

CLASSIFICATION OF AIRCRAFT WITH SHADOW USING KEYPOINT DESCRIPTOR FROM  
REMOTE SENSING IMAGES



A Thesis Submitted in Partial Fulfillment of the Requirements  
for the Degree of Master of Science in Applied Mathematics and Computational Science  
Department of Mathematics and Computer Science  
FACULTY OF SCIENCE  
Chulalongkorn University  
Academic Year 2019  
Copyright of Chulalongkorn University

การจำแนกประเภทของอากาศยานที่มีเงาด้วยตัวบอกคีย์พอยต์จากภาพรับรู้ระยะไกล



วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต  
สาขาวิชาคณิตศาสตร์ประยุกต์และวิทยาการคอมพิวเตอร์ ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์  
คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย  
ปีการศึกษา 2562  
ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

Thesis Title CLASSIFICATION OF AIRCRAFT WITH SHADOW USING  
KEYPOINT DESCRIPTOR FROM REMOTE SENSING IMAGES  
By Miss Natnicha Meeboonmak  
Field of Study Applied Mathematics and Computational Science  
Thesis Advisor Associate Professor NAGUL COOHAROJANANONE, Ph.D.

---

Accepted by the FACULTY OF SCIENCE, Chulalongkorn University in Partial  
Fulfillment of the Requirement for the Master of Science

..... Dean of the FACULTY OF SCIENCE  
(Professor POLKIT SANGVANICH, Ph.D.)

THESIS COMMITTEE

..... Chairman  
(Associate Professor PETARPA BOONSERM, Ph.D.)

..... Thesis Advisor  
(Associate Professor NAGUL COOHAROJANANONE, Ph.D.)

..... Examiner  
(Associate Professor RAJALIDA LIPIKORN, Ph.D.)

..... Examiner  
(Assistant Professor KRUNG SINAPIROMSARAN, Ph.D.)

..... External Examiner  
(Suriya Natsupakpong, Ph.D.)



# # 6071932223 : MAJOR APPLIED MATHEMATICS AND COMPUTATIONAL SCIENCE

KEYWORD: aircraft classification, remote sensing image, deeply supervised network, short connection, keypoint selection, Fourier descriptor, centroid contour distance, random forest, image processing

Natnicha Meeboonmak : CLASSIFICATION OF AIRCRAFT WITH SHADOW USING KEYPOINT DESCRIPTOR FROM REMOTE SENSING IMAGES. Advisor: Assoc. Prof. NAGUL COOHAROJANANONE, Ph.D.

Presently, aircraft classification from remote sensing images is widely used in military and civilian. However, aircraft type classification is still challenging to discriminate. The reason is that the aircraft displayed in the image has different in color, shape, size, and orientation. Moreover, there is the shadow appeared over and behind the aircraft that obscures some details of aircraft. Therefore, this work proposes the classification of aircraft with shadow method. This method modifies the deeply supervised salient object detection with short connections to make the aircraft be outstanding from the shadow and other objects in the background and then segment it. The segmentation experiment performs on our dataset, which is divided into two categories: simple case and difficult case. The results show that our proposed method performs well in both cases. In order to classify the types of aircraft, we extract features from each keypoint of aircraft segmented image by using the Fourier descriptor based on centroid contour distance, which is invariant to position, scale, and rotation. After that, we use the random forest algorithm to classify the aircraft types into six classes in the classification experiment. The result shows that the method can efficiently classify the aircraft and work well on aircraft images with shadow.

Field of Study: Applied Mathematics and Computational Science      Student's Signature .....

Academic Year: 2019      Advisor's Signature .....

## ACKNOWLEDGEMENTS

First and foremost, I would like to express my sincere gratitude to my advisor, Associate Professor Nagul Cooharajanone, for his patient guidance and helpful suggestions throughout my master's degree. Without his persistent help, this work could not have accomplished its goal.

My grateful are extended to the thesis committee: Associate Professor Petarpa Boonserm, Associate Professor Rajalida Lipikorn, Assistant Professor Krung Sinapiromsaran, and Dr. Suriya Natsupakpong, for reviewing this thesis book and giving valuable comments.

I would also like to acknowledge the Development and Promotion of Science and Technology Talent project (DPST) for providing all the financial support.

Moreover, I would like to thank my friends in Applied Mathematics and Computational Science Program and colleagues in Machine Intelligence and Multimedia Information Technology Laboratory (MIMIT) for sharing useful knowledge.

Finally, I would like to offer my special thanks to my beloved parents for giving important support and encouragement throughout my study.

## TABLE OF CONTENTS

	Page
.....	iii
ABSTRACT (THAI).....	iii
.....	iv
ABSTRACT (ENGLISH).....	iv
ACKNOWLEDGEMENTS.....	v
TABLE OF CONTENTS.....	vi
LIST OF TABLES.....	ix
LIST OF FIGURES.....	x
CHAPTER I Introduction.....	1
1.1 Objective.....	7
1.2 Scopes and Assumptions.....	7
CHAPTER II Background Knowledge.....	8
2.1 Image processing.....	8
2.1.1 Elementwise AND operation.....	8
2.1.2 Binary image thresholding.....	9
2.1.3 Dilation.....	10
2.2 Convolutional Neural Network.....	12
2.2.1 Convolution operation.....	13
2.2.2 Max pooling.....	14
2.2.3 Rectified Linear Unit.....	14
2.2.4 Fully connected layer.....	15

2.2.5 Softmax.....	16
2.2.6 VGG16.....	16
2.3 Deeply supervised salient object detection with short connections.....	17
2.3.1 Base convolutional layer.....	18
2.3.2 Side-output layer.....	18
2.3.3 Short-connection.....	19
2.3.4 Fusion layer.....	20
2.4 Keypoint selection.....	21
2.5 Boundary feature descriptor.....	24
2.5.1 Centroid contour distance.....	24
2.5.2 Fourier descriptor based on centroid contour distance.....	25
2.6 Random Forest.....	26
2.6.1 Decision Tree.....	26
2.6.2 Entropy.....	27
2.6.3 Information Gain.....	27
2.6.4 Gini Index.....	27
2.6.5 Random Forest.....	28
CHAPTER III Proposed Method.....	29
3.1 Segmentation.....	30
3.1.1 Modified Deeply Supervised Salient Object Detection with Short Connections.....	30
3.2 Feature extraction.....	32
3.3 Classification.....	33
CHAPTER IV Results and Discussion.....	34



4.1 Segmentation test.....	34
4.2 Classification test.....	38
CHAPTER V Conclusions.....	43
5.1 Future work.....	44
REFERENCES.....	45
VITA.....	47



## LIST OF TABLES

	<b>Page</b>
Table 1 The truth table of the logical AND operator for a and b.....	8
Table 2 The details of channels and kernel sizes of convolutional layers in each side output.....	18
Table 3 The confusion matrix for a binary classification problem.....	37
Table 4 The performance comparison between DSS and modified DSS method .....	38
Table 5 The confusion matrix of classification result using the DSS network with FD-CCD and RF classifier.....	40
Table 6 The confusion matrix of classification result using the modified DSS network with FD-CCD and RF classifier .....	40
Table 7 Example of incorrect prediction results .....	41
Table 8 The Accuracy of the DSS and modified DSS method with FD-CCD and RF classifier .....	42

## LIST OF FIGURES

	Page
Figure 1 Example of aircraft from remote sensing images .....	2
Figure 2 Example of aircraft with shadow from remote sensing images .....	3
Figure 3 The architecture of the HED network.....	4
Figure 4 Example patterns of the DSS network .....	5
Figure 5 Example results of using DSS method with the aircraft image with shadow... 6	6
Figure 6 The logical AND operation of A and B.....	9
Figure 7 (a) A grayscale input image, (b) the intensity histogram of (a), and (c) the thresholding result of (a) using $T=127$ .....	10
Figure 8 Example results of dilation A by two different B.....	11
Figure 9 The visualization of dilation A by B .....	12
Figure 10 A general architecture of CNN.....	13
Figure 11 Example of convolution operation with 3x3 kernel.....	13
Figure 12 Example of max pooling using 2x2 filters and stride 2 .....	14
Figure 13 The visualization of ReLU activation function.....	15
Figure 14 Example of softmax calculation .....	16
Figure 15 The architecture of VGG16.....	17
Figure 16 The framework of the DSS architecture.....	17
Figure 17 Example of short connections structure.....	19
Figure 18 (a) Arbitrary shape with N boundary points and (b) included angle at point $p_i$ .....	22
Figure 19 The illustration of complex shape and its moment graphs .....	23
Figure 20 The illustration of distance from contour point to centroid point $P_m$ .....	24

Figure 21 The structure of the decision tree .....	26
Figure 22 The illustration of the random forest structure .....	28
Figure 23 The framework of our proposed method .....	29
Figure 24 The framework of the modified DSS architecture .....	30
Figure 25 Example of each side-output result from the DSS network .....	32
Figure 26 The post-processing framework.....	32
Figure 27 Example images from (a) the simple case and (b) the difficult case .....	35
Figure 28 The visualization of each side-output result from the DSS network.....	35
Figure 29 The visualization of each side-output result from the modified DSS network .....	35
Figure 30 The comparison results between DSS method and modified DSS method	36
Figure 31 Example segmentation images of failure cases.....	36
Figure 32 Example images of 6 aircraft types used in this work. (a) Boeing, (b) B-1, (c) B-2, (d) B-52, (e) E-3, and F-22 .....	39

## CHAPTER I

### Introduction

In recent years, the development of remote sensing technology has been increasing rapidly. Large amounts of remote sensing images are obtained and used in many computer vision areas such as detection, recognition, and classification. Object classification from remote sensing images is one of the most useful tasks in the computer vision field. It is used to classify different objects from the given images into one of the finite sets of classes.

Aircraft type classification is a beneficial work in the object classification field. This task can protect civilians from air raids by planning to prevent the attack using image analyzation data to determine the type of adversary's aircraft. Besides, the task can also be used in airport management for civil purposes. Nevertheless, this task is challenging due to the various shape, color, and position of aircraft engines such as jet engines, spoilers, and flaps. Moreover, the captured images from different locations contain different aircraft that vary in size, rotation, and posture. In addition, other objects in the background, such as cars, aircraft boarding ramps, and containers, cause more difficulty to classify. Figure 1 shows different aircraft images from our dataset.



*Figure 1 Example of aircraft from remote sensing images*

There are many methods used for differentiating categories of objects. Generally, image segmentation, which is a process of clustering the pixels into some groups, is used before feature extraction to separate aircraft from the background. It can solve the various colors of aircraft problem and the other objects in the background problem. In 2011, A. G. Karacor et al. [1] used image processing techniques such as gray-scaling, morphological structuring, and Sobel operator to fill gaps in the background and emphasize edges in the segmentation stage. The obtained segmented image was used to classify four types of military aircraft from an aircraft's side view in the sky by their artificial neural network.

