are demonstrated. The goal of this research was to determine which image blending method was more applicable for the multi-view image capturing device and what main factors constrain the feature prediction performance. We further provide potential solutions to help improve the performance of image blending and feature detection.

CONTENTS

List of Figures

List of Tables

Abbreviations and acronyms

CNN	Convolutional Neural Network
EDOF	Extended Depth of Field
CT	Computed Tomography
SFS	Structure from silhouette
SFM	Structure from Motion
CAM	Class Activation Mapping
Grad-CAM	Gradient-weighted Class Activation Mapping
EOS	Electro Optic Systems
DSLR	Digital Single-Lens Reflex
DLT	Direct Linear Transform
LoG	Laplacian of Gaussian
VGG	Visual Geometry Group
ILSVRC	ImageNet Large Scale Visual Recognition Challenge
TF	Term Frequency
IDF	Inverse Term Frequency
PCK	Percentage of Correct Key-points

Chapter 1 Introduction

Digitisation and 3D modelling of natural history collections have become a significant challenge in indexing and archiving biodiversity (Hudsn et al., 2015). Among zoological collections, insects are unparalleled in their number of species and specimens (Ströbel, et al., 2018). Thus, automated image acquisition and modelling are important for large-scale digitisation projects. Due to the small-scale structure of insect bodies, 3D modelling of insects requires images to be taken with an extended depth of field (EDOF) technique (Brecko, et al., 2014), which applies focus stacking to generate an all-in-focus image and to extend the shallow depth of field. Several commercially available software solutions allow EDOF calculation by post-processing focus stack images, and there are also some cameras have inbuilt EDOF functions (Brecko, et al., 2014). However, digitisation of small-scale objects requires multi-view imaging, which is a highly-repeatable process. The commercial software solutions become time-consuming because they cannot reuse information on camera position into repeated EDOF calculation. As a result, large-scale digitisation projects apply multi-view images and EDOF calculations in only one or two viewing directions if necessary, although multi-view EDOF images offer not only digital access to morphological characters of the specimen, they can also be reconstructed into coloured and textured 3D models (Mathys, et al., 2013; Mathys, et al., 2015; Nguyen, et al., 2014).

Traditionally, researchers focus on 1-D descriptions (e.g., body width, length, or body ratios between parts) for characterising species. However, natural body and organisms have 3D shapes, and 2D or 3D traits that are also meaningful as complementary information for characterisation (Tatsuta et al., 2017). As a result, biodiversity research increasingly focuses on functional traits in 2D or 3D structures of species rather than considering only on species identities or numbers (Loreau, et al., 2001; Mouillot, et al., 2012). A trend is that the latest image dataset contains not only labels for classification purposes, but also locations of essential 2D or 3D structures as “ground truth”. For example, CUB-200-2011 dataset (Welinder, et al., 2011) contains 11,788 bird images of 200 different species, and it offers 15 part locations annotated by pixel location of each bird image. This dataset benefits research on attribute-based and part-based methods, but it requires many user indications to collect the “ground truth” part locations. In this scenario, the application of visual attribute grounding is a solution for reducing the workforce. Visual attribute grounding is an established method for locating target 2D or 3D attributes from images, by utilising a pre-trained convolutional neural network (CNN) model and a Bayesian Inference framework.

In this paper, we combine two research topics together, which are a 3D modelling system and a CNN-based approach for visual attribute grounding. The combination of these topics forms a system that automatically captures high-resolution EDOF images of small-scale objects and analyses attribute locations of interest. There are two functionalities of this system:

- **Multi-view image capturing** aims to produce all-in-focus images of small insect bodies by applying EDOF techniques. An image-based image capturing device is established for implementing focus stacking. Central to the system is an algorithm to calibrate and blend stack images. Two different blending strategies are applied, that is the saliency measurement method and guided filtering approach. Furthermore, the device can perform 3D modelling of target specimens by feeding captured EDOF images into 3D reconstruction software. *The goal of my research here was to determine which method was more applicable for the multi-view image capturing device for the purpose of 3D modelling.*

- **Visual attribute grounding** aims to indicate the target attributes from an input image. The CUB-200-2011 dataset of bird images is applied as a training and testing target, as there is no well-labelled insect dataset providing part locations. Central to the algorithm is a Bayesian Inference framework that investigates relationships between the last convolutional layer filters and the visual attributes of the image. *The goal of this research was to determine what main factors constrain the feature prediction performance.*

The main contribution of this system can be summarised as follows:

- We employ a readily available image-based 3D modelling device with repeatable EDOF techniques. It will benefit trait-based research areas in functional ecology, ecophysiology and evolutionary biology.
- We propose the novel combination of the 3D modelling device and target feature detection. It can potentially reduce the workload involved in archiving and indexing natural collections.

Chapter 2 Related Works

3D DIGITISATION AND RECONSTRUCTION

Standard 3D modelling and reconstruction techniques have mainly two classes, which are using X-ray computed tomography and using image-based computer vision methods.

Digitisation by X-ray Computed Tomography (CT) aims to capture landmark data with high precision. However, this technology is not able to capture colour/texture information of the object's surface, and it usually requires costly equipment or limited beamtime at suitable synchrotron facilities. (Ijiri, et al., 2018) presented a 3D digitisation experiment for natural objects. They reconstruct a 3D model by segmenting the CT volumetric images and then they project the digital photographs of the target body onto the 3D model. Their method successfully digitalises physical specimens in 3D models containing internal structure and external texture information, but the applications of both X-ray CT device and digital camera are much more expensive and time-consuming than simple image-based 3D modelling systems.

Image-based 3D Digitisation usually requires a high-magnification lens for capturing the millimetre-scale structure. To overcome the shallow depth of field of the lens, a stack of images is captured at different distances from the target specimen and then merged into one single extended depth of field (EDOF) image (Brecko et al., 2014, Nguyen, et al., 2014). Such a blended EDOF image contains all information of the target specimen for 3D reconstruction, and this image blending process is known as Focus Stacking (Ströbel et al., 2018).

By applying triangulation and 3D reconstruction techniques, multiple EDOF images taken from virtually any view compose the expected 3D model. Typical reconstruction methods include 'Shape from Silhouette' (SFS) and 'Structure from Motion' (SFM). SFS (Franco & Boyer, 2003) (Xiang et al., 2016) extracts the silhouette of the object surface based on viewing rays tangential to the object. It computes a virtual volume from silhouettes at many viewing angles and generates a 3D visual hull to approximate the shape of the actual object. However, it cannot capture indentations on the target surface because there are no silhouette rays from those parts. One solution is photo-consistency, but it is only applicable when there is no strong specular reflection (Haro, 2014). Moreover, SFS requires scaled markers included in the reconstruction images for calibration. A common approach is to pin the target insect specimen on a dot matrix target, but it usually requires adding an auxiliary second pin or to re-pin the insects, which may permanently damage the specimen. SFM (Westoby et al., 2012, Bolles, et al., 1987, Brostow, et al., 2008) identifies features points on target surface from different viewing directions, and it does a simultaneous calibration according to these detected features points. Hence there is no need for a pre-calibration of camera positions or a re-pining process. This method requires a number of well-textured feature points that occur in overlapping areas from each image and it is able to reconstruct concave surfaces. Currently commercialized SFM software include Agisoft PhtoScan, 3DSOM, and open source modules like Alice Vision and Visual SFM.

Existing image-based 3D reconstruction systems include the 3D nature-colour modelling device (Nguyen et al., 2014) and DISC3D (Ströbel et al., 2018), as shown in Figure 2.1. These two systems are designed for generating EDOF images and therefore, they are both equipped with a macro rail controlled by a computer or built-in controller to capture images at different distances to the target specimen. Motorized two-axis turntable rotates and tilts the specimen at any desired angle of view. DISC3D uses two hemispherical illumination domes (a 'front-light' dome on the side of the camera and a 'back-light dome' on the far side facing the camera). Instead of a point-shaped or ring-shaped illumination source (e.g., camera ring flashlight), each dome is designed to produce scattering and nearly homogeneous illumination to the specimen. Furthermore, the back-light dome can generate a uniform background illumination for the camera to capture a silhouette image of the insect, which can be useful for cropping out the insect body from the EDOF background.

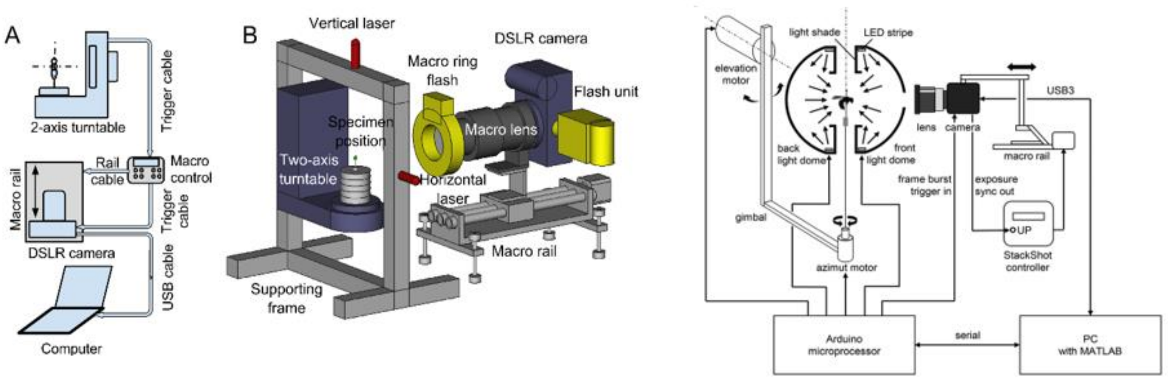


Figure 2. 1 Left: 3D Nature-Colour Capturing Device (Nguyen, et al., 2014). Right: DISC3D (Ströbel, et al., 2018)

Multi-scale Image Blending is an essential step for EDOF image-based 3D modelling. As discussed previously, multiple stack images of the same viewing direction can be merged into a single EDOF image, which preserves enough comprehensive and detailed target features for 3D reconstruction (Brecko et al., 2014). The raw images obtained from image stacking are usually out of registration, and it requires sort of pre-calibration process for the Image-based 3D reconstruction systems (Nguyen et al., 2014, Ströbel, et al., 2018). Due to Camera vibration during movement, it causes random image translation and some image blending method that is robust to random translation will be more applicable for this 3D modelling experiment. A saliency measurement algorithm (Ströbel et al., 2018) based on spatial consistency is proposed for image blending, and it can generate high quality blending result with texture information well preserved. However, it may suffer from an edge-shaped distortion when the calibration error exists. Another proposed method for image blending, named as Guided Filtering approach (Li et al., 2013, He, et al., 2013), has been proved to be more robust to image translation. Guided filtering enables the blending process to be referenced, and it reduces the effect of misaligned edges. In the following sections of this paper, both methods will be implemented and discussed based on their blending outputs.

FINE-GRAINED RECOGNITION AND VISUAL ATTRIBUTE GROUNDING

Network Interpretation To understand the "black box" of the neural network, two main approaches are filter-level interpretation and holistic-level interpretation. The former method aims to understand features that specific neurons learn, by visualising convolutional filter's weights. For some shallow and simple network structures, it is exceptionally easy to extract essential features or patterns directing the network's decisions by filter-level interpretation. However, deeper and more complicated network structures increase the difficulty of visualisation of filter weights, since those deep filters usually act as complex composite functions of other shallower filters. Some recent studies of filter-level interpretation include up-convolutional neural representations (Dosoviskiy & Brox, 2015), which can reconstruct target images from feature maps of conv-layer filters. However, this up-convolutional neural network cannot mathematically prove that visualisation results are related to actual neural activations. (Bau, et al., 2017) proposed a method of network dissection, which computes each neuron unit's interpretability according to pre-defined annotations such as texture and colour. This method mathematically defines the interpretability of each filter, and it is a closer step for understanding the knowledge of the CNN black box. Furthermore, (Zhang et al., 2018) proposed an Explanatory Graph model to further disentangle object-part pattern components from each filter without any object annotations. These previous researches perform well in the visualisation of neuron-level behaviours, but they are insufficient to interpret the network's decisions. On the contrary, holistic-level interpretation can provide a layer-level explanation of the network, and it is sometimes more efficient to reveal the network's decision-making process globally.

A common practice of holistic-level interpretation is to visualise the critical subregions of the input image by re-weighting conv-layer feature maps, for example, CAM (Zhou et al., 2015) and Grad-CAM (Selvaraju et al., 2016). They combine the last conv-layer feature maps to produce a localisation map, which highlights the critical sub-region in the image of prediction. These two applications can provide coarse-level information but not semantic-level explanations. For example, they cannot explain how each attribute or feature of the input image influences the prediction result. See Figure 2.2 for examples.

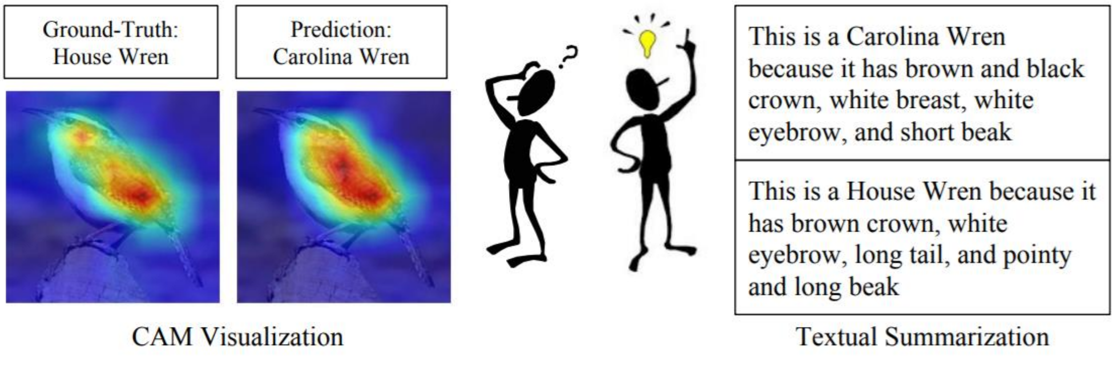

Ground-Truth: House Wren
Prediction: Carolina Wren
This is a Carolina Wren because it has brown and black crown, white breast, white eyebrow, and short beak
This is a House Wren because it has brown crown, white eyebrow, long tail, and pointy and long beak
CAM Visualization
Textual Summarization

Figure 2. 2: Visualization based interpretation (left) and Semantic-level textual Summarization (Right). The former can indicate important image region for network's decision, but the latter can provide much more sematic explanation. (Guo, et al., 2018)

Visual Attribute Grounding is another attempt at holistic summarization, and it aims to highlight the region of high-level semantic features (e.g., orange breast or belly as shown in Figure 2.3) in the target images. (Zhang, et al., 2017) proved that a graphical model named explanatory graph successfully highlights attributes of interest in an image, for example breast region of a bird. However, their method is not applicable for grounding attributes carrying high information entropy, such as colour information (e.g. orange breast region of a bird or black beak region of a bird). Comparing to this explanatory group method, a Bayesian inference algorithm (Guo et al., 2018) is proposed to extract attributes carrying high information entropy. It establishes a paired Filter|Attribute relationship by Bayesian inference and this algorithm will be implemented and analysed in this research.