However, the aircraft classification from the top view of an aircraft on the ground is more complicated because the image sometimes includes various texture locations and noise. In 2016, X. Zhu et al. [2] proposed work that aims to identify three types of aircraft parked on the ground. Their proposed method used the optimization bag of words based on space block SIFT sampling and Fuzzy C-Means clustering as a feature. The support vector machine was then performed to distinguish different aircraft.

The most challenging problem is a shadow, a dark area originating from light rays blocked by opaque objects. It often appears behind and over the aircraft. Figure 2 shows the example images of aircraft with shadow from our dataset. Sometimes their shades and the color of marking on the side of fuselage are similar. Furthermore, it probably covers engines and some parts of the body. These reasons make the segmentation method confuse and cause problems to the classification result.

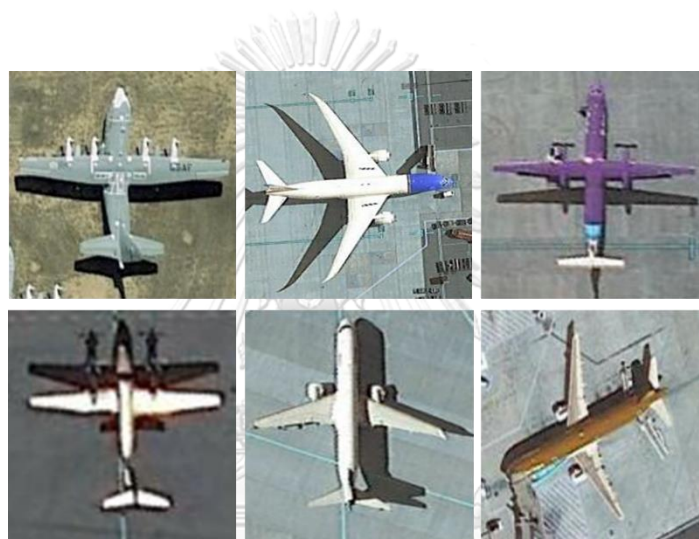


Figure 2 Example of aircraft with shadow from remote sensing images

จุฬาลงกรณ์มหาวิทยาลัย  
CHULALONGKORN UNIVERSITY

In the past few years, deep learning plays a significant role in many researches because it achieves better performance than other techniques. The salient object detection is one of the efficient methods based on deep learning. This method aims to extract the most distinctive objects from an image. Accordingly, the method is suitable for creating segmented aircraft images with shadow in the segmentation step because it can make the aircraft body outstanding from the shadow, background, and other objects.

In 2015, S. Xie and Z. Tu [3] proposed edge detection called holistically-nested edge detection (HED), based on CNNs and deeply-supervised networks. The HED network is illustrated in Figure 3. In this network, the last convolutional layer from each stage of the primary deep network directly connects to each side output, and the fusion layer is then added to capture the features of each side output. Afterward, in 2018, Hou et al. [4] proposed the deeply supervised salient object detection with short connections (DSS), which uses the skip-layer structure with deep supervision ideas from HED for saliency detection. Furthermore, they designed the link between the deep and shallow side-output layers within the original HED architecture to obtain better saliency maps. DSS, therefore, locates the salient region better than the previous method. Figure 4 illustrates the examples of two different DSS architecture patterns.

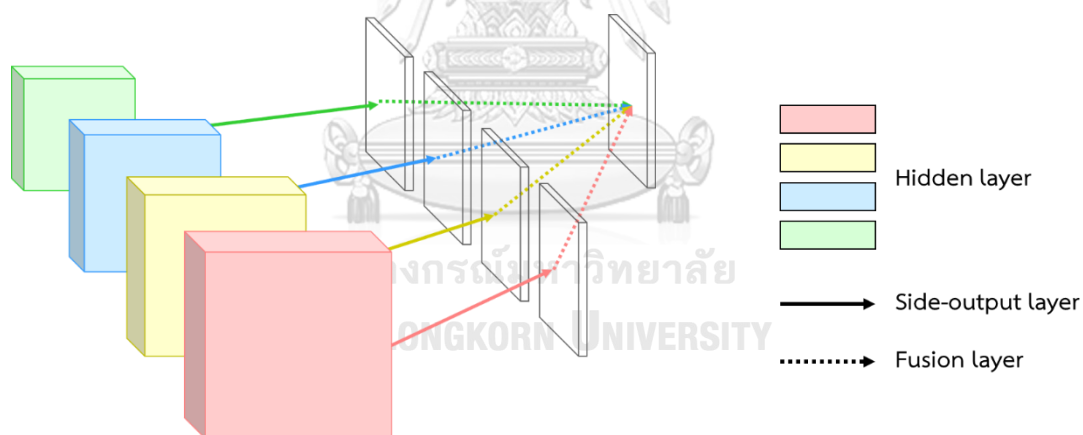


Figure 3 The architecture of the HED network



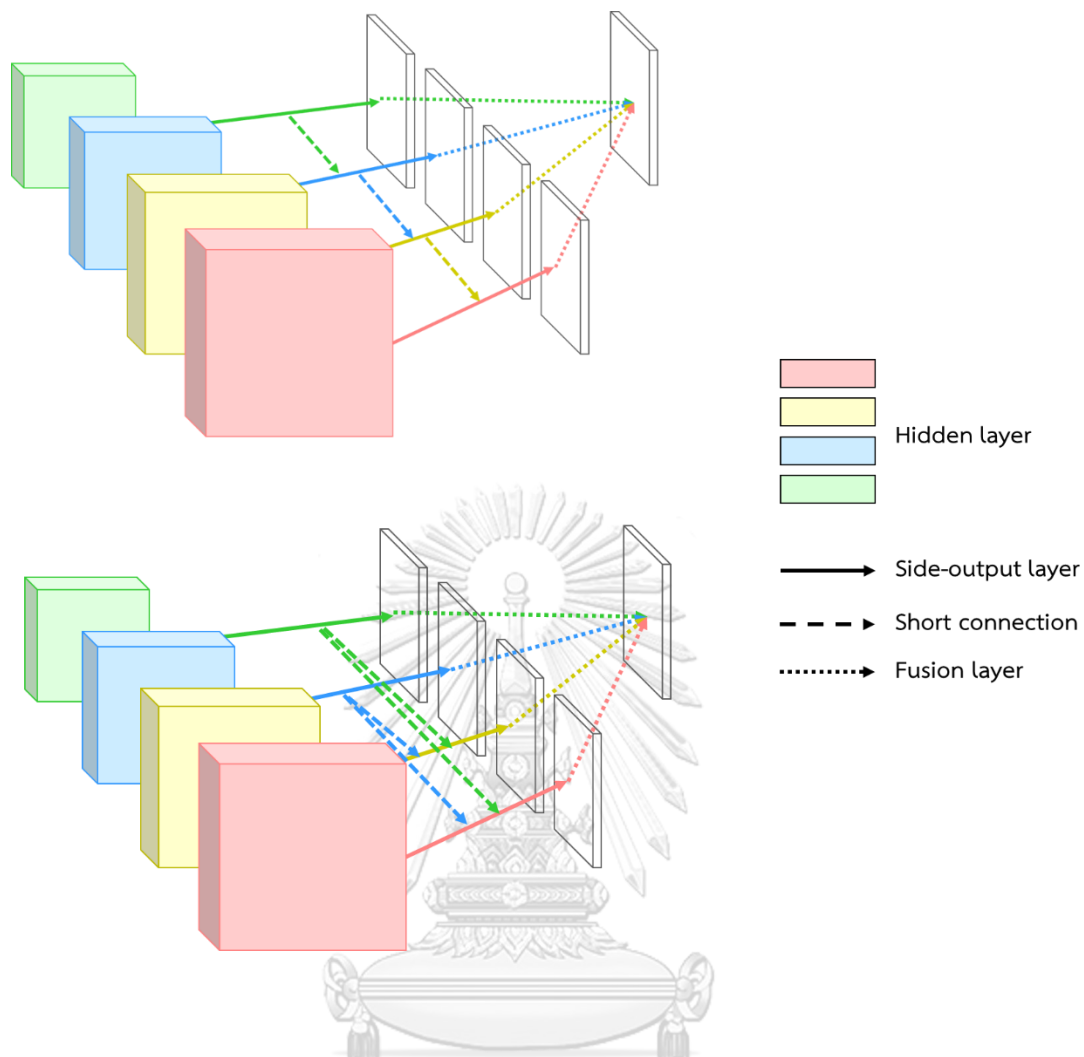


Figure 4 Example patterns of the DSS network

After implementing the DSS algorithm with the aircraft images with shadow, we found that the method can segment the aircraft image, but the shadow and objects in the background still affect segmentation results, as shown in Figure 5. Moreover, we notice that the saliency map received from the shallowest side output most affects the result. Accordingly, in this research, we propose the aircraft classification method to classify aircraft types from remote sensing images of aircraft with shadow. The method modifies the DSS network to make the result more salient by increasing the shallowest saliency map's performance. We take advantage of

L2-Normalization as presented in [5] by adding the L2-Normalization layer between the connection of conv1\_2 from the primary network and the shallowest side-output layer. Also, we create a post-processing method to remove noise in the background.



*Figure 5 Example results of using DSS method with the aircraft image with shadow*

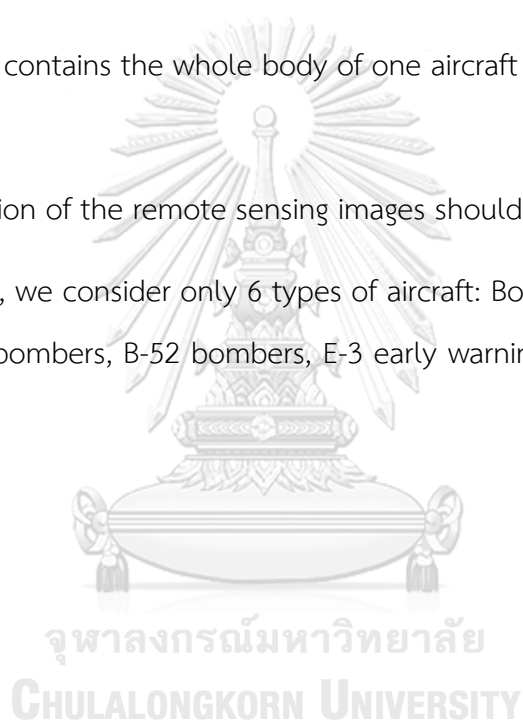
The feature extraction is also an important part of the object classification. Since each type of aircraft can easily classify by shape, the shape descriptor representing each object's feature by the boundaries is appropriate for using as our descriptor. To overcome the vary of size and rotation problems, we utilize the shape descriptors named Fourier descriptor based on centroid contour distance, which is invariant to both scale and rotation, in this work. Since the previous object segmentation focuses on the salient area, the aircraft's edges of the resulting segmented image might have noise. Hence, we select the keypoint boundaries that invariant to noise before extract feature. Finally, we apply a random forest algorithm to identify the type of aircraft.