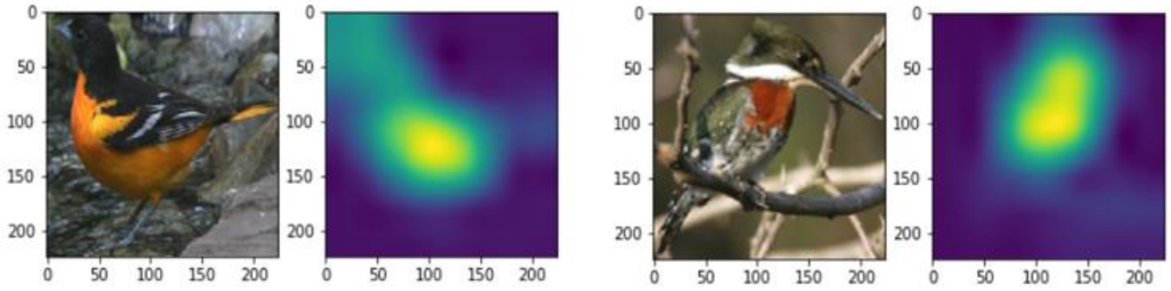


Figure 2. 3: Examples of Visual Attribute Grounding. The image masks indicate the region of orange belly (left) and red chest (right) of target birds.

Residual Network Model Studies (Simonyan & Zisserman, 2014) have revealed that the network depth is of the crucial importance of network performance (Simonyan & Zisserman, 2014) (Loffe & Szegedy, 2015). The significance of network depth raises a question: can a network model can be optimised by simply stacking more layers? A degradation problem (He et al., 2015) (Srivastava et al., 2015) has been proposed and it shows that the prediction accuracy becomes saturated and then degrades rapidly while the network depth is increasing, see Figure 2.4 left for an example. This degradation problem demonstrates that not all networks or systems are similarly easy to optimise**.** A deep residual learning framework is introduced by (He et al., 2015) to address the degradation problem. They add "shortcut connections" (Figure 2.4 right) between different conv-layers in order to skip one or more layers. These shortcut connections simply map the identical input "x" to the output of the stacked layers. Therefore, they let the network learn the residual mapping $(F(x) + x)$ instead of theoriginally desired mapping $(F(x))$ and they conclude that this residual mapping is easier to be optimized than the original unreferenced mapping.

An *"identity mapping"* example can explain why this residual model works. Let us consider two network architectures, which are a shallower architecture and a deeper architecture. The deeper architecture is just the shallower architecture with some more layers stacked on the top. If the stacked layers perform identity mapping, then theoretically the deeper model should generate no higher training error than the shallower model. However, experiments show that a network with these stacked identity layers cannot produce better training performance, and it may even generate higher training errors. This result reveals that the stacked layers cannot be perfectly trained to perform identity mapping. The residual architecture solves this identity mapping constraint by pushing the residual of stacked layer outputs to zero (pushing "$F(x) + x$" to *"x"*, in Figure 2.4). Reducing the value of residuals is easier than straightforwardly fitting an identity mapping by a stack of nonlinear layers. As a result, the residual network can be trained to perform identity mapping and it can avoid negative outcomes when increasing network depth.

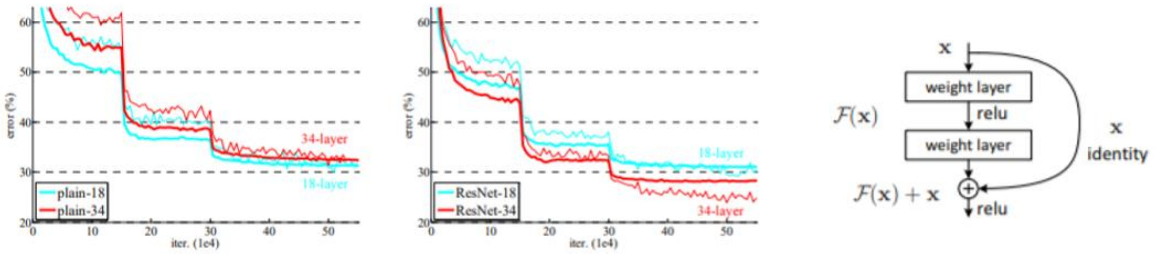


Figure 2. 4: Training on ImageNet. Left: training and validation error on plain networks and Resnet models of 18 and 34 layers. It can be observed that increasing the depth of Resnet model results in a lower error rate. Right: a building block of Residual learning (He et al.)

Chapter 3 Method and Experiment

3.1 PART 1. MULTI-VIEW IMAGE COLLECTION

In this research, a multi-view imaging system is developed for capturing image stacks of target specimen. The raw images obtained from this system are pre-calibrated and blended to generate all-in-focus EDOF images. Different blending methods are tested, and their performances are discussed.

3.1.1 Experiment Setup

We implemented our 3D modelling System, as shown in Figure 3.1. Such a system contains:

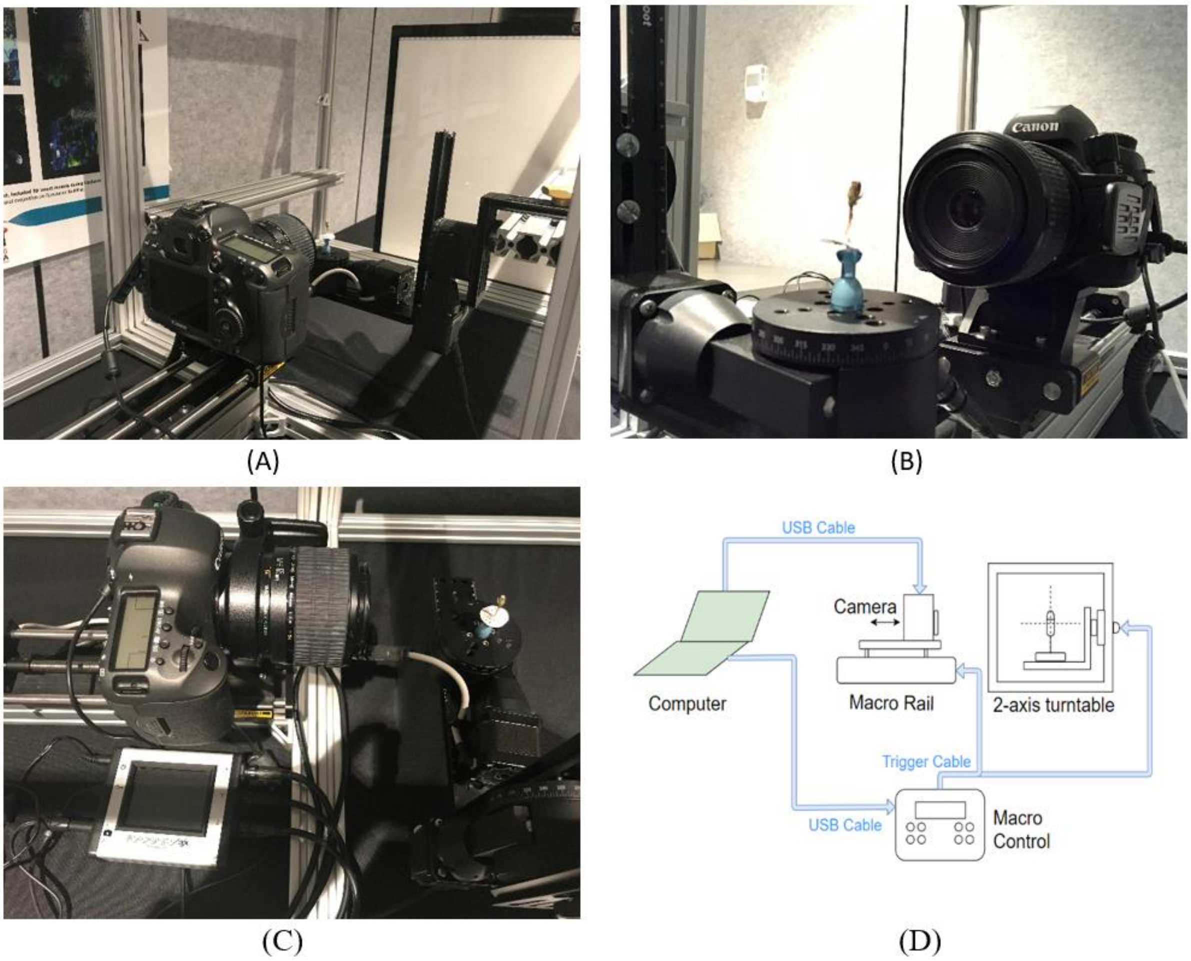


Figure 3. 1 Experiment setup. Figure (A), (B) and (C): pictures of the 3D modeling system. Such a system contains sveral core components: a DSLR camera with a macro lens, a moterized macro-rail, two rotatory tables and a inbuilt controller. Figure (D): schematic setup

- A Canon EOS DSLR camera with a macro lens to capture high-magnification images
- A motorised macro-rail to adjust the distance between the camera and the specimen
- Two rotatory tables to adjust the view angle of the specimen by two-axis rotation
- A StackShot controller to control the motion of the macro-rail and the rotatory tables.
- A PC connected to the camera and the controller to implement focus stacking and image blending algorithms
- A macro ring flash to provide front-side illumination

- A LED lighting board to provide background illumination
- Two laser pointers to indicate the centre of the two-axis rotation

A rigid box frame covers the entire system, and it avoids any accidental touch to the specimen and camera. Furthermore, it creates a dark room with light-proof panels, such that the scan is not affected by ambient illumination. Another advantage of such a design is that the position of the specimen can be easily adjusted for fitting with different camera models.

3.1.2 Image Acquisition

Due to the shallow depth-of-field of a macro lens, the target specimen is usually partially at focus but is blurry at other regions. To extend the depth-of-field of the captured images, focus stacking method is applied to blend several partially-focused images into an all-in-focus image. The Cognisys Stackshot macro rail system enables an axial motion of the camera moving toward the specimen. The range of the axial motion R is predefined for the scan to cover entire insect body throughout the imaging process and it can be obtained as $R \geq \max(w)$, where $\max(w)$ denotes the maximal width of target specimen at different viewing angles. The forward motion starts at where the camera begins to capture a sharp image for any component of the specimen and the number of steps of focus stacking is determined by $\mathrm{N} = \frac{R}{f}$, where f denotes the focal depth of the lens.

3.1.2.1 Calibration and Image Registration

The partially-focused stack images need to be registered to ensure that all the target objects in images are within the correct frame of perspective. Thus, the focus stacking process is pre-calibrated. Feature detection and matching (e.g., S.I.T.F.) is an efficient method for calibration. However, the camera's motion has high repeatability, and it is a waste to run a feature detection algorithm on each captured image. Thus, a MATLAB calibration algorithm is developed to compute the homography and geometric transform between images.

A flat dot matrix target (see Figure 3.2 A as an example), featuring a rectangular grid of 7×5 circular black marks on a white background, is placed at the rotatory table (same position as insect specimen for scanning). This dot matrix is treated just as a scanning target. Therefore, stack images are captured at each step of the camera's motion toward the target, just as the same process in insect scan. In order to compute the geometric transform between each image, the following key steps are performed:

1) Load dot matrix images into MATLAB.
2) Detect circular patterns in images. Circular Hough transform is applied to detect the circles in the image. The sensitivity parameter of the transform accumulator array is set to be 0.99, which detects as many circles as possible from the image. To avoid redundant or failure detections, the radius of circles is predefined in a range of 20 to 40 pixels. As shown in Figure 3.2 B, the location and radius of detected circles are returned.
3) Delete redundant detections. Redundant detection occasionally appears, as shown in Figure 3.2 B. Euclidean distance between detected circles are compared, and the redundant ones can be located if they are too close to each other. From circular Hough transform, the

detections can be ranked according to their strength in the Hough domain, and the weak ones are deleted. Therefore, the redundant detections are removed, and there are precisely 7 × 5 detections remained, see figure 3.2 C for an example.

4) Match circle-pairs. The detected circular marks are paired up from different images. Again, the Euclidean distance is computed for any two detected circle from two images. The nearest circle centre indicates a pair of circles pointing to the same circular mark.

5) Compute the geometric transformation matrix. The last image captured (the closest one to the specimen) in the stack is defined as a reference image. Direct Linear Transform (DLT) algorithm is applied to compute the homography estimation, which warps other target images to the reference image. Let us say we want to warp the circular marks in the i-th image (x_i) to the reference image (x_r), therefore it defines

$$X_r = HX_i \tag{1}$$

or equivalently,

$$X_r \times HX_i = 0 \tag{2}$$

where H is the homography matrix. In homogeneous coordinate,

$$X_r = \begin{bmatrix} x_r \\ y_r \\ w_r \end{bmatrix} \tag{3}$$

$$HX_i = \begin{bmatrix} {h_1}^T x_i \\ {h_2}^T y_i \\ {h_3}^T w_i \end{bmatrix} \tag{4}$$

$$X_r \times HX_i = \begin{bmatrix} y_r {h_3}^T w_i - w_r {h_2}^T y_i \\ w_r {h_1}^T x_i - x_r {h_3}^T w_i \\ x_r {h_2}^T y_i - y_r {h_1}^T x_i \end{bmatrix} \tag{5}$$

From the above equations, if we factorise the unknowns, it becomes

$$\begin{bmatrix} 0 & -w_r y_i^T & y_r w_i^T \\ w_r {x_i}^T & 0 & -x_r w_i^T \\ -y_r {x_i}^T & x_{r y_i^T} & 0 \end{bmatrix} \begin{bmatrix} h_1 \\ h_2 \\ h_3 \end{bmatrix} = 0 \tag{6}$$

Equivalently,

$$A_i H = 0 \tag{7}$$

There are only two rows of A are linearly independent because there are two degrees of freedom for each 2D point. In other words, a 2 × 9 matrix A_i is computed for each pair of detected circles. Since there are 7 × 5 pairs of circles, a 30 × 9 matrix A is formed to solve for the homograph H. Singular value decomposition is applied to matrix A and H is determined by the right-most column of V.

6) Visually check the calibration result. The homography matrix is applied to stack images, and they are warped to the reference plane (the one closest to the specimen). Therefore, these black circular marks from different images are registered and each of them is transformed to the same perspective for superimposition. Figure 3.2 D visualises this process. The green dots are the target of warping from the reference image and the red ones

are from another stack image to be warped. The tiny red arrow indicates the corresponding geometric transformation

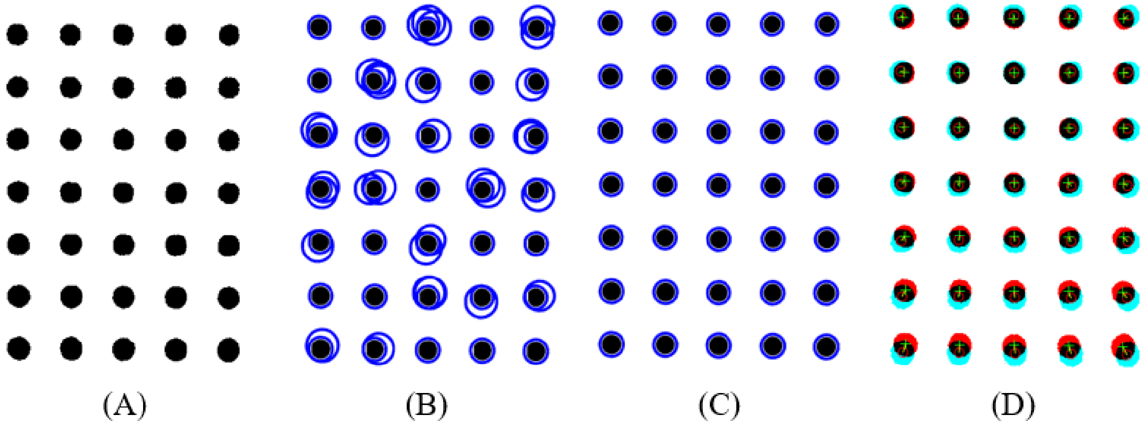


Figure 3. 2: Examples to describe the calibration process: (A) image of calibration target: black circular marks; (B) detected circles with redundancy; (C) detected circles without redundancy; (D) visualization of geometric transformation: the red circles are transformed into the green circles by implementing homography estimation.

3.1.2.2 Focus stacking and image blending by saliency measurement

After registering the raw stack images, a saliency blending algorithm is applied to generate the final EDOF output. This algorithm implements an additive weighted superposition, and the weights are measured from the local sharpness/saliency of the images. The sharp region in the registered images implies a high value of local saliency. The final EDOF output can be obtained by merging the sharp areas from all stack images. The following key steps explain this algorithm in detail:

1) Capture and load a stack of the target image. Down-sampling of input images may be performed due to time limitation. In this experiment, a dimension of at least 1080p (1920 x1080) is acceptable. See Figure 3.3 (A) and (C) for example stack images.

2) Load Homography matrix and register the stack images.

3) Compute local saliency for each image by applying a Laplace of Gaussian (LoG) filter. The filter parameters (filter size, standard deviation) can be adjusted concerning different brightness and dimension of images. Figure 3.3 B and D demonstrate the different focused regions and different distributions of sharpness of two random stack images

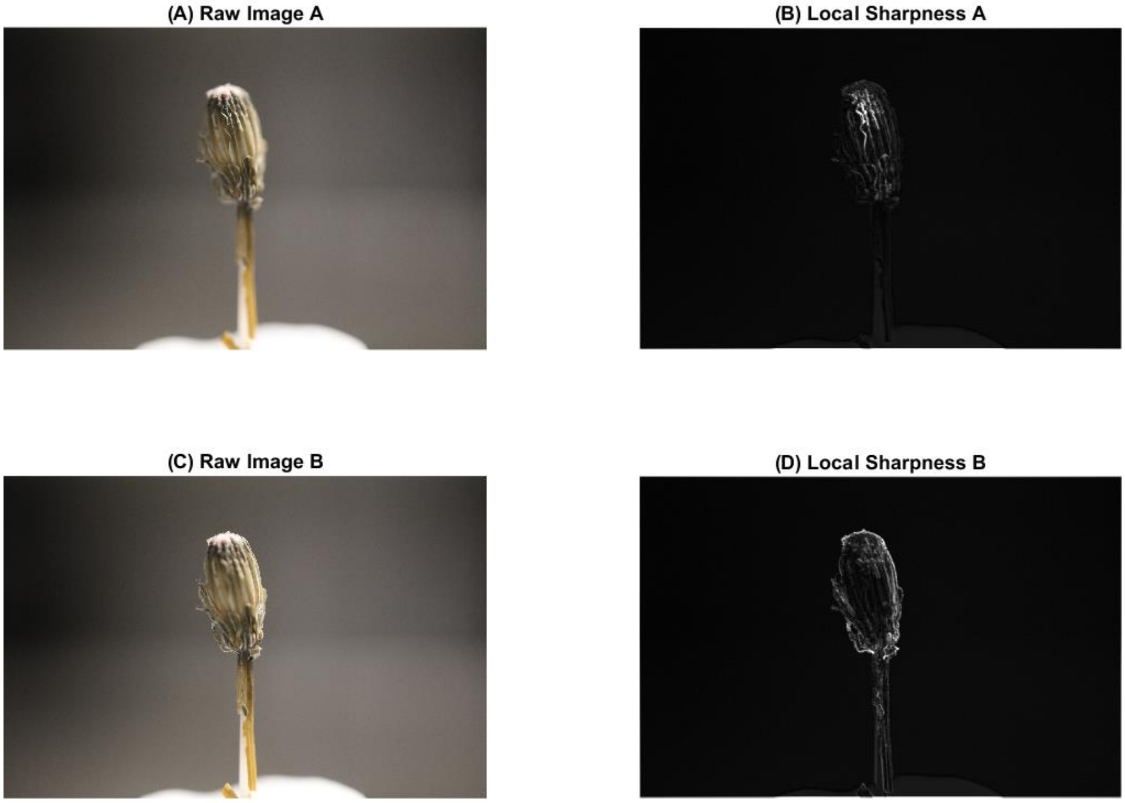


Figure 3. 3: Examples of Local Sharpness Detection of image A and B, which have different region at focus. Left hand side: image A and B. Right hand side: detected saliency of A and B . It can be observed that Image A is focused at front region of the specimen, but Image B is focused at edge contours of the specimen.

4) Perform weighted sum of stack images. The final blended image (F) can be obtained as

$$F = \sum_{i=N} W_i * Img_i \tag{8}$$

where Img_i is the i-th stack image and W_i is a weight matrix correspondingly. By performing element-wise production, the weight matrix determines the superimposition of blending and it can be computed by:

$$W_i = \frac{Local\ Sharpness_i{}^P - min_i}{\sum_{i=N} Local\ Sharpness_i{}^P} \tag{9}$$

where i denotes the n-th image and N denotes the total number of images in a stack. min_i denotes the minimum sharpness of each image and it is usually zero in our case. P indicates an amplifying factor, and it allows for different degrees of averaging between stack images. A higher value of P tends to generate a sharper EDOF images, which strictly follows the local sharpness detection but may result in some undesirable edge-shaped distortion. A lower value of P tends to produce an averaged blending result, which may be slightly blurred. Figure 3.4 reveals the effects of different values of P. The stack sharpness plots (left hand side) indicate the global sharpness of a whole stack of images, and the stacked image plots (right hand side) display the final blending result.

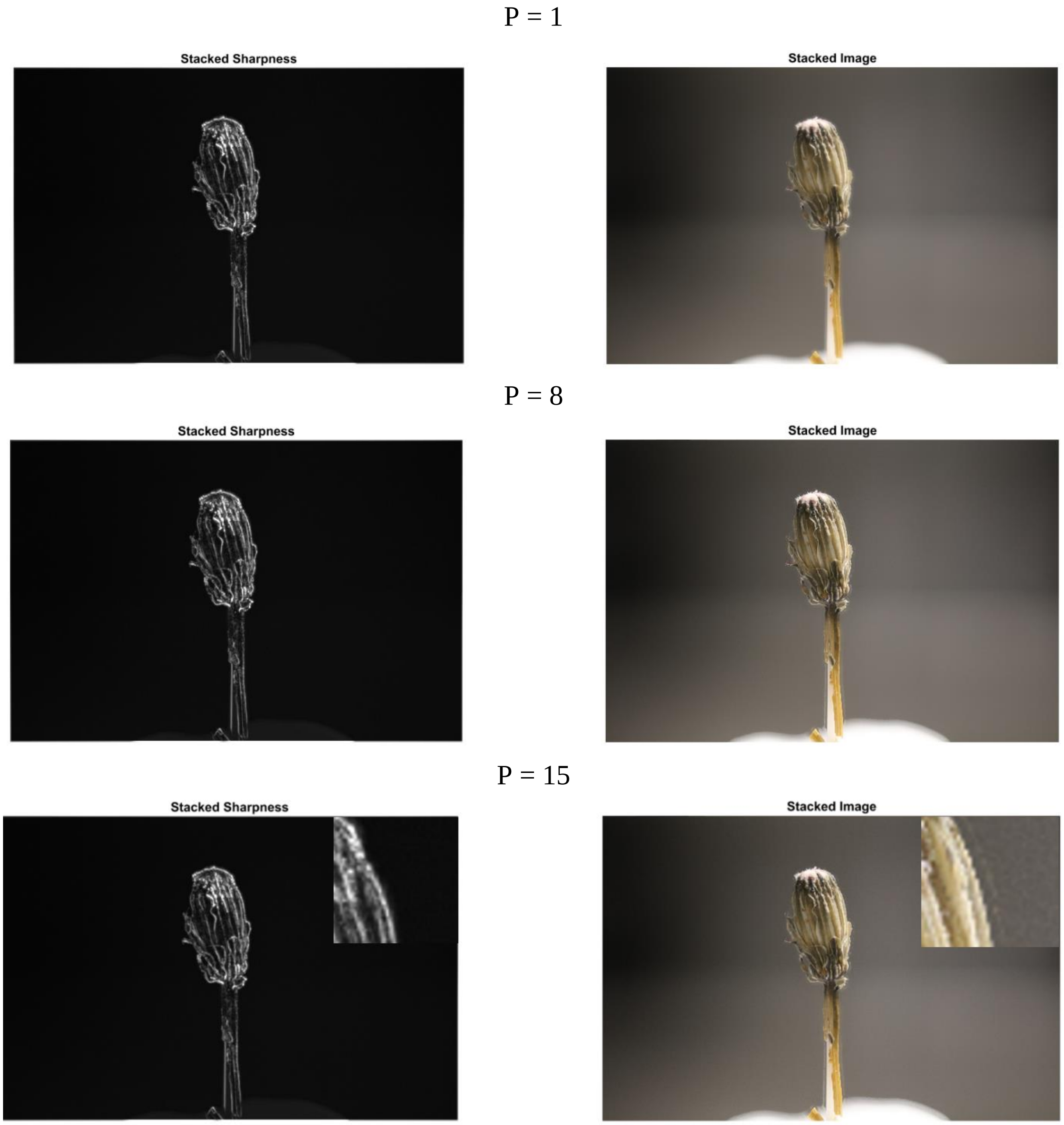


Figure 3. 4: Comparison of blending result from ten stack images with different P values. Stacked Sharpness(left): the maximal saliency at each pixel, among all stack images. Stacked Image(right): Blending results. From top to bottom, the sharpness of stack image gets stronger but “edge-shaped” noise (a white edge around the target object, see zoomed-in region of P=15 for an example) is also enhanced.

3.1.2.3 Image Blending with Guided Filter method

Similarly, the stack images are first registered according to the calibrated homography. Then, a guided filter based image blending algorithm is applied to merge the registered images. Unlike the saliency measurement method introduced in the previous section, this algorithm decomposes each stack images into a two-scale representation and then fuse these representations by applying a weighted average method with guided filters. The following sections will introduce the concept of the guided filter and then, it will explain how this filter can be applied for the image blending purpose.

- **Guided image filtering**

The guided filter is efficient while dealing with small-scale objects (e.g., insect or plant specimen) because of its edge-preserving feature, which avoids blurring sharp edges in the decomposition process. This filter takes two images as input, which are a guidance Image I and an input image P. The filter will go through the input image and enhance its' details according to the guidance image, see Figure 3.5 for an example.

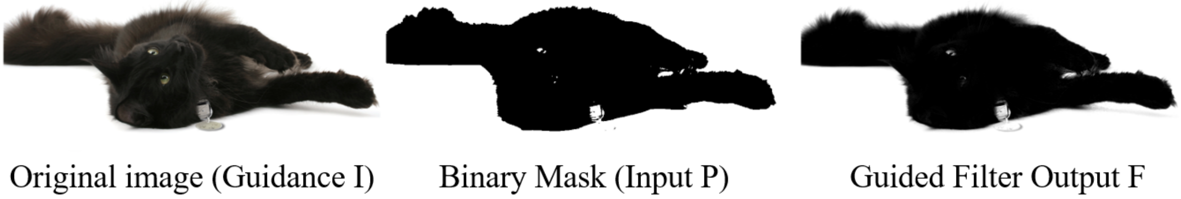


Figure 3. 5 Example output of Guided Filtering. From left to right: the guidance image I (RGB), the input image P (Grayscale) and the output image F (Grayscale). It can be observed that the input binary mask is enhanced according to the guidance image and it proves the edge-preserving feature of the filter.