## 1.1 Objective

To propose an algorithm which can classify types of aircraft with shadow from remote sensing images

## 1.2 Scopes and Assumptions

1. The remote sensing images used in this work are representations of the top view of aircraft.
2. Each image contains the whole body of one aircraft parked on the ground and visible.
3. The resolution of the remote sensing images should be at least 64x64 pixels.
4. In this work, we consider only 6 types of aircraft: Boeing passenger aircraft, B-1 bombers, B-2 bombers, B-52 bombers, E-3 early warning aircraft, and F-22 fighter aircraft.



## CHAPTER II

### Background Knowledge

#### 2.1 Image processing

##### 2.1.1 Elementwise AND operation

AND operator is one of the logical operators denoted as  $\wedge$ . It commonly applies to two binary images. Generally, each pixel in the binary images has a value of either **0** (background) or **1** (foreground). When employing the elementwise AND operator, it operates on corresponding pixel pairs between the images. This operator refers to the intersection where TRUE and FALSE are denoted as **1** and **0**, respectively. Table 1 shows the truth table defining the logical AND operator for two logical variables  $a$  and  $b$ . It yields **1** only when both  $a$  and  $b$  are **1**, and **0** otherwise. The logical AND operation of two regions  $A$  and  $B$ , defined in Table 1, is shown in Figure 6. The result is the set of elements that belong to both  $A$  and  $B$ .

Table 1 The truth table of the logical AND operator for  $a$  and  $b$

$a$	$b$	$a \wedge b$
0	0	0
0	1	0
1	0	0
1	1	1

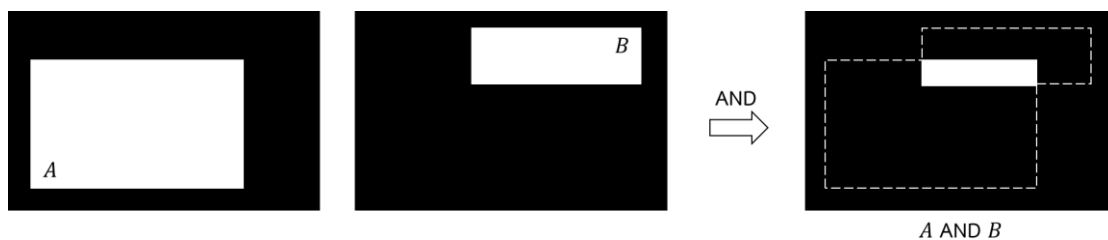


Figure 6 The logical AND operation of A and B

### 2.1.2 Binary image thresholding

Binary thresholding is a simple approach to divide an image into two classes of pixels. In other words, it separates foreground regions from the background region. Its input is typically a grayscale or color image, and the output is a binary image.

Suppose that  $f(x,y)$  is an input image, and the histogram displayed in Figure 7(b) is the intensity histogram corresponding to  $f(x,y)$ . In order to extract the light area (object point) from the dark area (background point), threshold  $T$  is manually selected. The segmented output image can be represented by  $g(x,y)$  as follows:

$$g(x,y) = \begin{cases} 1 & \text{if } f(x,y) > T \\ 0 & \text{if } f(x,y) \leq T \end{cases}$$

where  $T$  is a constant. Then, any point  $(x,y)$  in the image at which  $f(x,y) > T$  is called a foreground point. Oppositely, called a background point. Figure 7 is an example result of the binary image thresholding using  $T = 127$ .

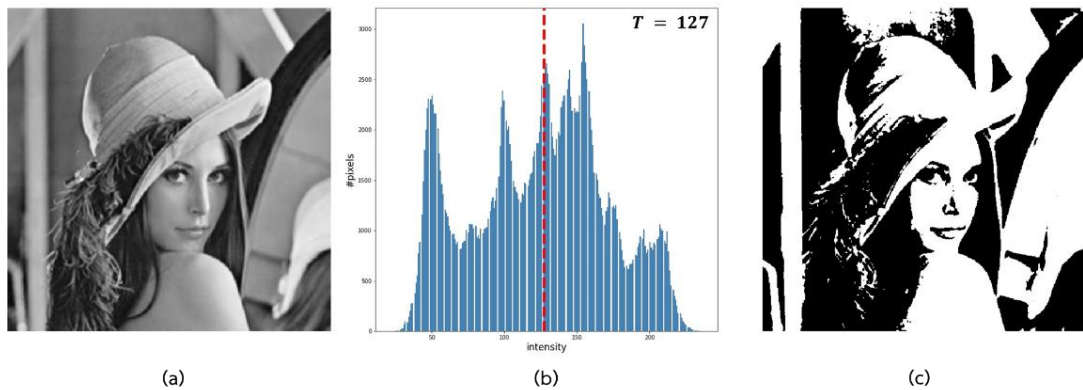


Figure 7 (a) A grayscale input image, (b) the intensity histogram of (a), and (c) the thresholding result of (a) using  $T=127$

### 2.1.3 Dilation

Dilation is one of the significant operations in morphological image processing. It usually works on grayscale and binary images. The effect of this operator is to expand the foreground areas slowly, and the small holes inside those areas are filled or become smaller. The objects in this image, therefore, become broader.

Let  $A$  be a set of foreground pixels and  $B$  is a structuring element. Both  $A$  and  $B$  are set of points in 2-D. The dilation of  $A$  by  $B$  is the set operation represented by  $A \oplus B$  as

$$A \oplus B = \{z | (\hat{B})_z \cap A \neq \emptyset\},$$

where  $\hat{B}$  refers to the reflection of a set  $B$  about its origin defined as

$$\hat{B} = \{w | w = -b, \text{ for } b \in B\},$$

and  $(B)_z$  associates with the translation of a set  $B$  by point  $z = (z_1, z_2)$  calculated by

$$(B)_z = \{c | c = b + z, \text{ for } b \in B\}.$$

The results of dilation set  $A$  of size  $d \times d$  by two different designs of  $B$ , square and rectangle, are illustrated in Figure 8. The dashed line represents the edge of the original object, and the dot in  $B$  represents the origin. Figure 9 shows the dilation result of text image  $A$  by structuring element  $B$ . The result shows that the boundaries of texts are thickened and the hole inside each character is small.