In theory, the guided filter is defined to be linear translation-variant (He et al., 2013). The filtering output F is a linear transformation of the guidance image I in a window w_k centred at the pixel k of size $(2r+1) \times (2r+1)$:

$$F_i = a_k I_i + b_k \qquad \forall i \in w_k, \tag{10}$$

where (a_k, b_k) are linear coefficients and constants in w_k. These linear coefficients are determined by minimizing the difference between the output image F and the input image P, therefore a cost function can be written as:

$$E(a_k, b_k) = \sum_{i \in w_k} ((a_k I_i + b_k - P_i)^2 + \epsilon a_k^2) \tag{11}$$

where ε denotes a regularization parameter preventing a_k from being too large. By linear regression, the solution of the above equation can be obtained as:

$$a_k = \frac{\frac{1}{|w|} \sum_{i \in w_k} I_i P_i - \mu_k \bar{P}_k}{\sigma_k^2 + \epsilon} \tag{12}$$

$$b_k = \bar{P}_k - a_k \mu_k \tag{13}$$

Here μ_k and σ_k are the mean and variance of I in w_k correspondingly. $|w|$ denotes the number of pixels in the window w_k and $\bar{P}_k$ is the mean value of P in the window.

Therefore, the linear model in (10) can be applied to solve for the filtering output F. However, there are multiple windows w_k that contains pixel i, so the value of F_i in (10) is not a constant value while computing for different windows. The strategy applied here is to average all possible values of F_i and it becomes:

$$\begin{aligned} F_i &= \frac{1}{|w|} \sum_{k: i \in w_k} (a_k I_i + b_k) \\ &= \bar{a}_i I_i + \bar{b}_i \end{aligned} \tag{14}$$

where $\bar{a}_i = \frac{1}{|w|} \sum_{k \in w_k} (a_k)$ and $\bar{b}_i = \frac{1}{|w|} \sum_{k \in w_k} (b_k)$.

The linear relationship between the guidance image I and the output image F no longer maintains since the linear coefficients $(\overline{a_\iota}, \overline{b_\iota})$ are spatially-variant. However, the gradients of $(\overline{a_\iota}, \overline{b_\iota})$ are supposed to be much smaller than that of I near sharp edges, since $(\overline{a_\iota}, \overline{b_\iota})$ are outputs of an average filter. Approximately the linear relationship still holds in this case and it implies that abrupt gradient changes in I can be preserved in F.

- **Image blending by guided filtering**

The overall workflow of blending with the guided filter can be summarised as the following steps:

1) Capture and load a stack of the target image. Down-sampling may be performed to reduce the image dimensions and increase computation speed. In our experiment, a dimension of at least 1080p (1920 x1080) is acceptable. See Figure 3.3 (A) and (C) for example stack images.
2) Load Homography matrix and register the stack images.
3) Decompose stack images. Each source image is decomposed into a two-layer representation (base layer and detail layer) by average filtering. The base layer B_i can be obtained as:

$$B_i = I_i * Z \tag{15}$$

Where I_i denotes the i-th source image and Z is a 31×31average filter. The detail layer D_i can be obtained as:

$$D_i = I_i - B_i \tag{16}$$

The base layer extracts the large-scale variations in intensity, and the detail layer keeps the small-scale texture information. Figure 3.6 demonstrates an example pair of decomposition output.

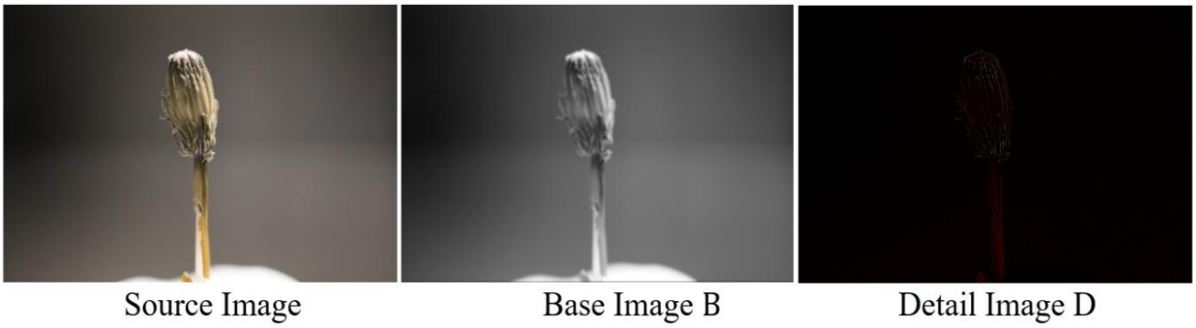


Figure 3. 6: Image Decomposition Example. A source image is decomposed into a base layer and a detail layer.

4) Construct weight maps with guided filtering
Each source image is applied with a Laplacian filter to generate the high-pass image H_i.

$$H_i = I_i * L \tag{17}$$

where L is a 3×3 Laplacian filter. Therefore, the saliency maps S_i can be constructed by the local average of the absolute value of H_i:

$$S_i = |H_i| * g_{r_g,\sigma_g} \tag{18}$$

Here, g is a Gaussian low-pass filter with a size of $(2r_g + 1)(2r_g + 1)$ and (r_g, σ_g) are pre-defined as 5. These saliency maps offer characterisation of the saliency level and local sharpness of each source image. According to the values of saliency, the weight maps are obtained as below:

$$P_i^K = \begin{cases} 1 & if\ S_i^K = \max(S_1^K, S_2^K, S_3^K, \ldots, S_N^K) \\ 0 & otherwise \end{cases} \tag{19}$$

Here S_i^K denote the saliency value at pixel k in the i-th image and N is the number of source images. This weight map reveals which input image has the highest value of saliency at each pixel. At this stage, the obtained weight maps are usually noisy and cannot be straightforwardly applied for blending. To denoise the weight maps, guided image filtering is applied on each weight map P_i with its source images I_i as guidance:

$$W_i^B = G_{r_1,\epsilon_1}(P_i, I_i) \tag{20}$$

$$W_i^D = G_{r_2,\epsilon_2}(P_i, I_i) \tag{21}$$

Where W_i^B and W_i^D are the guided weight maps of the base and detail layer of each source images and $(r_1, \epsilon_1, r_2, \epsilon_2)$ are the parameters of the guided filter. The values of N guided weight maps are normalized, and the summation of weight values becomes “1” at each pixel.

In Figure 3.7, two sets of example outputs generated from the above procedures are displayed for comparison. The entire process does not change when blending multiple images. From top to bottom:

- The source images I_1 and I_2, with different focused regions.
- The saliency maps S_1 and S_1reveal the saliency value of each source image correspondingly. These maps are similar to the local sharpness plots in Figure 3.3.
- The weight maps P_1 and P_2, which runs a simple comparison between the saliency maps. For each pixel of the weight map, the weight value is assigned to “1” if the corresponding saliency map has a maximal value among all saliency maps. Otherwise, the weight value is “0”.
- Guided weight maps $Wb1$ $Wb2$, $Wd1$ and $Wd2$ for blending the base and detail layer. The weight maps P_1 and P_2, are fed into a guided filter as input images, and the source images are used as guidance. Comparing to the weight maps P_1 and P_2 before guiding, it can be viewed that the guided weight maps has fewer noises and important edges are preserved.

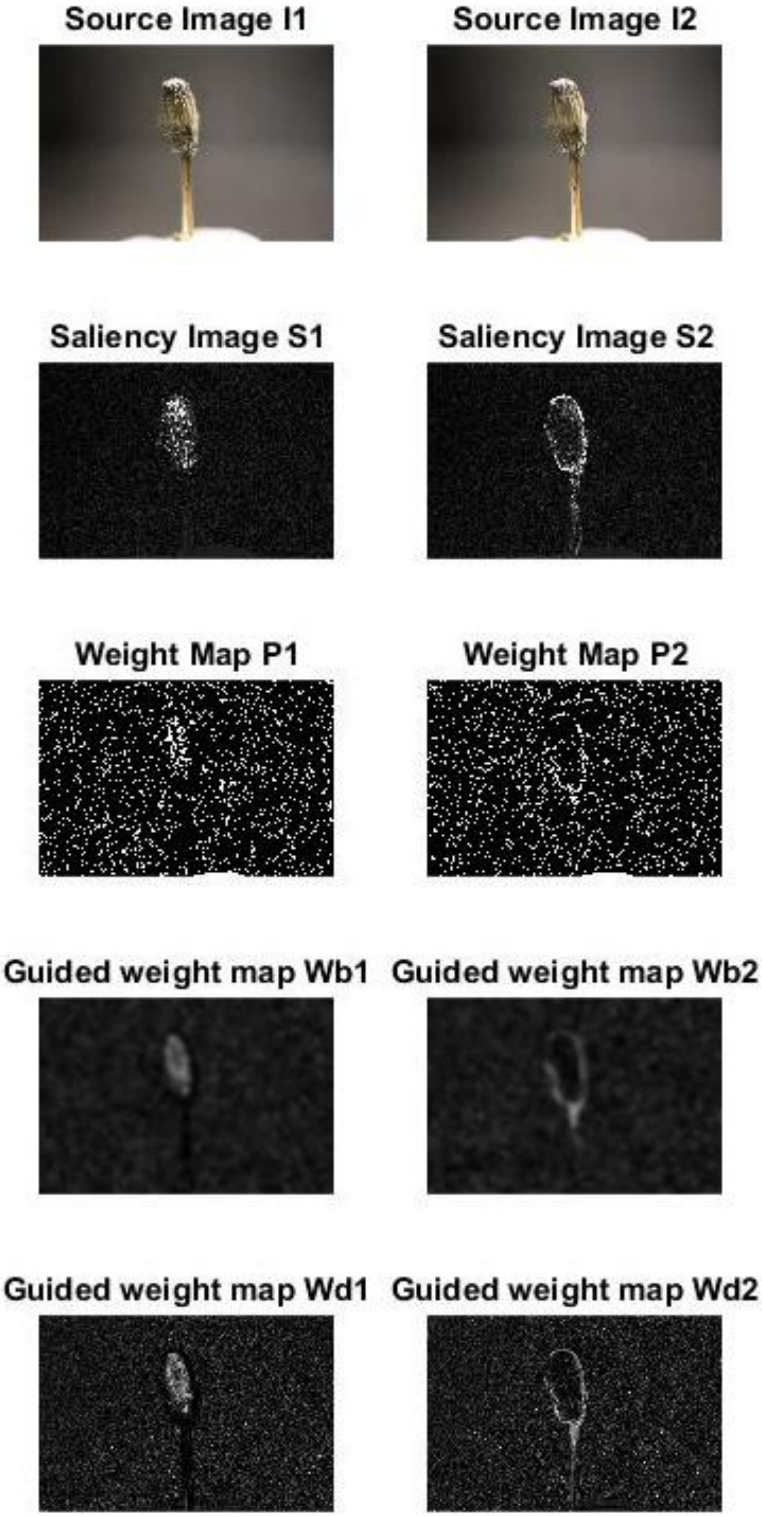


Figure 3. 7 A set of example outputs of weight-map construction. From top to bottom: a) The source images I_1 and I_2, with different focused regions. b) The saliency maps S_1 and S_1 reveal the saliency value of each source image correspondingly. c) The weight maps P1 and P2, which runs a simple comparison between the saliency maps. d) Guided weight maps Wb1 and Wb2 for blending the base layer. The weight maps are fed into a guided filter as input images, and the source images are used as guidance. Parameters of guided filter are set to be: (r_1=45, ϵ_1=0.3) by default. e) Guided weight maps Wd1 and Wd2 for blending the detail layer. Similarly, the weight maps and source images are fed into the guided filter as input and guidance image correspondingly. Parameters of guided filter are set to be: (r_1=7, ϵ_1=10e-6) by default

5) Reconstruct decomposed images. The base and detail layers of different source images are blended according to the guided weight matrix:

$$\bar{B} = \sum_{i=1}^{N} W_i^B B_i \tag{22}$$

$$\bar{D} = \sum_{i=1}^{N} W_i^D D_i \tag{23}$$

The final blending output is obtained as:

$$F = \bar{B} + \bar{D} \tag{24}$$

Figure 3.8 shows an example of reconstruction output from blending ten source images. It can be observed that both layers are reconstructed while feeding in more source images. Note that the detail layer is amplified in the figure, in order for better display.

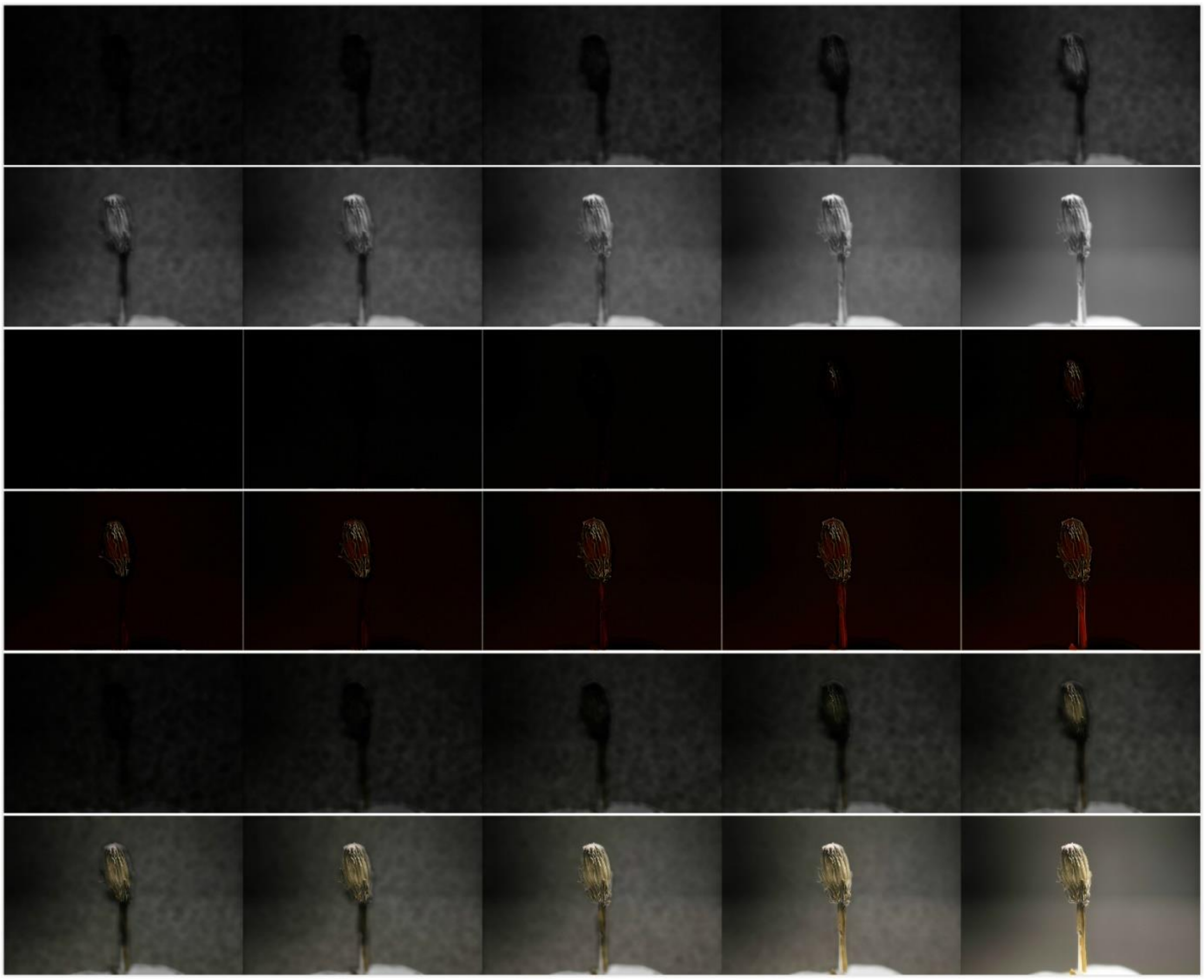


Figure 3. 8: An example of two-scale image reconstruction process for a stack of 10 plant images. There are 10 source images and the summations in formula (22) (23) and (24) are repeated for 10 times. The above images demonstrate the output of each step of summation. From top to bottom: 1) step-wise summation of base-layer images 2) step-wise summation of detail-layer images, note that these detail images are amplified as the original images are hardly visible 3) step-wise summation of blending results. These images demonstrate a process that the fusion results are gradually generated by blending each source image.