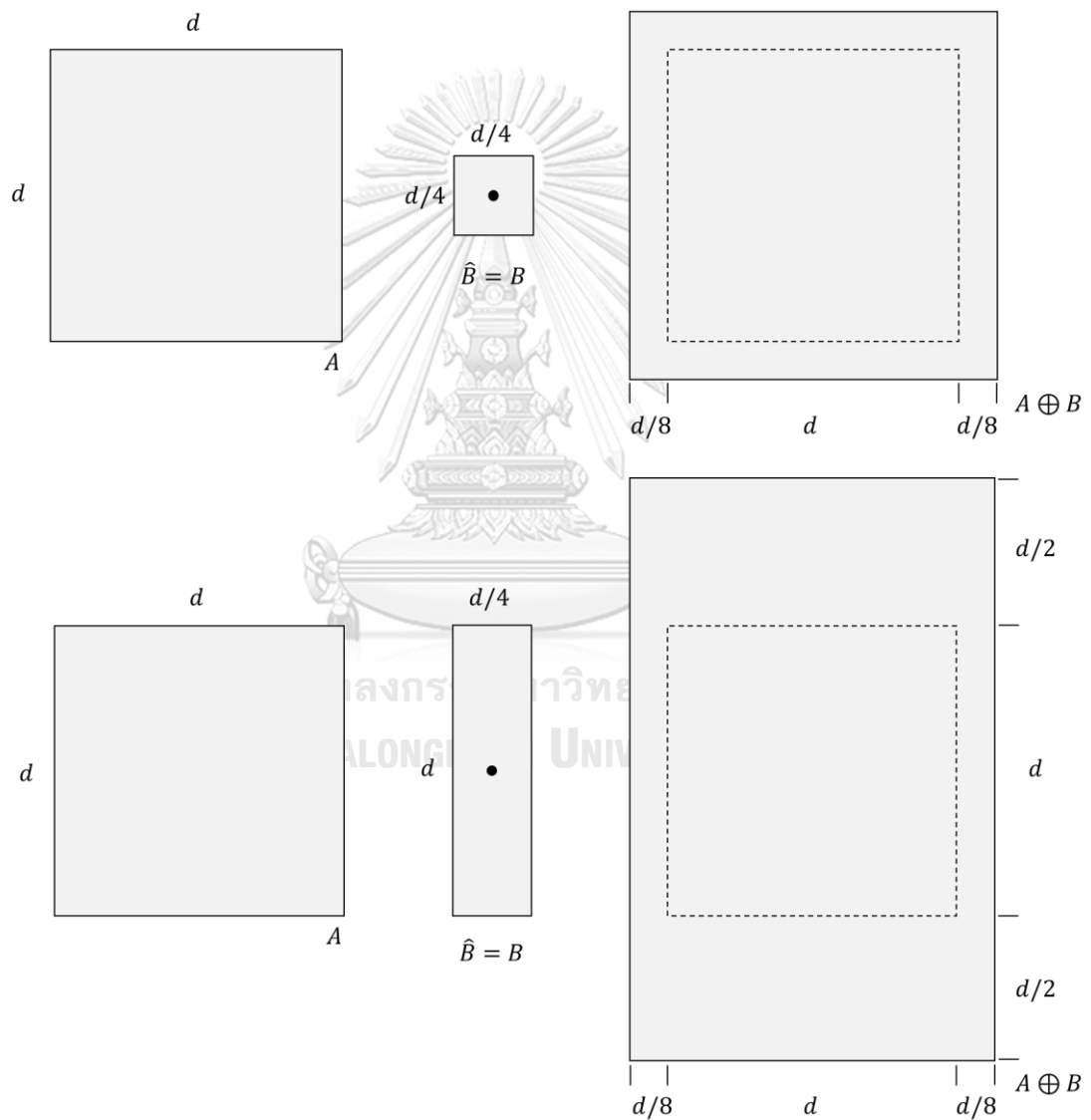


Figure 8 Example results of dilation  $A$  by two different  $B$

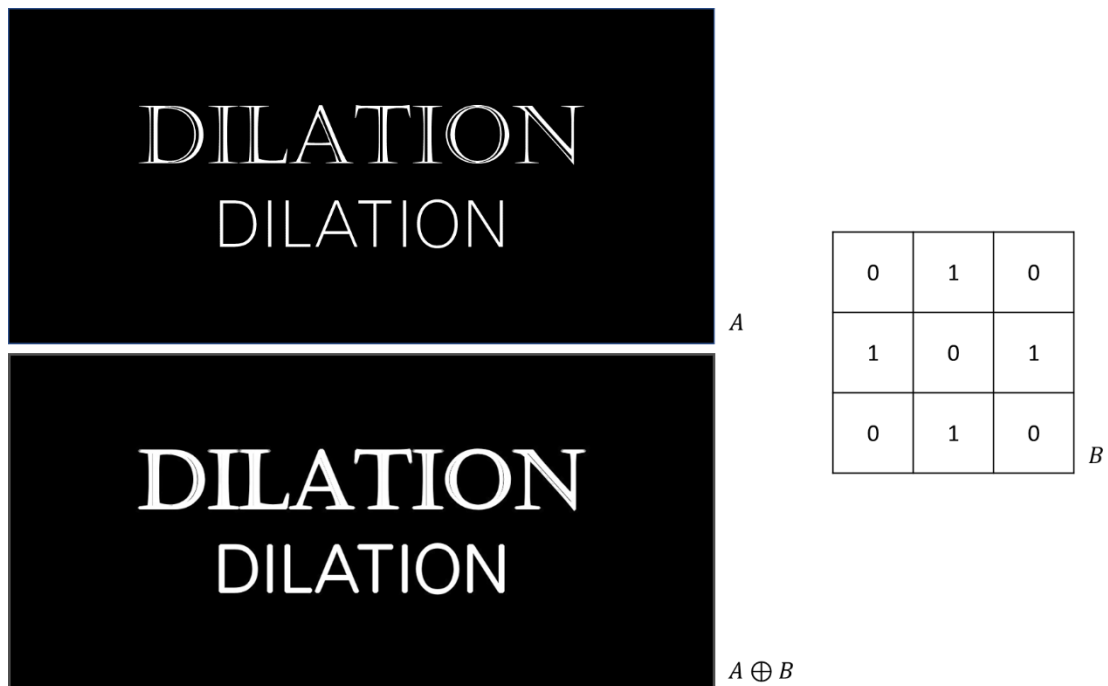


Figure 9 The visualization of dilation A by B

## 2.2 Convolutional Neural Network

Convolutional neural network (CNN), a type of deep learning network, is designed to process structured arrays of data such as images. CNN employs a mathematical operation called convolution in at least one of their layers. It is applied directly to an image's pixels for extracting data features and finding their hidden patterns.

CNN is multilayer perceptrons that used to build a mapping from the input to the target output. Each neuron in CNN is connected to the neuron in a local area of the previous layer. The general architecture of CNN, shown in Figure 10, consists of an input layer, convolutional layers alternated with pooling layers, and followed by fully connected layers and an output layer. The final output layer produces classification results.



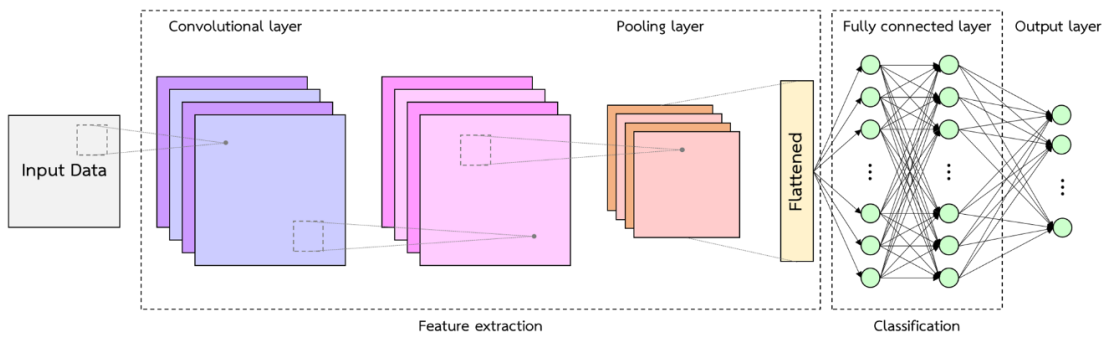


Figure 10 A general architecture of CNN

### 2.2.1 Convolution operation

The convolution is a mathematical operation apply to two functions of a real-valued argument. It operates on a local area of the image with a specific convolutional kernel size. The convolution of  $w$  and  $f$  at point  $(x, y)$  can be calculated by

$$(w * f)(x, y) = \sum_{s=-a}^a \sum_{t=-b}^b w(s, t) \cdot f(x + s, y + t),$$

where  $w$  is a kernel of size  $(2a + 1) \times (2b + 1)$ , and both  $a$  and  $b$  are non-negative integers.

Figure 11 shows the example result of convolution operation in 2-D with a kernel size  $3 \times 3$ .

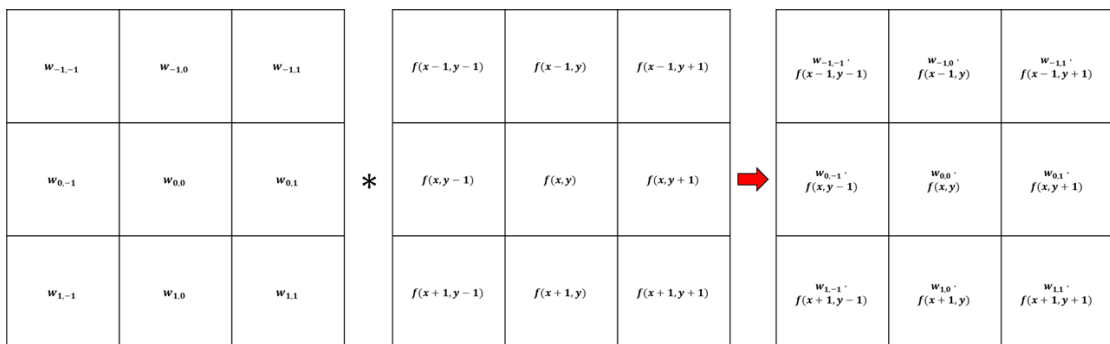


Figure 11 Example of convolution operation with  $3 \times 3$  kernel

### 2.2.2 Max pooling

Max pooling is a downsampling strategy on CNN. It is used to find the maximum value of the area that kernel convolves. The objective of max pooling is to reduce layer dimensionality and keep the features that are insensitive to their precise regions even when the input data is slightly shifted. Figure 12 shows the example result of max pooling operation with  $2 \times 2$  kernel size and stride 2 from  $4 \times 4$  input size.

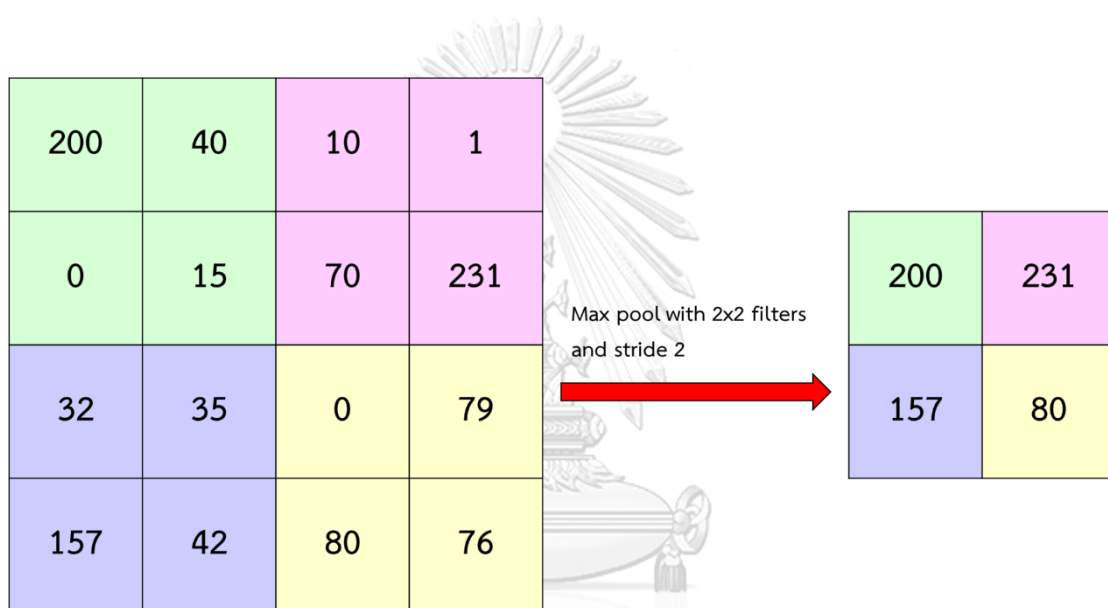


Figure 12 Example of max pooling using  $2 \times 2$  filters and stride 2

### 2.2.3 Rectified Linear Unit

The Rectified Linear Unit (ReLU) is the most frequently used activation function, which is the mathematical function that determines the output of a neural network. This function returns zero if the value is negative, but it returns that value for any positive value. The graphical of the ReLU activation function is shown in Figure 13. Therefore, ReLU is defined by

$$f(x) = \max(0, x).$$

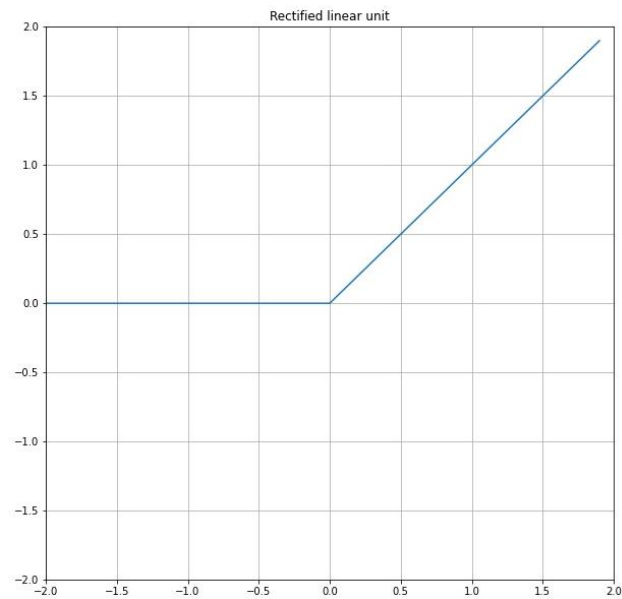


Figure 13 The visualization of ReLU activation function

The advantage of ReLU is that its computational cost is not too high. The forward and backward passes through the ReLU are both just a mere if statement. So, if we have to deal with the vast networks with many neurons, ReLU can help significantly reducing training time.