3.2 PART 2. FINE-GRAINED RECOGNITION AND VISUAL ATTRIBUTE GROUNDING

In this part of the research, we aim to train a fine-grained classification network and utilise it to implement visual attribute grounding in a supervised manner.

3.2.1 Data set and Network model

Fine-grained classification refers to the problem of discriminating between visually similar sub-categories, such as different species of insects or birds. Therefore, the fine-grained dataset CUB-200-2011 (Welinder, et al., 2011), which contains 11794 images of 200 bird species, is applied as a source of data in this research. There are initially 5997 training images and 5797 testing images. However, the ratio between training and testing images is adjusted to 9:1 for expanding training space and therefore, it results in 10608 training patterns and 1180 testing patterns. Bounding box algorithm is applied to the images to reduce background noise. Figure 3.9 shows examples of an original bird image and bounding-boxed image from CUB-200-2011.

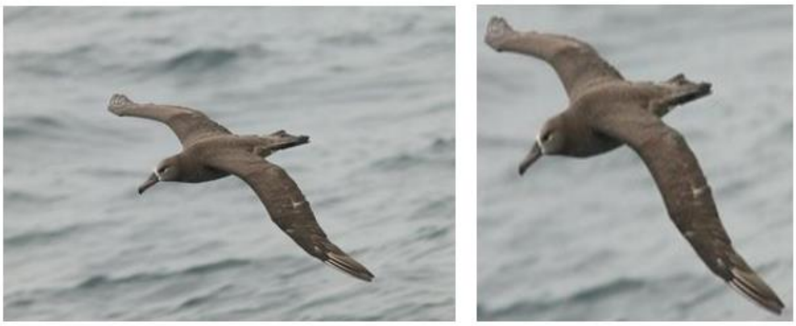


Figure 3. 9: Original source image from CUB-200-2011 (320x223) and Cropped image (224x224) from bounding box algorithm. A slight distortion on the bird's dimension can be observed, but the main features are still preserved after cropping.

One important reason that we select this CUB dataset is that it provides the ground truth of visual attributes of each image. For example, the bird in Figure 3.9 has a long beak, grey wing and black eyes. This information is essential for later experiments.

We use a Resnet-50 model for the fine-grained classification task. The reasons that we apply this network model are that:

1) Comparing to other well-known network structures (VGG and GoogLeNet, etc.), Resnet model has an outstanding performance on image classification tasks, and it won the 1st place on the ILSVRC 2015 classification task.
2) There is a pre-trained Resnet model established in Keras (Chollet, 2018), which is a high-level neural network API, and it is pre-trained on the ImageNet data set.

To fit the Keras Resnet-50 model to the CUB dataset, the layers after the last convolutional layer are modified. A global pooling layer and a fully connected layer are added to generate 200 classification output for the CUB dataset.

	Learning Rate	Number of Epochs	Batch Size
Transfer Learning	1e-4	7	12
Fine Tuning	1e-5	5	12

Table 3. 1: Parameters of training. Different values of learning rate and different numbers of epochs are used for transfer learning and fine-tuning.

This modified network structure is transferred on the CUB dataset and then fine-tuned by the parameters stated in Table 3.1.

3.2.2 Visual Attribute Grounding

We assume that each visual attribute (e.g., black beak or long wings in Figure 3.9) of the input images should correlate to one or several filters in the last convolutional layer. For each visual attribute, its corresponding filters will always be activated if images containing that attribute are fed continuously into the network. If we feed several images that contain the target attribute into the network, the accumulated activation of its corresponding filter will be higher than other filters. An important assumption here is that for training the framework to recognise target visual pattern, it requires a set of images that all contain the target pattern but no other common patterns. Thus, the relationship between last conv-layer filters and visual attributes can be investigated through a filter-attribute probability density function, which indicates the probability of a filter activates for a visual attribute. Theoretically, the region of target visual attribute can be obtained by a linear combination of last-layer feature maps, with the filter-attribute probability density function as weights.

To obtain the filter attribute probability density function, we first compute the Term Frequency (TF) and the Inverse Document Frequency of each attribute t_i

$$TF = number\ of\ occurances\ of\ t_i\ among\ all\ features \qquad (25)$$

$$IDF = \log(N/D) \qquad (26)$$

Where N is the total number of images and D is the number of images containing feature t_i. Thus, the value of TF/IDF implies the relative importance of the feature t_i.

Then, then filter attribute probability density function $p(t_i|f_k)$ can be formed to represent the correlation between the feature t_i and last conv-layer filter f_k:

$$p(t_i|f_k) = p(t_i) * p(f_k|t_i) \qquad (27)$$

Here, $p(t_i)$ is the value of TF/IDF of feature t_i and $p(f_k|t_i)$ represents the probability that the occurrence of the feature t_i is the reason of activation of the filter f_k:

$$p(f_k|t_i) = \sum_{j=1}^{m} p(f_k|x_j, t_i) * p(x_j|t_i), \qquad (28)$$

where m denotes the total number of images and $p(x_j|t_i)$ reveals the occurrence of the feature t_i in image x_j:

$$p(x_j|t_i) = \begin{cases} 1, if\ t_i\ is\ in\ x_j \\ 0,\ \ otherwise \end{cases} \qquad (29)$$

$p(f_k|x_j, t_i)$ in (28) measures the likelihood that the occurrence of feature t_i in image x_j is the reason for activation of filter f_k and it can be approximated by the normalized global pooling layer output:

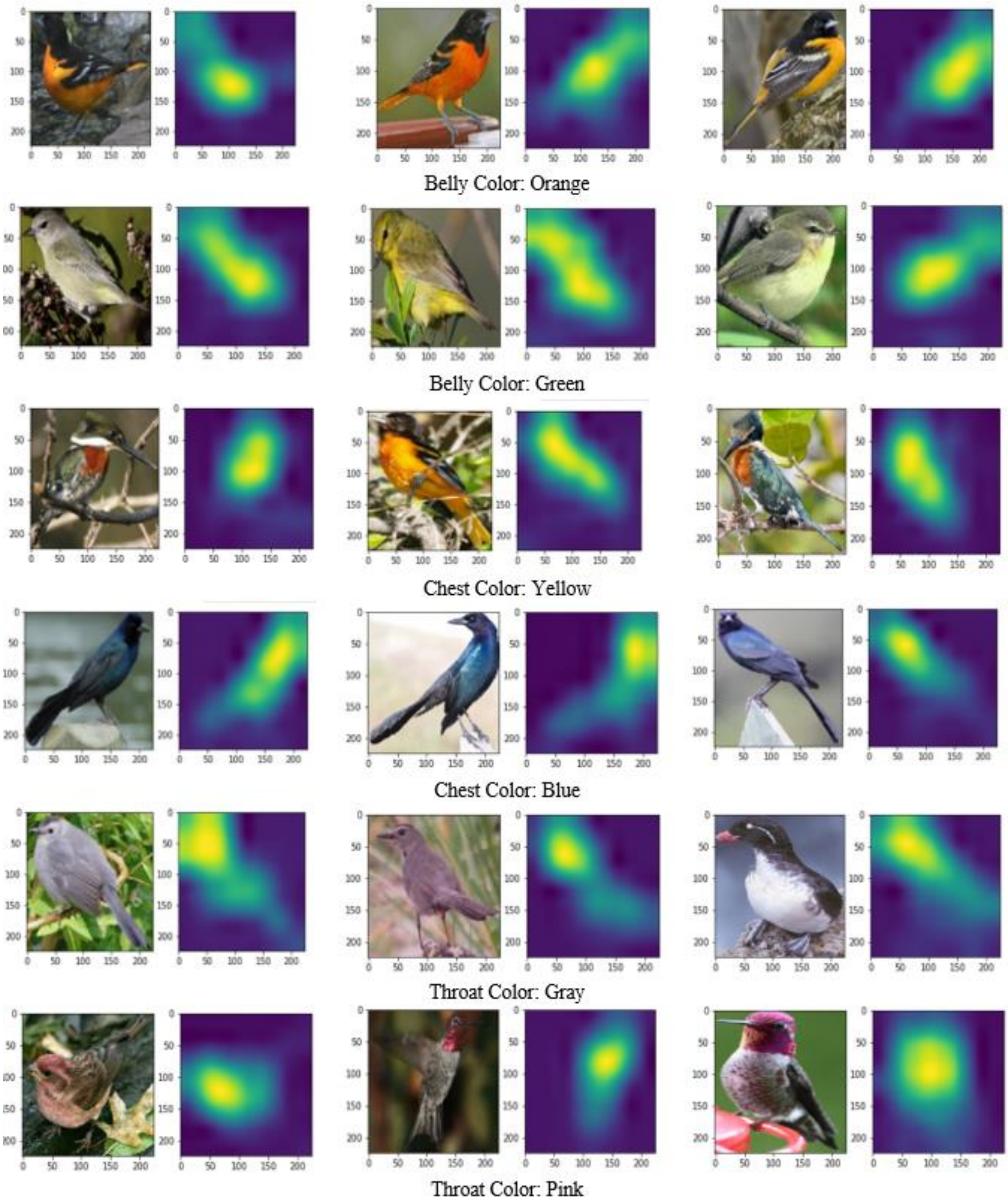
$$p(f_k|x_j, t_i) \approx \sigma(\varphi(f_k(x_j))) \tag{30}$$

where σ represents the normalization function, φ is the global pooling layer output, and $f_k(x_j)$ is the filter's output of image x_j as input.

Thus, the image region of interest can be obtained by reweighting the final conv-layer feature maps according to the obtained filter- attribute probability density function:

$$Region\_of\_interest = \sigma(\sum_{k=1}^{m} p(t_i|f_k) * f_k(x_j)), \tag{31}$$

The entire training set is fed into the modified Resnet-50 model for training the Bayesian Inference Framework. Figure 3.10 demonstrates some example bounding-boxed CUB images and their visual attribute activations. Each row displays a different visual attribute with the activation heatmaps indicating the region of interest. It can be observed that the proposed Bayesian-inference framework generates decent predictions at the region of target features.



Belly Color: Orange

Belly Color: Green

Chest Color: Yellow

Chest Color: Blue

Throat Color: Gray

Throat Color: Pink

Figure 3. 10 Image example and heatmaps correspond to different semantic attribute. Each row shows three example CUB images with their visual attribute activations. Here, the activations indicate the location of target attribute.

Chapter 4 Discussion

4.1 MULTI-VIEW IMAGE COLLECTION

4.1.1 Edge-shaped distortion of the saliency measurement method

As shown in Figure 4.1, the edge-shaped noise always appears as a side-effect of image blending, especially when the averaging power *P* is high. Recall that a high value of *P* will cause stronger edges preserved in the blending result. This edge-shaped distortion implies that the error is introduced by imperfect registration and calibration of stack images. Some sources of error that may bias calibration are listed as:

- We are using a 2D flat dot matrix to calibrate another 3D object with an entirely different size, shape and dimension. There is always perspective and position error between these two objects.
- The repeated movement of the camera may not be the same due to machine error and camera vibration.

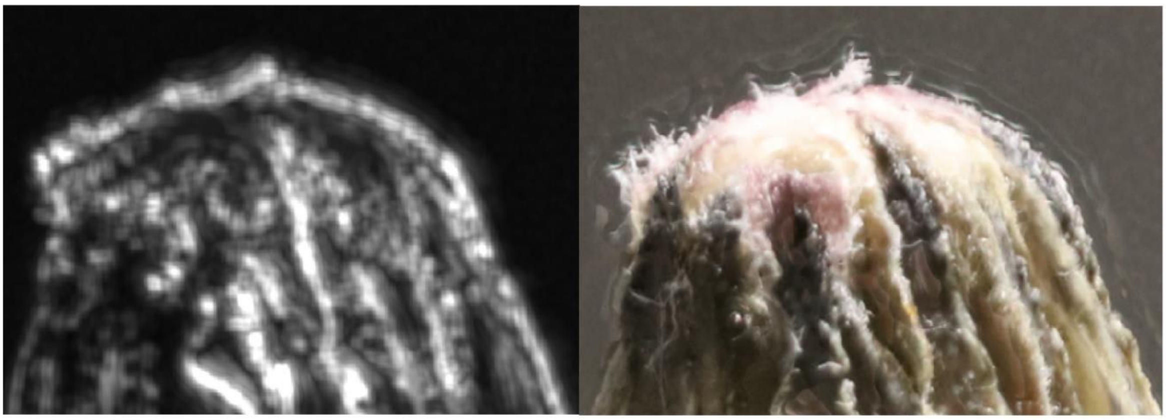


Figure 4. 1: Calibration Error when P = 15. Left: global stacked sharpness detection. Right: final blended image. The edge-shaped noise has similar shape and pattern in both images and therefore, it implies that this error is introduced by imperfect registration of stack images.