#### 2.2.4 Fully connected layer

A fully connected layer consists of multiple hidden layers. Each neuron in the hidden layer is fully interconnected with the neuron of the subsequent layer. The fully connected layer aims to map the vector from a feature map into a class probability distribution. The output layer uses a classification function such as softmax to output the classification results.

### 2.2.5 Softmax

The softmax function is a mathematical function that maps a vector of  $K$  real numbers and normalizes it into a vector of  $K$  probabilities that sum to 1. The softmax formula of  $\mathbf{z} = (z_1, \dots, z_K) \in \mathbb{R}^K$  for  $i = 1, \dots, K$  can be computed from

$$\sigma(\mathbf{z})_i = \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}},$$

where  $z_i$  is the element  $i$  of the input vector  $\mathbf{z}$ .

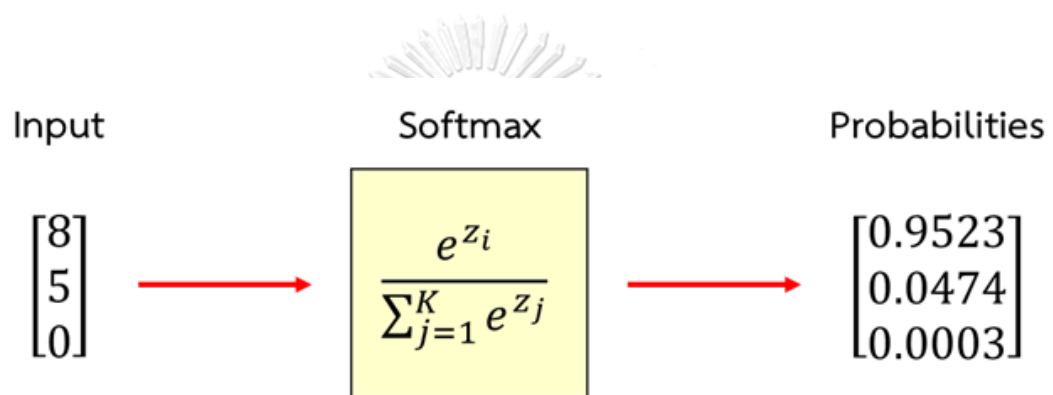


Figure 14 Example of softmax calculation

### 2.2.6 VGG16

VGG16 is a famous CNN model proposed by K. Simonyan and A. Zisserman [6]. This model is widely considered as one of the most famous architectures of CNNs because it works excellently in many computer vision tasks. VGG16 won first place in the 2014 ILSVRC challenge and achieved a 92.7% top-5 test accuracy in ImageNet, which contains over 14 million images belonging to 1,000 classes.

VGG16 architecture contains 16 principle layers; 13 convolutional layers with  $3 \times 3$  convolutional kernel followed by three fully connected layers. The architecture with order and size is illustrated in Figure 15.

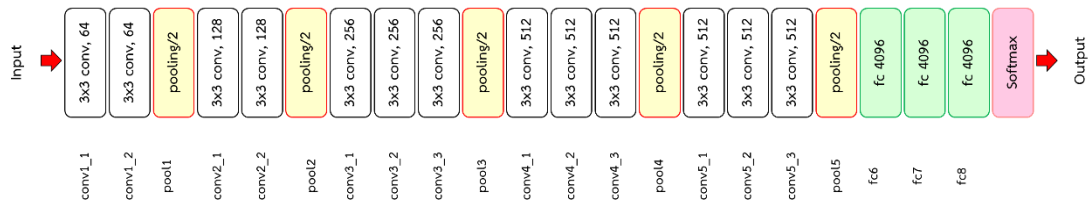


Figure 15 The architecture of VGG16

### 2.3 Deeply supervised salient object detection with short connections

The deeply supervised salient object detection with short connections (DSS) [4] is the framework for salient object detection. In general, CNN intends to generate the high-level features derived from the last fully connected layer. Therefore, it might ignore the low-level features, which are minor details of the image, such as edges and lines. To get the high-level features that provide better regions along with the low-level features, DSS builds some connections, called short connections, in the skip-layer structures introduced in [3]. The network of DSS mainly consists of 4 parts: 1) Base convolutional layer, 2) Side-output layers, 3) Short connection, and 4) Fusion layer. Figure 16 illustrates the architecture of the DSS network.

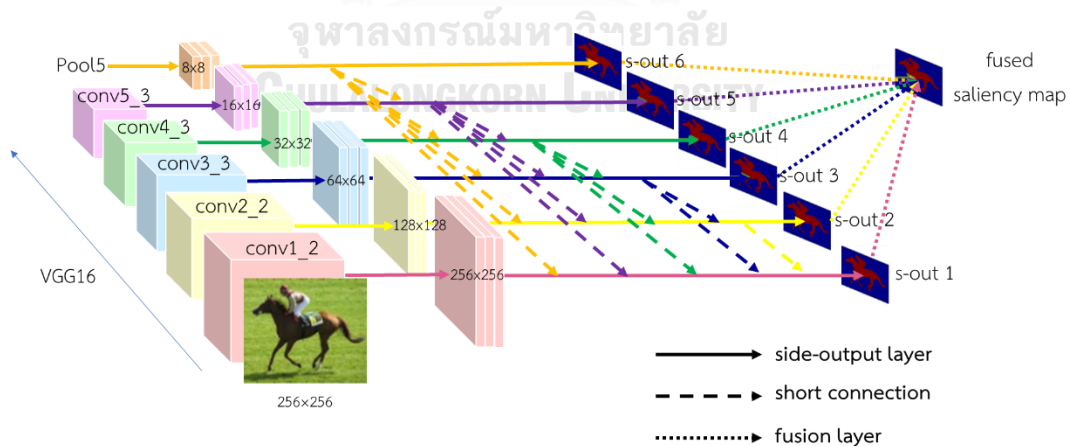


Figure 16 The framework of the DSS architecture

### 2.3.1 Base convolutional layer

In this layer, VGG16 is used as the primary CNN (the architecture of VGG16 is in section 2.2.6). In order to extract the saliency maps, the last obtained convolutional layer of each stage (conv1\_2, conv2\_2, conv3\_3, conv4\_3, and conv5\_3) and the last max-pooling layer (pool5) from the VGG16 network are used as input of the side-output layers.

### 2.3.2 Side-output layer

The architecture of the side-output layer, as illustrated with the solid line in Figure 16, consists of 3 convolutional layers, which have the number of channels and the size of the kernel shown in Table 2. In each side output, the first two convolutional layers are followed by ReLU.

*Table 2 The details of channels and kernel sizes of convolutional layers in each side output*

Layer	1	2	3
conv1_2	128, $3 \times 3$	128, $3 \times 3$	1, $1 \times 1$
conv2_2	128, $3 \times 3$	128, $3 \times 3$	1, $1 \times 1$
conv3_3	256, $5 \times 5$	256, $5 \times 5$	1, $1 \times 1$
conv4_3	256, $5 \times 5$	256, $5 \times 5$	1, $1 \times 1$
conv5_3	521, $5 \times 5$	521, $5 \times 5$	1, $1 \times 1$
pool5	521, $7 \times 7$	521, $7 \times 7$	1, $1 \times 1$

### 2.3.3 Short-connection

Among side-output layers, the deeper side-output layer is combined with the shallower side-output layer by short-connections, as displayed with the dashed line in Figure 16. Example of short connections structure is shown in Figure 17.

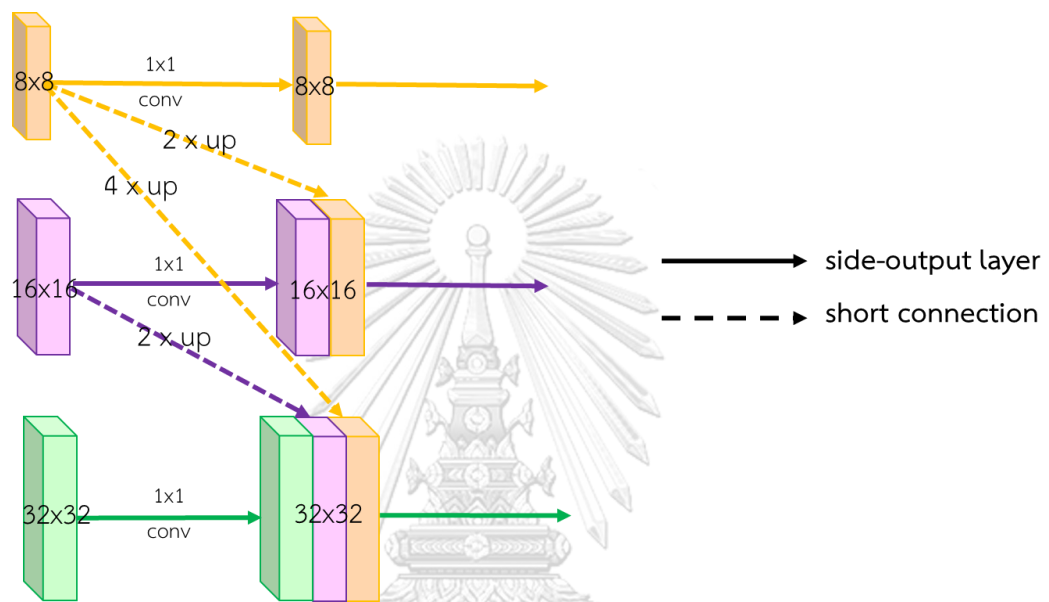


Figure 17 Example of short connections structure

Let  $X = \{x_j, j = 1, \dots, |X|\}$  is an input image, and  $Z = \{z_j, j = 1, \dots, |Z|\}$  is the ground truth saliency map for  $X$ , which is 1 for white pixel and 0 for black pixel. Then, the training data set is represented by  $T = \{(X_n, Z_n), n = 1, \dots, N\}$  where the number of images is  $N$ .

Given  $W$  be the collection of all standard network layer parameters and  $w = (w^{(1)}, w^{(2)}, \dots, w^{(M)})$  is the corresponding weights of each side output, where  $M$  is the number of side outputs. The weight of a short connection from side output  $i$  to side output  $m$  ( $i > m$ ) is represented by  $r_i^m$ . Let  $r = \{r_i^m\}, i > m$ . The side loss function denoted by  $L_{side}$  is defined as

$$L_{side}(W, w, r) = \sum_{m=1}^M \alpha_m l_{side}^{(m)}(W, w^{(m)}, r),$$

where  $\alpha_m$  is the weight of the  $m$ th side loss, and  $l_{side}^{(m)}(W, w^{(m)}, r)$  refers to the cross-entropy loss function such that

$$l_{side}^{(m)}(W, w^{(m)}, r) = - \sum_{z_j \in \mathcal{Z}} z_j \log \Pr(z_j = 1 | X; W, w^{(m)}, r) + (1 - z_j) \log \Pr(z_j = 0 | X; W, w^{(m)}, r),$$

where  $\Pr(z_j = 1 | X; W, w^{(m)}, r)$  be the probability of the activation value in the  $m$ th side output.