In practice, it is difficult to obtain a calibration result that generates perfectly-registered source images, by a sensitive macro lens and a regular camera model. One solution may be applying “tilt-shift lens”, which is designed for fixing the image perspective error and reducing camera vibrations. However, at the current stage of this research, redundant edges are always detected and presented in the final blending result as edge-shaped noises.

To eliminate redundant edges, one effective solution is to apply dilation on saliency maps to provide a tolerance for faulty calibration. A dilation process is performed on the local salience maps before computing the weight matrix. The dilated edges become thicker and the overlapping of edges is enhanced. Comparing the stacked sharpness before dilation (Figure 4.1 left) to itself after dilation (Figure 4.2 left), it can be observed that the later reduces the superimposed edge-shape noises.

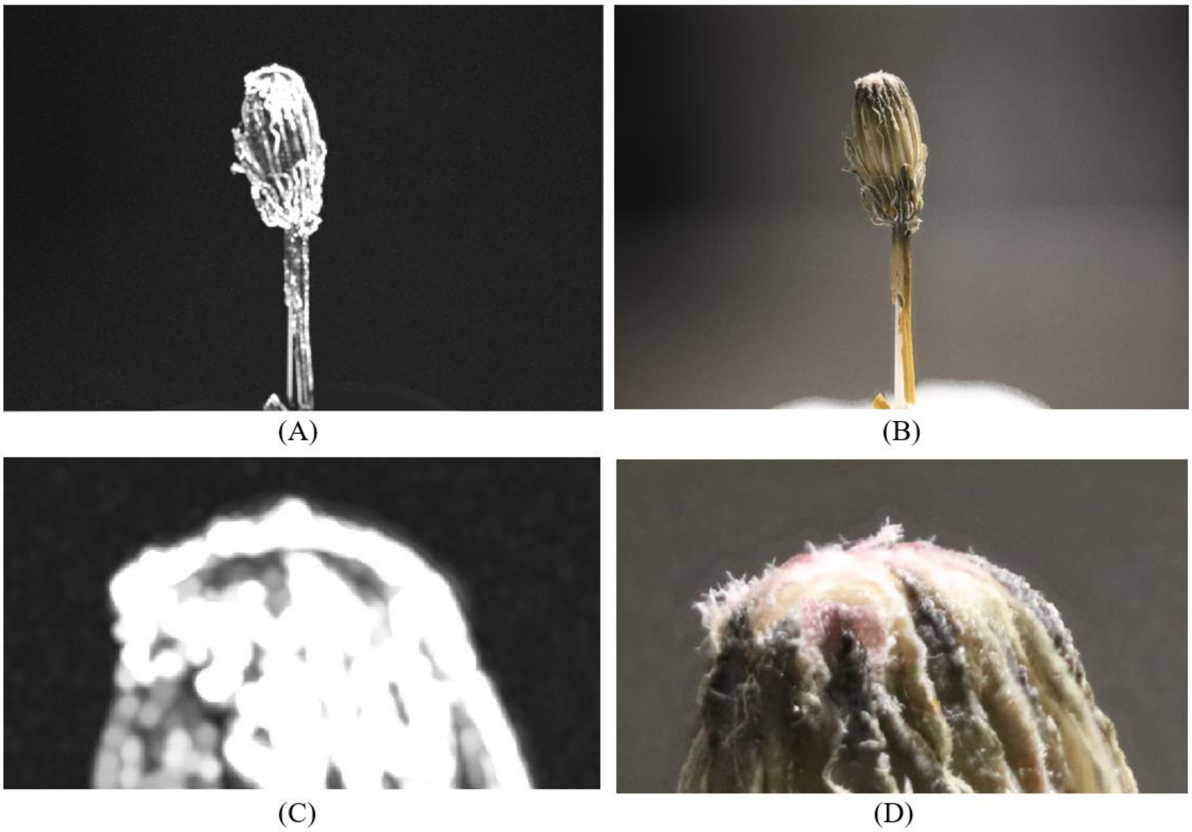


Figure 4.2 The blending output with dilation radius = 11 pixels and averaging power P = 15. A): Global stacked sharpness/edge; B): Blended result; C) and D): zoomed in version correspondingly. It can be observed that the edge-shaped noise is reduced significantly, comparing to Figure 4.1

The reason that dilation works is that it can effectively enhance the overlapping between detected sharp edges (Figure 4.2 A and C). The blending algorithm relies on saliency measurement to decide the source value for each pixel of the blending result. An overlapping saliency measurement helps the algorithm to learn more about saliency level of pixels and therefore, it can result in a sharper and clearer merged output.

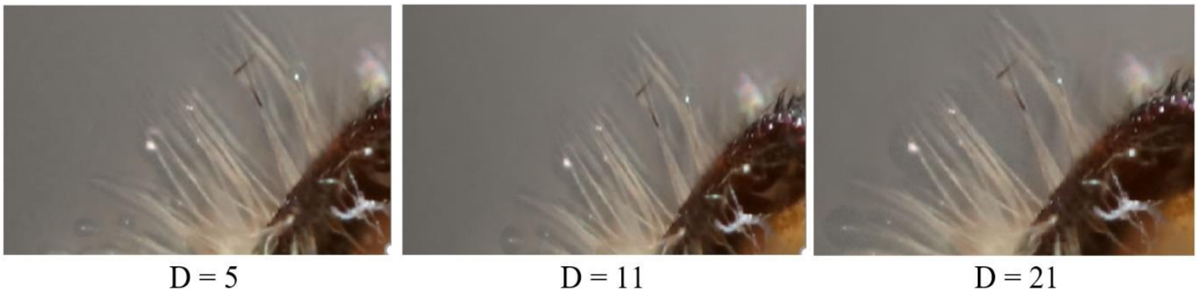


Figure 4. 3 Comparison of blending result from different dilation diameter. In this experiment, we used a circular filter for dilation and as a result, the introduced distortion also follows a circular shape. It can be observed that dilation distortion is enhanced while increasing its diameter.

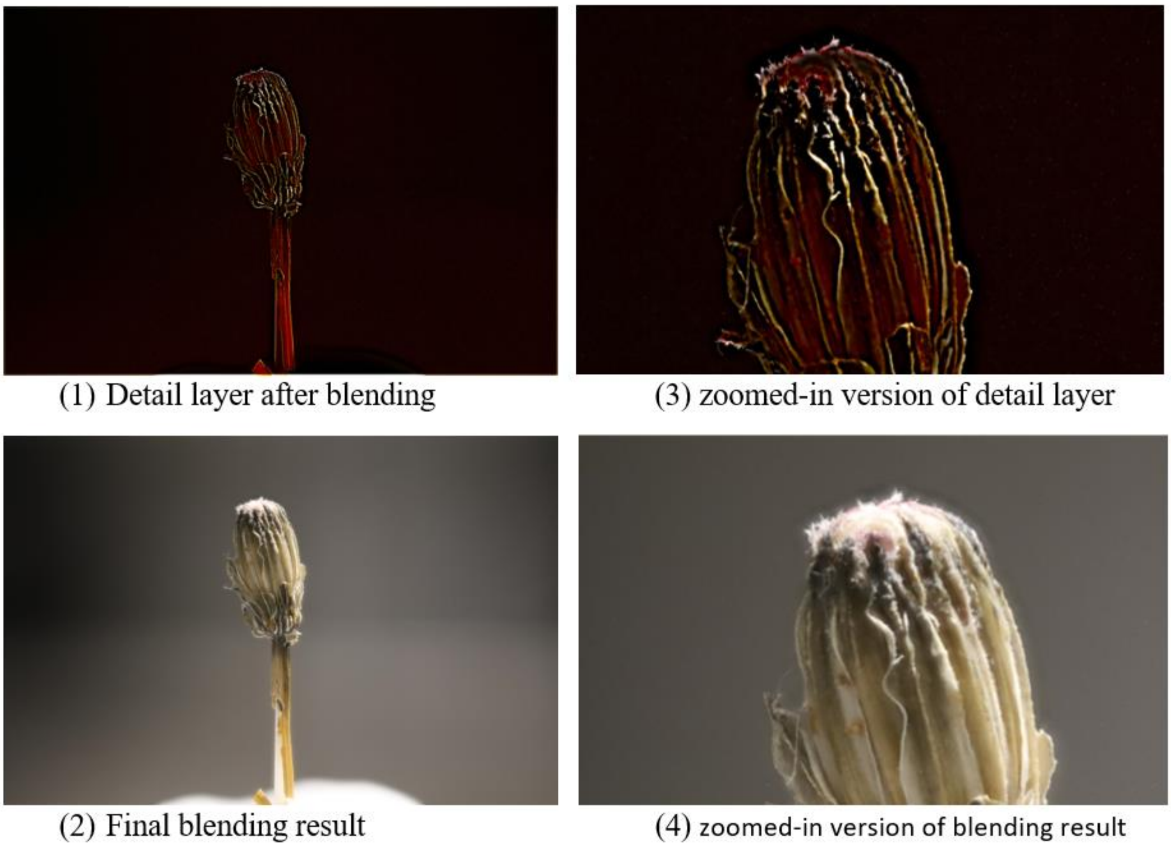
However, the dilation method is not applicable for blending over-complicated structures, such as fur or hair on the insect body. Slight circular distortion may occur when the dilation diameter D is relatively larger than the object size, as shown in Figure 4.3. While blending furry specimens, it requires careful decisions of the value of dilation diameter, in order to reach a balance between the edge-shaped noise and the circular distortion.

In summary, the current blending method relies on local saliency detection and it suffers from edge-shaped noises, which is introduced from imperfect dot-matrix calibration. Dilation of saliency maps can be regarded as a solution, and it effectively reduces the calibration error. However, slight circular distortion may be introduced when blending complicated body structures, such as furs and hairs.

4.1.2 Results of Image Blending with Guided Filtering

The fusion result generated by guided filtering method is demonstrated in Figure 4.4. Similarly, a slight edge-shape noise is revealed from the results. Since the noise appears in both detail layer (black shade around the object) and blending result, it implies that calibration error is still the main reason for distortion.

We try to add a dilation process into the algorithm, as what we did in the other method. It makes no sense if we dilate before the guided filtering process, as the guided filter will suppress the dilation result according to the source images. However, if we dilate after applying guided filtering, then we must dilate the weight maps for base and detail layers together. In this case, dilation takes much more computational resources, which is beyond what we are expecting from an image fusion algorithm for the 3D reconstruction system. As a result, the dilation solution is not a proper solution for the guided filtering method to remove distortions.



(1) Detail layer after blending

(3) zoomed-in version of detail layer

(2) Final blending result

(4) zoomed-in version of blending result

Figure 4. 4: Results of Image blending with guided filtering method (Here we ignored the base layer result as the over-smooth figure cannot provide much meaningful information). The edge-shaped noise appears in both of (1) and (2), which implies that the error is still caused by faulty calibration and registration of stack images.

4.1.3 Comparison between two methods

In this experiment, four fusion quality metrics are adopted in order to assess the image fusion performance:

1) Normalised mutual information Q_{MI} $(\text{Hossny, et al.}, 2008)$ is a metric that utilized information theory and it can be computed as:

$$Q_{MI} = 2\left[\frac{MI(A,F)}{H(A)+H(F)} + \frac{MI(B,F)}{H(B)+H(F)}\right] \tag{25}$$

where A, B and F are the two source images and fusion result correspondingly. $H(\,)$ stands for marginal entropy (amount of information presented in the image) and $MI(\,)$ indicates the mutual information between images. The quality metric Q_{MI} reveals how well the mutual information between original and fused image is preserved.

2) The ratio of spatial frequency error rSFe (Zheng et al., 2007) measures the quality of the fused image according to spatial frequency. It determines if the fused image can be potentially improved or not. This metric is computed as

$$rSFe = (SF_F - SF_R)/SF_R \tag{25}$$

here SF_F SF_R denotes the four-directional spatial frequency of fused the image and reference image correspondingly, where the reference image is generated from the source images. A positive value of $rSFe$ implies an over-fused image, with some noise or distortion introduced during fusion. A negative $rSFe$ reveals an under-fused image, with loss of some meaningful information. An ideal value of $rSFe$ is zero.

3) Modified fusion artifacts $N^{AB/F}$ (Kumar, 2015) measures the fusion artifacts and its equation is given as:

$$N^{AB/F} = \frac{\sum_{\forall i}\sum_{\forall j} AM_{i,j}[\left(1 - Q_{i,j}^{AF}\right)w_{i.j}^{A} + (1 - Q_{i,j}^{BF})w_{i.j}^{B}]}{\sum_{\forall i}\sum_{\forall j}(w_{i.j}^{A} + w_{i.j}^{B})} \tag{26}$$

Here, (i, j) stands for each pixel in the image and $AM_{i,j}$ denotes the location of fusion artifacts, where the fused gradients are enlarged from the input. $(Q_{i,j}^{AF}, Q_{i,j}^{BF})$ indicates the total information transferred from source image A and B to the fused image respectively and $(w_{i.j}^{A}, w_{i.j}^{B})$ stands for the perceptual weights of source images. We would expect a smaller value of $N^{AB/F}$, since it implies that there are fewer artifacts introduced during fusion.

4) Computation Time is another important metric, especially when the image acquisition time is limited for a faster 3D scan. We use a computer from ANU lab, with a CPU: Intel(R) Core™ I7-7700 CPU@3.60GHz and 16 GB memory to run these two different blending methods. A stack of ten 4344×2898 images is applied for blending.

	Q_{MI}	rSFe	$N^{AB/F}$	Processing Time
Saliency measurement Method	1.0471	-0.1513	0.0765	49.916s
Saliency Measurement Method with dilation	1.0699	-0.3090	0.0446	55.163s
Guided Filtering Method	1.3150	-0.0717	0.0045	101.595s

Table 4. 1: Comparison between blending methods concerning the performance metrics. The "better" results from the comparison are marked red.

Table 4.1 demonstrates the comparison of above four metrics. These quantitative comparisons reveal some properties that cannot be visually observed from the comparison between blending outputs:

- The highest value of Q_{MI} indicates that the guided filtering approach can provide a fusion result that is closest to the source images. It cannot be visually observed from the blending result, but the guided filtering approach can extract the most valuable information (for example, edges and textual) from the source images, comparing to the single saliency measurement method with or without dilation. This conclusion is expected, since the purpose of applying guided filtering method is to preserve more edge features from the source images.
- The higher value of $N^{AB/F}$ reveals that the dilation method is indeed reducing the edge-shaped artifacts, comparing to the case without dilation. However, the guided filtering approach still performs the best in terms of avoiding artifacts.
- According to the negative value of rSFe, all these methods are offering under-fused results that imply loss of information during fusion. They apply Laplacian of Gaussian filtering (or Difference of Gaussian) to measure the local saliency and sharpness of each source images. The filter parameters determine the quality of saliency measurement, and it may be a source of information loss. Accordingly, future optimisation should focus on improving the saliency measurement to reduce information loss.
- The above result of performance metrics proves that blending with guided filtering method can produce qualitatively best-blending output, which has the least information loss and least additive artifacts during fusion. However, the guided filtering algorithm takes much more computational resources (e.g. time), according to the comparison with the saliency measurement method. In practice, the latter is preferable when we want to apply this blending algorithm in the pipeline with image capturing, in order to speed up the overall scanning procedures.

In conclusion, the optimal choice of the above three methods depends on the purpose. The guided filtering method is a better option if best-quality fusion is required, without any limitation on the cost of time, for example, some image collection experiments. On the other hand, the saliency measurement method is preferable when the fusion process is expected to be as fast as possible and sacrificing some blending accuracy is affordable. In the case of 3D modelling, it is usually the second case, where a fast scanning process is preferred.

4.2 FINE-GRAINED RECOGNITION AND VISUAL ATTRIBUTE GROUNDING

4.2.1 Training and Testing performance of the modified network

The modified Resnet-50 model was trained and then fine-tuned (with a decreasing learning rate) on the target CUB dataset. As a result, the network reached a training accuracy of 99.99% and a testing accuracy of 82.12%. Therefore, we concluded that the network was fully-trained to apply for later experiments. Figure 4.5 demonstrates the training log, which reveals the same story as what we described here.

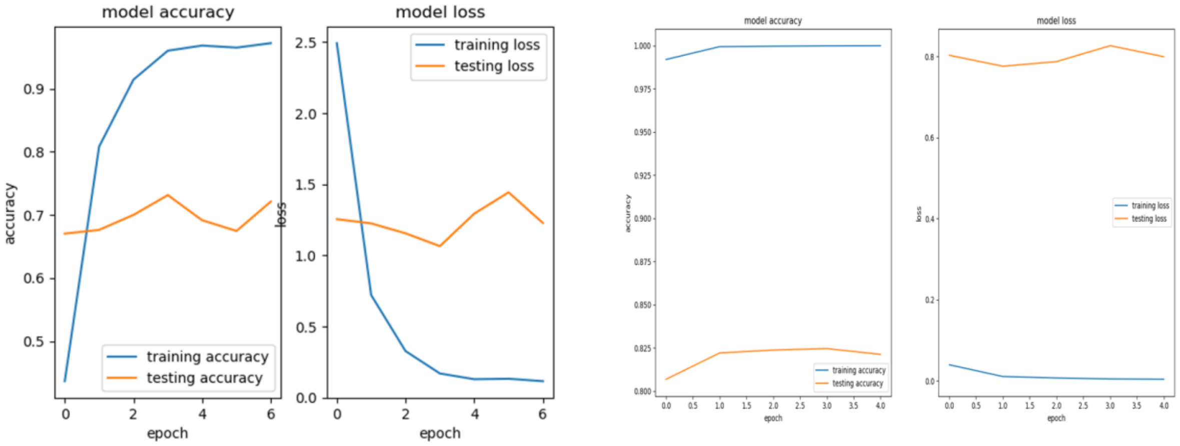


Figure 4. 5: Training and Testing logs for transfer learning (left) and fine-tuning (right). A significant increase in accuracy can be observed from these plots.

4.2.2 Performance of Visual Attribute Grounding

In order to investigate the accuracy of the proposed visual grounding method, the generated heatmap peak is compared to the ground truth of feature location. A percentage of correct key-points (PCK) algorithm is performed on the top 50 most typical features. A successful detection is marked if the peak of the generated heatmap is within a circular region of the corresponding ground truth location, where such a region has a radius of $0.3 * bird\_size$. Table 4.2 shows the detection accuracy of the proposed methods. For comparison, performances of other baseline methods, including raw feature maps and raw feature peaks, are also listed out. It can be observed numerically that the proposed visual attribute grounding method can produce a meaningful feature extraction, although its accuracy of feature extraction can still be improved.

	Raw Filter Map	Raw Filter Peak	Visual Attribute Grounding
PCK@0.3	28.3%	47.5%	60.9%

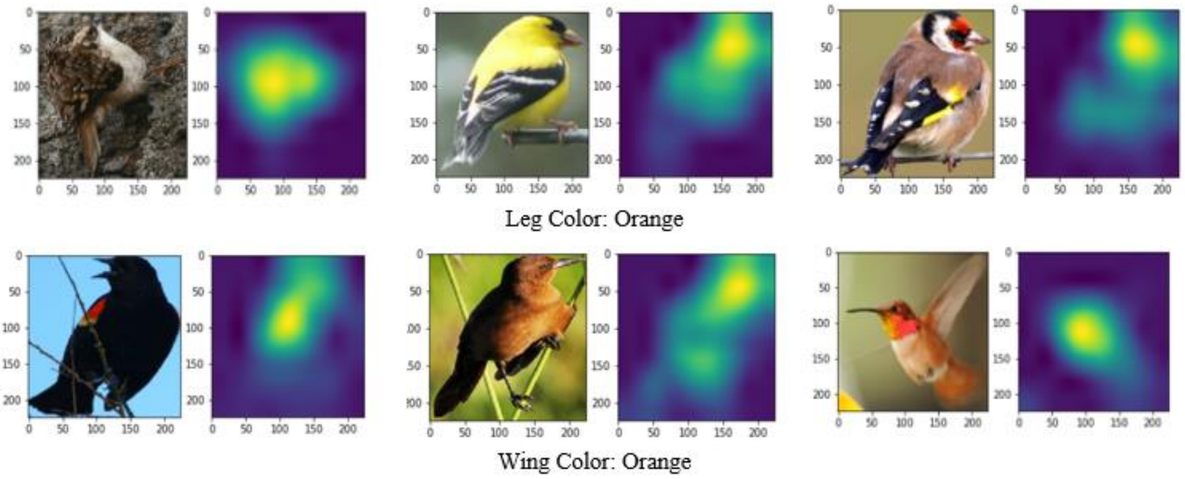
Table 4. 2: Comparison of accuracy and purity of part semantics. The detected patterns from visual attribute grounding are compared to other methods, which includes raw feature maps and the highest activation peaks on features maps.

Furthermore, we perform some more experiments to evaluate the limitations of the proposed framework:

- Weak detection performance on bird legs and wings

In this experiment, we focus on the attributes at different parts of birds (head, wings, body and feet). Feature extraction performance of each part is compared to reveal the limitation of the proposed method.

We use human users to annotate the detection accuracy of each part. It is observed that the proposed Bayesian framework has relatively worse performance on detecting attributes at wings and feet, and Figure 4.6 shows some failure detections.



Leg Color: Orange

Wing Color: Orange

Figure 4. 6: Example of failure detections of "orange leg" and "orange wing". The activation heatmaps indicate the wrong regions of target attribute.

There are two reasons that could result in a degraded performance in detecting wings and feet:

1) Wings and legs naturally have many more poses than the other body parts like the breast or the beak. According to Figure 4.6, the poses of birds' feet and wings have a significant variation between different images. For example, opening and closing the wings results in two entirely different patterns. A variable pattern may require many more filters to be trained for detection, but from the nature of the neural network, it will eventually converge to secure and stable features that require fewer filters for classification tasks.
2) According to Table 4.3, it counts the number of CUB attributes at different parts of a bird, and there are fewer attributes at wings or legs. It results in fewer training images containing wing or leg features that can be applied in the Bayesian Inference framework and therefore, the detection of wing or leg attributes can be easily biased by arbitrary noises in the images.

	head	Body	Wing	Leg
Number of attributes	62	57	5	16

Table 4. 3: Number of CUB attributes at a different part of birds. It can be observed that there are few attributes about wings or legs.

- Unexpected redundant patterns in training set

The Bayesian inference framework assumes that we are feeding images, which contains target visual pattern with no other common patterns, into the framework to accumulate the activations and to investigate the filter-attribute probability. This is an essential assumption to ensure that the accumulation of filters' activations of target pattern will not fail due to accumulated activations from the other patterns. However, this assumption does not always hold in practice. In this experiment, two different metrics (accumulated activations and distinctiveness measurement) will be applied to reveal the existence of redundant patterns that bias the performance of feature detection.

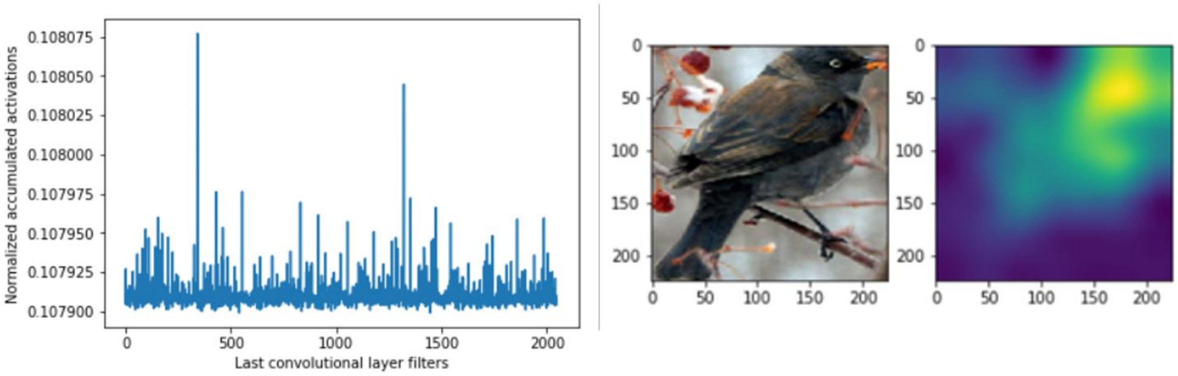


Figure 4. 7: Left: Normalized accumulation of filter activations fed by 221 bird images with white eyes. The noisy signal implies that biases exist in detection. Right: an example raw image and feature detection result of white-eye bird.

Figure 4.7 (left) demonstrates the accumulated activations of the pooling layer connected to the last conv-layer filters. There are 221 bird images with white eye patterns fed into the framework as a training set. The peaks in the plot indicate the strength of filter activations during training, and the amplitude of each peak can be applied as a weight for the linear combination of last conv-layer feature maps. Hence, it generates a mask mapping to the target feature (white birds eye) in the raw image, as shown in Figure 4.7 (right). According to the assumption of the Bayesian inference framework, we would expect one or a few filters to be continuously activated by the target pattern (white eyes of birds) during training. In other words, the normalised accumulation activations are expected to have only a few peaks. However, Figure 4.7 (left) shows a noisy activation behaviour, and it implies that there are many filters that have been accidentally activated by other patterns in the training set. This phenomenon reveals that there are other potential common patterns in the 221 bird images, and they all become noise in linear combination when we want to detect the region of white eyes.

In order to further investigate the potential features and redundant activations, we randomly select 20 images from the 221 bird images with white eye patterns, as shown in Figure 4.8. It can be visually observed that the white eyes are the only common attribute among all different birds in these images. As discussed previously, the accumulated activation can be easily biased by other attributes that occur in the training space, for example, black tail and red beak. Thus, we adopt a distinctiveness test on each hidden filter to reveal their activations.

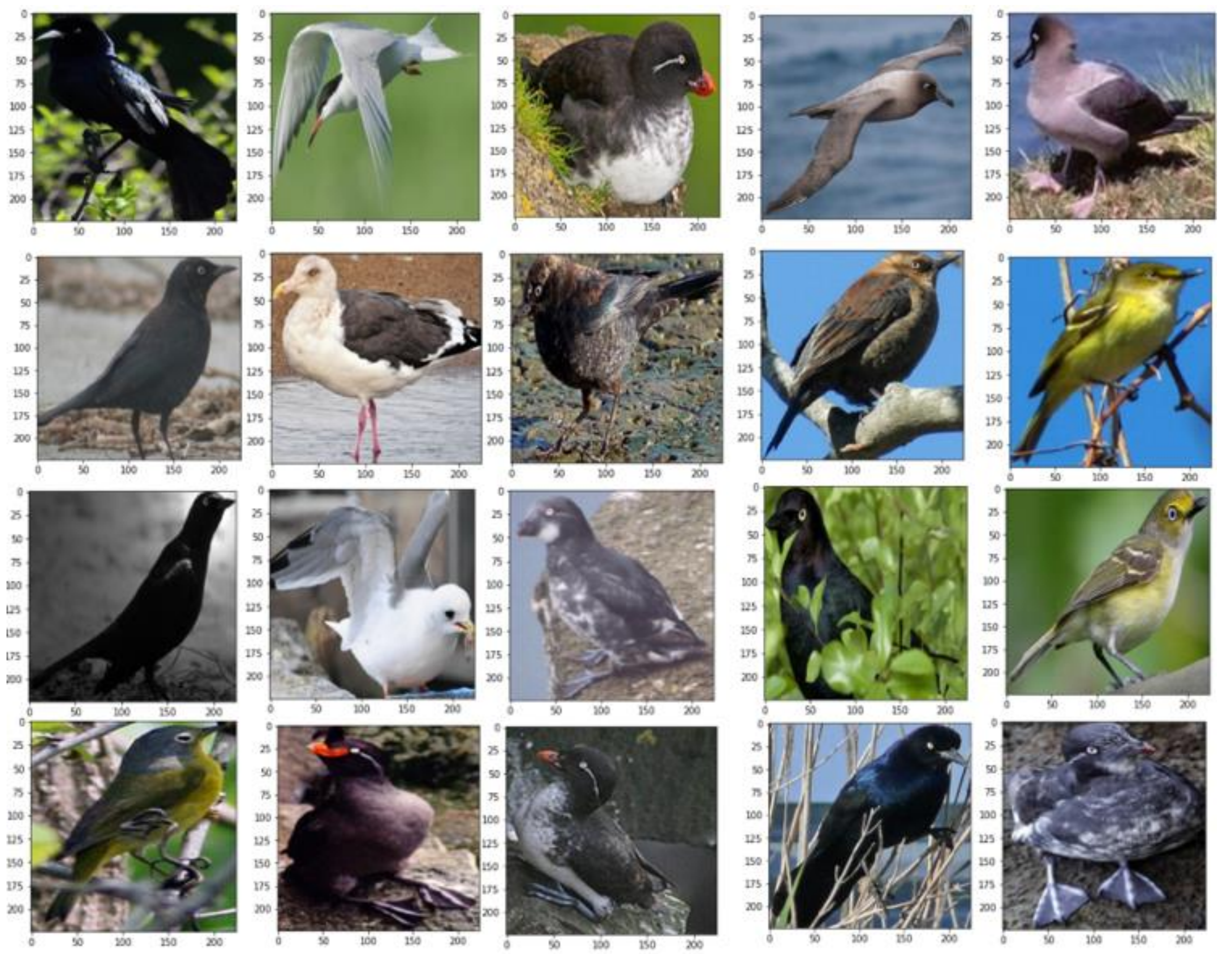


Figure 4. 8: Random samples of “white-eye” birds. It can be observed that “white eye” is the only common feature among these training images.