#### 2.3.4 Fusion layer

To optimize the modified DSS model, the fusion loss is applied in each side output, as illustrated with the dot line in Figure 16. The fusion loss function expressed by

$$L_{fuse}(W, w, f, r) = \sigma \left( Z, \sum_{m=1}^M f_m R_{side}^m \right),$$

where  $\sigma(\cdot, \cdot)$  denote the distance between the ground truth map and the fused predictions,  $f$  is the fusion weights, and  $R_{side}^m$  represent the side activations at the  $m$ th side output as



$$R_{side}^m = \begin{cases} \sum_{i=m+1}^M r_i^m R_{side}^{(i)} + A_{side}^{(m)}, & \text{for } m = 1, \dots, 5, \\ A_{side}^{(m)}, & \text{for } m = 6 \end{cases},$$

where  $A_{side}^{(m)} = \{a_j^{(m)}, j = 1, \dots, |X|\}$  be the activations of the  $m$ th side output. In addition, some short connections can be dropped out by setting  $r_i^m = 0$ .

Hence, the final loss function can be calculated by

$$L_{final}(W, w, f, r) = L_{side}(W, w, r) + L_{fuse}(W, w, f, r).$$

## 2.4 Keypoint selection

Normally, shape feature descriptors are extracted from all shape boundary points. However, the noises in the contour of the segmented mask may harm the feature descriptor performance. The keypoint selection [7] was proposed to overcome this problem.

Let  $\Omega_B$  be the set of the boundary points defined by

$$\Omega_B = \{p_1, p_2, \dots, p_i, \dots, p_{N-1}, p_N\},$$

where  $p_i = (x_i, y_i)$  is the  $i$ th boundary point and  $N$  is the total number of the boundary points. Figure 18(a) displays an arbitrary shape with  $N$  boundary points.

The included angle at  $p_i$ , as appearing in Figure 18(b), can be represented by  $\theta_k(i)$  as follows:

$$\theta_k(i) = \cos^{-1} \left( \frac{\overline{p_i p_{i-k}}^2 + \overline{p_i p_{i+k}}^2 - \overline{p_{i-k} p_{i+k}}^2}{2 \times \overline{p_i p_{i-k}} \times \overline{p_i p_{i+k}}} \right),$$

where  $\overline{p_i p_j} = \left( (x_i - x_j)^2 + (y_i - y_j)^2 \right)^{1/2}$ .

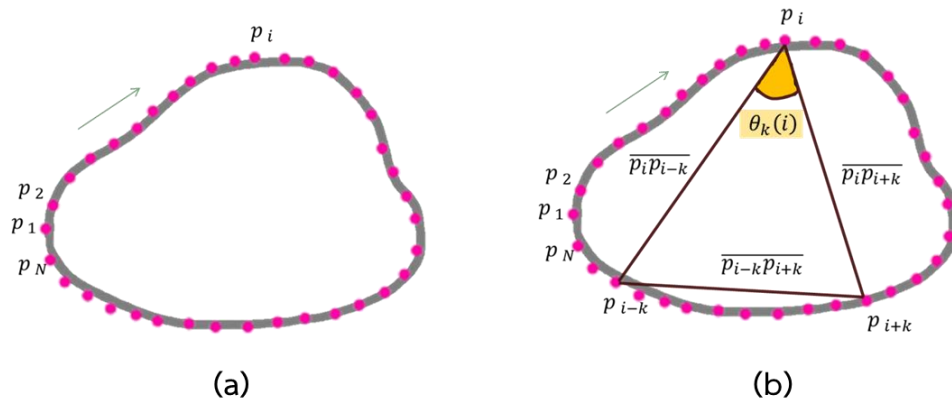


Figure 18 (a) Arbitrary shape with  $N$  boundary points and (b) included angle at point  $p_i$

To overcome the noise sensitivity, the included angles are treated as a random variable. The first to the third moments  $\Gamma_1(i)$ ,  $\Gamma_2(i)$ , and  $\Gamma_3(i)$  of  $\theta_k(i)$  can be approximated by

$$\Gamma_1(i) = E[\theta_k(i)] = \frac{1}{m} \sum_{k=1}^m \theta_k(i),$$

$$\Gamma_2(i) = E[\{\theta_k(i) - \Gamma_1(i)\}^2] = \frac{1}{m} \sum_{k=1}^m \{\theta_k(i) - \Gamma_1(i)\}^2,$$

$$\Gamma_3(i) = E[\{\theta_k(i) - \Gamma_1(i)\}^3] = \frac{1}{m} \sum_{k=1}^m \{\theta_k(i) - \Gamma_1(i)\}^3,$$

where  $m = \lfloor (N - 1)/2 \rfloor$ .

After that, the first to the third moments are used to find the keypoints of the shape boundary by selecting points at local extrema of each moment graph.

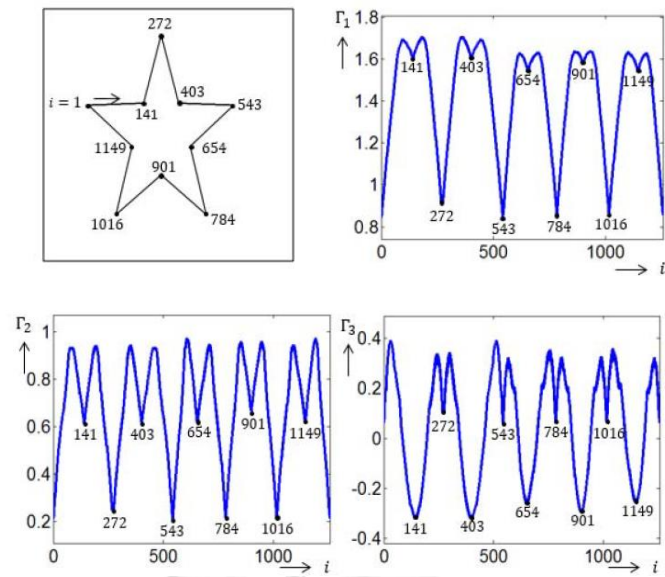


Figure 19 The illustration of complex shape and its moment graphs

Source: Adapted from [7]

Since each moment graph provides different keypoints, the entire keypoint set of a given shape ( $S_{total}$ ) can be defined as

$$S_{total} = S_1 \cup S_2 \cup S_3,$$

when the sets of the keypoints from the first to the third moment graphs is  $S_1$ ,  $S_2$ , and  $S_3$ , respectively.

To remove the unwanted local extreme points occurred from noisy boundaries, the persistence value of each local extreme point  $s$ , denoted as ( $p_s$ ), is calculated by the persistence 1D method as

$$p_s = \min(d_s^{m_{s1}}, d_s^{m_{s2}}),$$

where  $m_{s1}$  and  $m_{s1}$  are extreme points connected with  $s$ ,  $d_s^{m_{s1}} = |\Gamma_j(s) - \Gamma_j(m_{s1})|$ , and  $d_s^{m_{s2}} = |\Gamma_j(s) - \Gamma_j(m_{s2})|$  for  $j = 1, 2, 3$ .

If a persistence value of local extreme point smaller than a given threshold, this local extreme point is rejected from the local extreme point set.

## 2.5 Boundary feature descriptor

To get some properties of an object in the image, a feature extraction method is always used to extract these properties, and a feature descriptor is then used to represent it. In recent years, the boundary feature descriptor, global feature extraction method, has been used in many researches. Because it is easy to implement with low computational complexity and high recognition rates. The approach of this descriptor is to describe only region boundaries and ignore the information inside the shape.

### 2.5.1 Centroid contour distance

The centroid contour distance (CCD) feature [8] is a spatial domain feature vector expressed by the distance between the boundary points and the centroid point of a shape, as shown in Figure 20.

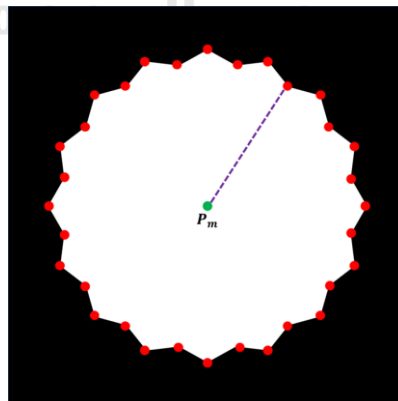


Figure 20 The illustration of distance from contour point to centroid point  $P_m$

Let  $\{p_1, p_2, \dots, p_N\}$  be a set of contour points where the number of contour points is  $N$ . The centroid point of these contour points denoted by  $P_m$  is defined as

$$P_m = \frac{1}{N} \sum_{i=1}^N p_i = \left( \frac{1}{N} \sum_{i=1}^N x_i, \frac{1}{N} \sum_{i=1}^N y_i \right),$$

where contour point  $p_i$  of a shape has coordinates  $(x_i, y_i)$ .

The Euclidean distance between the  $i$ th contour point and the centroid point  $p_m = (x_m, y_m)$ , denoted as  $d^i$ , can be calculated by

$$d^i = ((x_i - x_m)^2 + (y_i - y_m)^2)^{1/2}, \quad i = 1, 2, \dots, N.$$

The CCD feature, denoted as  $d_{ccd}^i$ , associates with the normalization of  $d^i$  is defined as follows:

$$d_{ccd}^i = (d^i / (\frac{1}{N} \sum_{j=1}^N d^j)) / \sqrt{N} = d^i \sqrt{N} / \sum_{j=1}^N d^j, \quad i = 1, 2, \dots, N.$$

Therefore, the sequence of the CCD features can be written as

$$D_{ccd} = \{d_{ccd}^1, d_{ccd}^2, \dots, d_{ccd}^N\}.$$

จุฬาลงกรณ์มหาวิทยาลัย  
CHULALONGKORN UNIVERSITY

### 2.5.2 Fourier descriptor based on centroid contour distance

Fourier descriptor based on CCD (FD-CCD) [9] is obtained after applying the Fourier transform on CCD. Thereby, CCD is transformed into a feature vector in the frequency domain. The advantage of using FD-CCD is a high speed, easy to compute, and insensitive to the rotation.