There are in total 2048 hidden filters in the last convolutional layer, and their activations are all recorded while feeding each selected bird image into the framework. Hence, it forms a 2048×20 matrix in pattern space. Each row of such matrix forms a pattern space vector, and it corresponds to the activations of a hidden filter throughout the training. If there are a pair or a group of pattern space vectors that have a small vector angle (smaller than 15 degrees), it implies that the pair or hidden filters are generating similar outputs for each input pattern, and hence, they indicate a latent attribute in the input images that results in similar activation of the pair of filters. Furthermore, multiple pairs of similar filters imply that there are multiple latent attributes causing redundant activations.

Hidden Filter 1	Magnitude of filter 1	Hidden Filter 2	Magnitude of filter 2	Angular distance (in degree)
5	4.27	**152**	6.93	13.62
119	6.87	**1108**	4.72	13.62
177	3.17	**593**	4.37	9.91
279	3.05	**1987**	1.81	14.42
395	8.68	**836**	6.86	13.91
492	3.69	**1283**	4.71	14.29

522	4.33	**1513**	5.10	14.84
581	1.78	**1902**	3.15	10.75
584	7.13	**1305**	4.48	11.72
588	6.68	**657**	2.17	13.13
588	6.68	**1824**	7.12	12.15
604	4.85	**1703**	3.21	13.40
657	2.17	**1592**	6.11	14.46
657	2.17	**1597**	5.18	13.59
657	2.17	**1824**	7.12	7.05
722	2.55	**1541**	6.86	14.82
805	5.33	**1051**	5.80	14.09
836	6.86	**1943**	2.12	14.24
850	3.49	**1180**	3.02	12.82
1127	6.91	**1989**	5.31	14.55
1133	3.27	**1181**	7.17	14.71
1404	6.20	**1544**	6.70	14.45
1571	3.40	**1717**	5.76	13.54
1592	6.11	**1597**	5.18	11.16
1592	6.11	**1824**	7.12	14.09
1597	5.18	**1824**	7.12	13.24
1824	7.12	**1943**	2.12	14.19

Table 4. 4: Pairs of last conv-layer filters that have similar pattern space vectors. The hidden filters are labelled by its sequential number in the conv-layer. The magnitude columns indicate the magnitude of pattern space vectors and the angle column indicate the angular separation between vectors. There are many more pairs of similar filters than expected, implying latent features are hidden in the pattern space.

Table 4.4 demonstrates pairs of last conv-layer filters, whose pattern space vectors have small angular separations. A large number of similar filters reveal that there are many latent patterns in the training set that may cause redundant activations. Moreover, the magnitude of each pattern space vector implies the strength of noises added by these redundant activations. Figure 4.9 demonstrates the magnitude of all 2048 pattern space vectors.

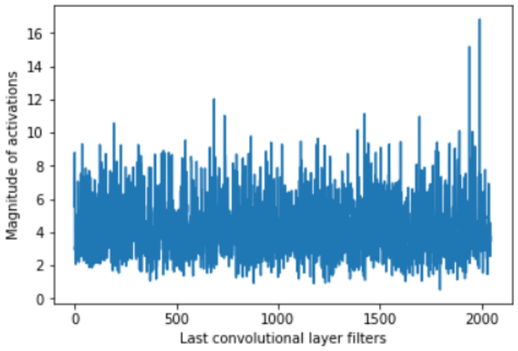


Figure 4. 9: Magnitude of all 2048 pattern space vectors. A higher magnitude means a stronger activation during training.

4.2.3 Future improvements

As discussed above, there are two significant limitations of the Bayesian framework; weak detection performance on bird legs/wings and unexpected redundant patterns in training set. Correspondingly, two suggestions can be provided in order to improve the performance of attribute detection:

1) Refine the labels from CUB. Due to the high complexity of feet/wing patterns, CUB labels can be refined and for example, the attribute label “wing colour: blue” can be divided into “opening wing colour: blue” and “closing wing colour: blue”. Thus, it can force the framework to learn the attributes with a more consistent manner.
2) Investigate latent patterns from pattern space vectors. In our experiment, we convert the activations of hidden filters into pattern space, with a single attribute as a target feature. However, more latent patterns can be revealed if we input the entire training space into the framework. The basic idea is to compute the vector angles of activations with all different visual attributes fed as target features. Thus, similar activations among different attributes can be located and removed in order to reduce the noises in feature detection.

Chapter 5 Conclusion

In this thesis, we establish a 3D modelling system for the digitisation of small-scale target objects. Moreover, this system is capable of detecting visual attributes from captured images to achieve an automated archiving and indexing of natural collections. First, we use a macro lens to capture small-scale insect bodies, and the captured images are blended by calibration and fusion algorithms. We evaluate the performance of different image fusion algorithms, and it is concluded that the saliency measurement method with dilation is more applicable in the 3D modelling system, as it generates a decent blending result in tolerable time. For the automated archiving and indexing of natural collections, we employ a Bayesian Inference framework. Here, the pixel location of the target visual attribute is detected. According to the analysis of the detection result, we summarise that the network cannot generate accurate feature detection on some attributes, including wings or legs, as their natural poses confuse the framework. Meanwhile, there are latent patterns that can be viewed only by the CNN black box and these patterns bias the detection performance.

Future work for this project includes applying tilt-shift lens to reduce camera vibration during motion. Calibration errors are introduced by random vibration, and a tilt-shift lens can effectively reduce the vibration by splitting the camera body and lens. Furthermore, the feature detection performance can be improved by computing the vector angles of filter activations and performing network pruning. Thus, the latent attribute in pattern space can be removed to reduce noise in feature detection.

Bibliography

Bau, D. et al., 2017. *Network Dissection: Quantifying Interpretability of Deep Visual Representations.* [Online]
Available at: https://arxiv.org/abs/1704.05796
[Accessed 25 March 2019].

Bolles, R. C., Baker, H. H. & Marimont, D. H., 1987. Epipolar-plane image analysis: An approach to determining structure from motion. *International Journal of Computer Vision,* March, 1(1), pp. pp 7-55.

Brecko, J. et al., 2014. *Focus stacking: Comparing commercial top-end set-ups with a semi-automatic low budget approach. A possible solution for mass digitization of type specimens.* [Online]
Available at: https://zookeys.pensoft.net/articles.php?id=4346
[Accessed 29 March 2019].

Brostow, G. J., Shotton, J., Fauqueur, J. & Cipolla, R., 2008. Segmentation and Recognition Using Structure from Motion Point Clouds. *European Conference on Computer Vision,* pp. pp 44-57.

Chollet, F., 2018. *Keras,* s.l.: GutHub.

Dosoviskiy, A. & Brox, T., 2015. *Inverting Visual Representations with Convolutional Networks.* [Online]
Available at: https://arxiv.org/abs/1506.02753
[Accessed 25 March 2019].

Franco, J.-S. & Boyer, E., 2003. *Exact polyhedral visual hulls.* [Online]
Available at: https://hal.inria.fr/inria-00349075
[Accessed 29 March 2019].

Guo, P., Anderson, C., Pearson, K. & Farrell, R., 2018. *Neural Network Interpretation via Fine Grained Textual Summarization.* [Online]
Available at: https://arxiv.org/abs/1805.08969v1
[Accessed 26 March 2019].

Haro, G., 2014. *Shape from silhouette consensus and photo-consistency.* [Online]
Available at: https://ieeexplore.ieee.org/abstract/document/7025980
[Accessed 29 March 2019].

He, K., Sun, J. & Tang, X., 2013. Guided Image Filtering. *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE,* 35(6), pp. 1397 - 1409.

He, K., Zhang, X., Ren, S. & Sun, J., 2015. *Deep Residual Learning for Image Recognition.* [Online]
Available at: https://arxiv.org/abs/1512.03385
[Accessed 26 March 2019].

Hossny, M., Nahavandi, S. & Creighton, D., 2008. Comments on 'Information measure for performance of image fusion. *Electronics letters,* 44(18), pp. 1066-1067.

Hudsn, L. N. et al., 2015. *Inselect: Automating the Digitization of Natural History Collections.* [Online]
Available at: https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0143402
[Accessed 28 March 2019].

Ijiri, T. et al., 2018. *Digitization of natural objects with micro CT and photographs.* [Online]
Available at: https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0195852
[Accessed 28 March 2019].

Kumar, B. S., 2015. Image fusion based on pixel significance using cross bilateral filter. *Signal Image and Video Processing,* July, 9(5), p. 1193/1204.

Li, S., Kang, X. & Hu, J., 2013. Image Fusion With Guided Filtering. *IEEE Transactions on Image Processing,* 22(7), pp. 2864-2875.

Loffe, S. & Szegedy, C., 2015. *Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift.* [Online]
Available at: https://arxiv.org/abs/1502.03167
[Accessed 26 March 2019].

Loreau, M. et al., 2001. Biodiversity and Ecosystem Functioning: Current Knowledge and Future Challenges. *Science,* 26 Octorber, 294(5543), pp. 804-808.

Mathys, A., Brecko, J. & Semal, P., 2013. Comparing 3D digitizing technologies: What are the differences?. *IEEE Xplore,* 28 Oct.pp. 201-204.

Mathys, A. et al., 2015. *Bringing collections to the digital era three examples of integrated high resolution digitisation projects.* Granada, Spain, IEEE.

Mouillot, D. et al., 2012. A functional approach reveals community responses to disturbances. *Trends in Ecology&Evolution,* 09 Nov, 28(3), pp. 167-177.

Nguyen, C. V., Lovell, D. R., Adcock, M. & Salle, J. L., 2014. *Capturing Natural-Colour 3D Models of Insects for Species Discovery and Diagnostics.* [Online]
Available at: https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0094346
[Accessed 28 March 2019].

Russakovsky, O. et al., 2014. *ImageNet Large Scale Visual Recognition Challenge.* [Online]
Available at: https://arxiv.org/abs/1409.0575
[Accessed 26 March 2019].

Selvaraju, R. R. et al., 2016. *https://arxiv.org/abs/1512.04150.* [Online]
Available at: https://arxiv.org/abs/1610.02391
[Accessed 26 March 2019].

Simonyan, K. & Zisserman, A., 2014. *Very Deep Convolutional Networks for Large-Scale Image Recognition.* [Online]
Available at: https://arxiv.org/abs/1409.1556
[Accessed 26 March 2019].

Srivastava, R. K., Greff, K. & Schmidhuber, J., 2015. *Highway Networks.* [Online]
Available at: https://arxiv.org/abs/1505.00387
[Accessed 26 March 2019].

Ströbel, B., Schmelzle, S., Blüthgen, N. & Heethoff, M., 2018. *An automated device for the digitization and 3D modelling of insects, combining extended-depth-of-field and all-side multi-view imaging.* [Online]
Available at: https://zookeys.pensoft.net/article/24584/element/4/430//
[Accessed 28 March 2019].

Tatsuta, H., Takahashi, K. H. & Sakamaki, Y., 2017. *Geometric morphometrics in entomology: Basics and applications.* [Online]
Available at: https://doi.org/10.1111/ens.12293
[Accessed 28 March 2019].

Welinder, P. et al., 2011. *Caltech-UCSD Birds 200.* [Online]
Available at: http://www.vision.caltech.edu/visipedia/papers/WelinderEtal10_CUB-200.pdf
[Accessed 04 May 2019].

Westoby, M. et al., 2012. *'Structure-from-Motion' photogrammetry: A low-cost, effective tool for geoscience applications.* [Online]
Available at: https://www.sciencedirect.com/science/article/pii/S0169555X12004217
[Accessed 29 March 2019].

Xiang, Y. et al., 2016. *3D Model Generation of Cattle by Shape-from-Silhouette Method for ICT Agriculture.* [Online]
Available at: https://ieeexplore.ieee.org/abstract/document/7791955/authors#authors
[Accessed 29 March 2019].

Zhang, Q. et al., 2017. *Computer Science > Computer Vision and Pattern Recognition.* [Online]
Available at: https://arxiv.org/abs/1708.01785
[Accessed 26 March 2019].

Zhang, Q. et al., 2018. *Explanary Graph for CNNs.* [Online]
Available at: https://arxiv.org/abs/1812.07997
[Accessed 25 March 2019].

Zheng, Y., Essock, E. A., Hansen, B. C. & Haun, A. M., 2007. A new metric based on extended spatial frequency and its application to DWT based fusion algorithms. *Information Fusion,* 8(2), pp. 177-192.

Zhou, B. et al., 2015. *Learning Deep Features for Discriminative Localization.* [Online]
Available at: https://arxiv.org/abs/1512.04150
[Accessed 26 March 2019].