Let  $F_{ccd}$  represent the FD-CCD feature based on  $D_{ccd}$ . It can be calculated by

$$F_{ccd}(k) = \frac{1}{N} \left| \sum_{i=0}^{N-1} d_{ccd}^{i+1} e^{\frac{-j2\pi ik}{N}} \right|, \quad k = 0, 1, \dots, N-1.$$

## 2.6 Random Forest

### 2.6.1 Decision Tree

A decision tree, a supervised learning algorithm, is often used for binary and multi-class classification tasks. This algorithm aims to produce a model that predicts target values by learning simple decision rules inferred from data attributes. This algorithm's structure is a series of nodes that are divided into three types: the root node, the internal node, and the leaf node, as shown in Figure 21.

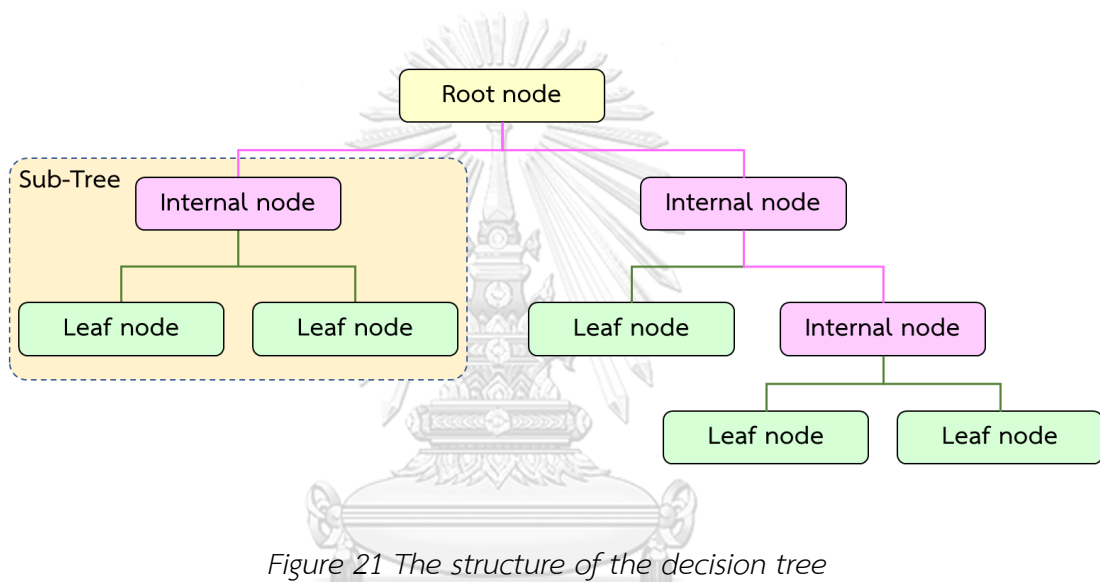


Figure 21 The structure of the decision tree

The decision tree is a top-down tree structure that starts the splitting process from the root node. The root node and the internal node are divided according to selected attributes into two or more sub-groups, and the leaf node which is a node that does not split or contains no branch will be given an appropriate class for the input sample.

For building the decision tree, the algorithm called ID3 [10] uses Entropy and Information Gain (IG) to select the attributes that give the best partition. The process of doing a partition will continue repeatedly splitting until a stopping condition is reached.

### 2.6.2 Entropy

Entropy is the value that indicates the impurity of the information in a dataset, and it is used to control the data splitting process in the decision tree algorithm. The entropy of dataset  $X$  denoted by  $E(X)$  can be calculated by the following equation:

$$E(X) = - \sum_{i=1}^c P_i \log_b P_i,$$

where  $P_i$  is the probability of finding an element or class  $i$ , and  $c$  is the total number of classes in the dataset.

### 2.6.3 Information Gain

The information gain (IG) is based on the change in entropy after a splitting process on an attribute. The goal of constructing a decision tree is about finding an attribute that gives the highest information gain. The information gain can be computed by the following equation:

$$IG(A) = E(X) - E_A(X),$$

where  $E_A(X)$  is the entropy of a split using attribute  $A$  defined by

$$E_A(X) = - \sum_{i=1}^c \frac{|X_i|}{|X|} \log_2 \frac{|X_i|}{|X|}.$$

### 2.6.4 Gini Index

Gini Index is used to measure the probability of a particular variable being incorrectly identified when it is randomly chosen. The gini index can be calculated by the following equation:

$$G(X) = 1 - \sum_{i=1}^c (P_i)^2,$$

where  $P_i$  is the probability of an object being classified to a particular class  $i$ .

Unlike the information gain, the computational cost of the gini index is not as high as information gain, which involves the entropy function calculation. Therefore, the gini index is often preferred over the information gain.

### 2.6.5 Random Forest

Random Forest (RF), one of the most attractive and powerful machine learning algorithms, is a model that consists of numerous decision trees called subtrees. Building a random forest with  $N$  subtrees, the random forest algorithm randomly produces multiple separate datasets into  $N$  subsets, and each subset will be used for creating one subtree. Figure 22 illustrates the structure of the random forest. For classifying, each subtree will predict the input data class, and then the result is voted by choosing the class with the most voting.

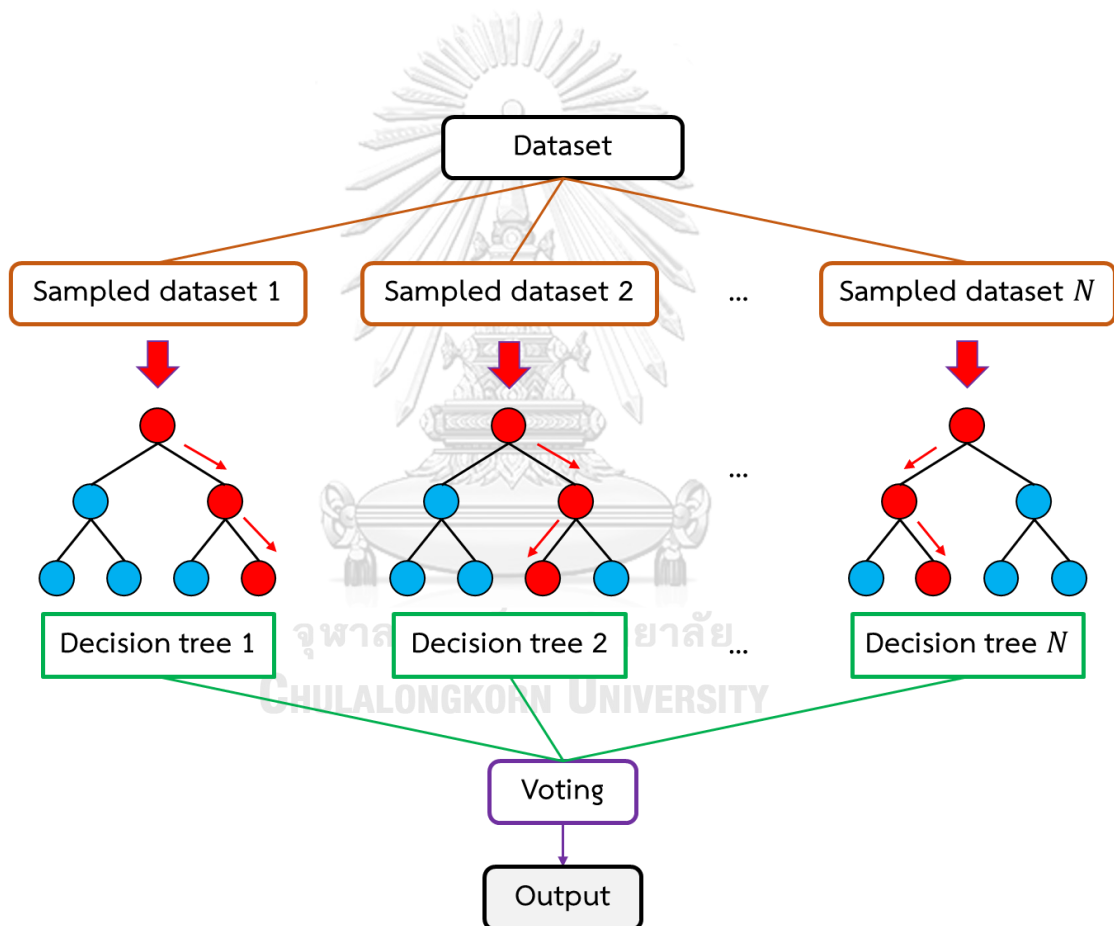


Figure 22 The illustration of the random forest structure



## CHAPTER III

### Proposed Method

In this work, the classification of aircraft with shadow method is proposed. This method consists of 3 steps: segmentation, feature extraction, and classification. Figure 23 illustrates the framework of our proposed method.

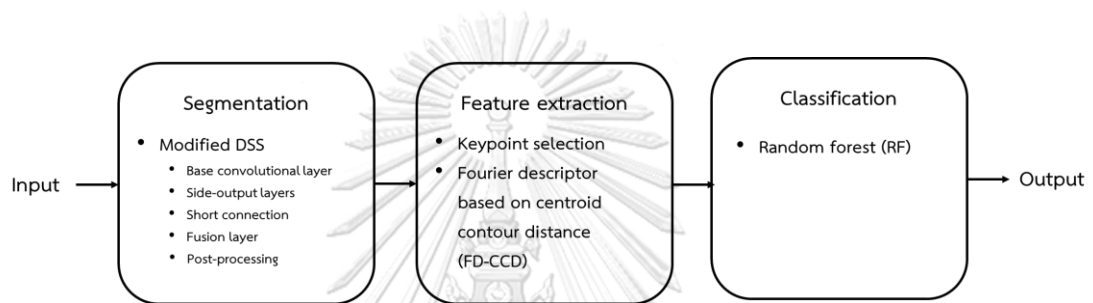


Figure 23 The framework of our proposed method

In the segmentation step, we employ the DSS network to outstand the most prominent part of the image, which is the aircraft. Then replace it as a white pixel and replace the part of the background and the other objects as a black pixel. Since the shadow problem mentioned before may affect segmentation results, the modified DSS is performed to get a better segmentation image by adding the L2-Normalization layer. Moreover, we propose the post-processing to remove noise that occurred from the other objects on the background.

During the feature extraction step, we aim to extract appropriate features for the next classification step. Before extraction, since there is noise on segmented image boundaries, we choose some points as a sampling point by doing the keypoint selection method. After that, FD-CCD is used for extracting features that invariance to scale and rotation.

Finally, the classification step, the obtained features from the previous step are used to classify the type of aircraft using the RF classifier.

### 3.1 Segmentation

#### 3.1.1 Modified Deeply Supervised Salient Object Detection with Short Connections

The modified DSS network consists of five parts. The first four parts are similar to the DSS network: Base convolutional layer, Side-output layers, Short connection, and Fusion layer. In addition, we propose the post-processing part after finishing this network. Figure 24 illustrates the architecture of the modified DSS network.

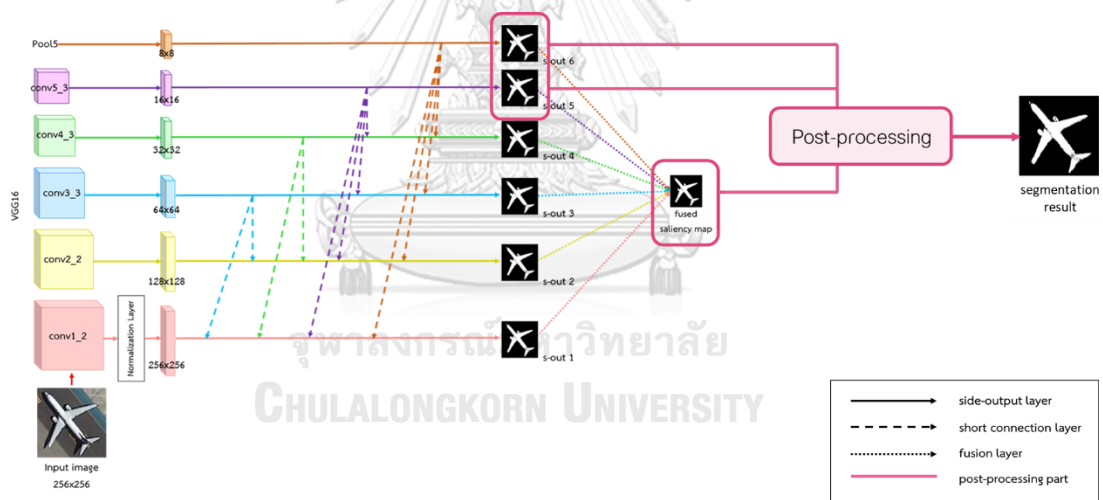


Figure 24 The framework of the modified DSS architecture

The base convolutional layer is the same as in DSS, we also use the VGG16 architecture as our base convolutional layer. However, in side-output layers, we add the L2-Normalization layer before connecting the shallowest side-output layer with the last obtained convolutional layer of the first stage in VGG16 (conv1\_2).

The L2-Normalization layer is useful for the feature combinations. The training process is learning more stable by using L2-Normalization to rescale the weight. Consequently, using the normalization layer can improve the first side-output saliency map (s-out 1) and fused saliency map.

Let  $\mathbf{x} = (x_1 \cdots x_d)$  is  $d$ -dimensional input layer, The L2-Normalization of  $\mathbf{x}$  defined as

$$\|\mathbf{x}\|_2 = \left( \sum_{i=1}^d |x_i|^2 \right)^{1/2}.$$

For the short connection part, we combine the deeper side output layer with the shallower side output layer using the short connection pattern defined as

$$R_{side}^m = \begin{cases} \sum_{i=3}^6 r_i^m R_{side}^{(i)} + A_{side}^{(m)}, & \text{for } m = 1, 2 \\ r_5^m R_{side}^{(5)} + r_6^m R_{side}^{(6)} + A_{side}^{(m)}, & \text{for } m = 3, 4 \\ A_{side}^{(m)}, & \text{for } m = 5, 6 \end{cases},$$

where  $R_{side}^m$  is the side activations at the  $m$ th side output, and  $A_{side}^{(m)}$  is the activations of the  $m$ th side output.

In the fusion layer part, we fuse each side-output saliency map into the fused saliency map by the formulas in section 2.3.4.

After doing the fusion layer steps, the s-out 5, s-out 6, and fused saliency map are used in the post-processing algorithm. In Figure 25, we will see that the saliency maps from two deeper side outputs, s-out 5 and s-out 6, create high-level features which have the ability to locate the most salient region but lack detail. So, we use it to create a mask for reducing the noise around the aircraft in the result. The structure of the post-processing is shown in Figure 26. Firstly, we apply Bitwise AND operator to the two side-output saliency maps, s-out 5, and s-out 6. Next,

we do the binary thresholding (threshold = 0.15) and the dilation (with a square structuring element of size  $5 \times 5$  whose components are all 1s) to create the binary mask and make the object inside it larger. In the last step, we combine the obtained dilated mask with the fused saliency map from the modified DSS.

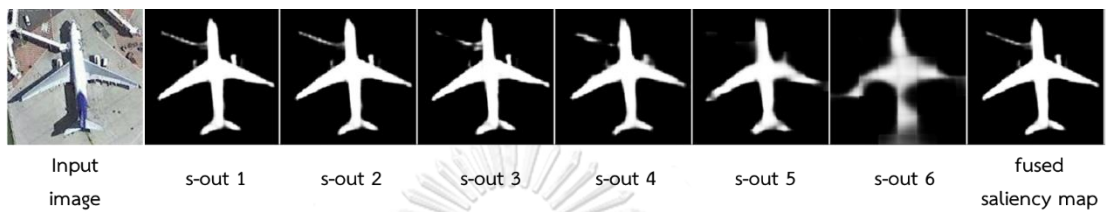


Figure 25 Example of each side-output result from the DSS network

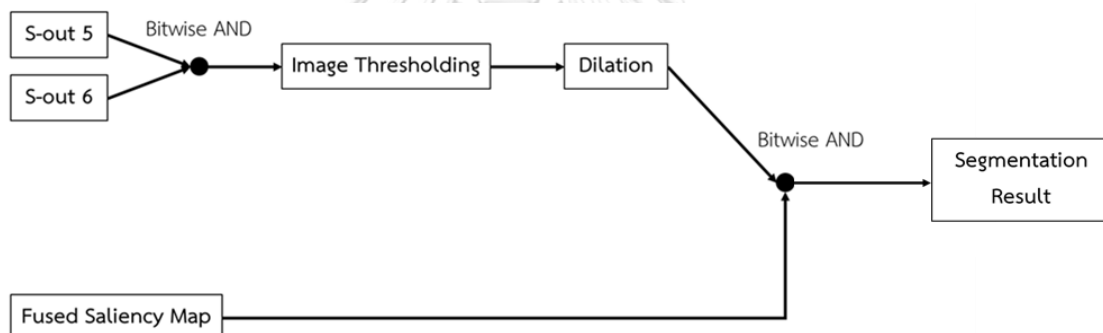


Figure 26 The post-processing framework

### 3.2 Feature extraction

In this work, we use the keypoint selection, explained in section 2.4, to choose the keypoints that invariant to the noise from the boundaries of segmentation result. These keypoints are treated as the initial sampling points. Afterward, FD-CCD (see in section 2.5.2) is used to extract descriptors from each sampling points.

### 3.3 Classification

In the classification, the RF classifier with 40 decision trees (see in section 2.6.5) is employed to learn the FD-CCD features obtained from the previous step and classify the types of aircraft. The classification criteria used in this work is the gini index (see in section 2.6.4).



## CHAPTER IV

### Results and Discussion

In this research, we divide our experiment into two parts. The first one is the segmentation test, which aims to compare aircraft segmentation capability in various backgrounds between our modified DSS and the traditional DSS model. The second one is the classification test. The proposed method is tested to classify six different types of aircraft. In addition, our proposed method is built using preexisting python code and self-programmed python code with PyTorch and OpenCV libraries.

#### 4.1 Segmentation test

In this part, we collect 1,000 aircraft images from Google Earth as our dataset. We first resize the images into 256×256 pixels. After that, divide them into the training set, validation set, and test set consisting of 542, 200, and 258 images, respectively.

For training, we use data augmentation techniques: random rotation, brightness, and contrast to increase the training samples. Then the pre-trained weights of DSS are used to improve the segmentation model's performance and avoid model overfitting.

For testing, we separate our test set into two cases, simple and difficult. Figure 27 shows example images from the simple case and the difficult case. The simple case consists of 100 images of aircraft with an uncomplicated background. There are 158 aircraft images with shadow and other objects such as docks, ramps, cargo containers, and cars in the background of the difficult case. The example visual results at each side-output layer are illustrated in Figure 28 and Figure 29, and the example segmentation results of our method and the original DSS method are compared in Figure 30.



Figure 27 Example images from (a) the simple case and (b) the difficult case

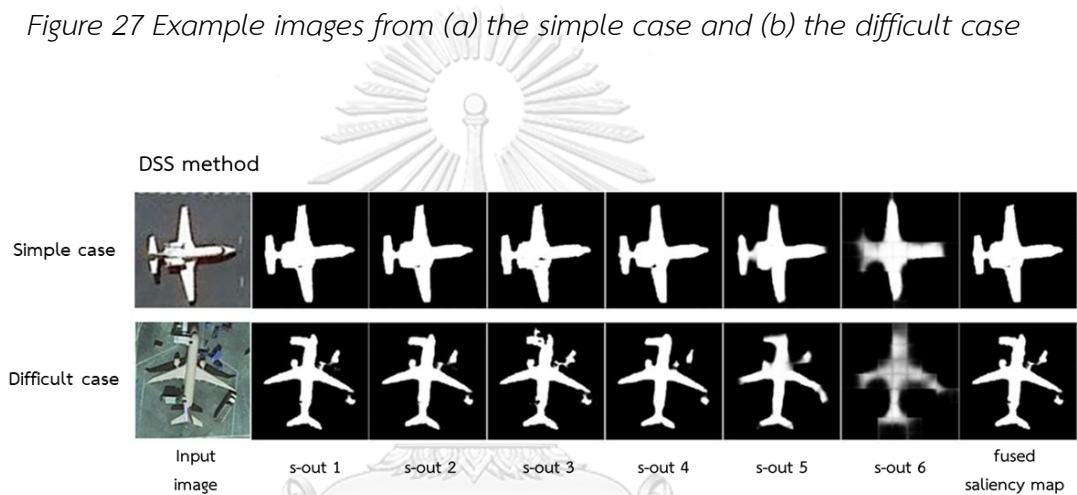


Figure 28 The visualization of each side-output result from the DSS network



Figure 29 The visualization of each side-output result from the modified DSS network



Figure 30 The comparison results between DSS method and modified DSS method

Experimental results show that our proposed segmentation method can segment the aircraft image with shadow and better reduce noise from the background compared with the original DSS method. However, this method still has a critical problem when the input image has several colors on the body of aircraft and has high or low contrast, as we can see in Figure 31.



Figure 31 Example segmentation images of failure cases



In order to evaluate the model, we use the mean absolute error (*MAE*) that can be computed from

$$MAE = \frac{1}{H \times W} \sum_{i=1}^H \sum_{j=1}^W |\hat{S}(i, j) - \hat{Z}(i, j)|,$$

where  $H$  and  $W$  are height and width of an input image,  $\hat{S}$  be the continuous saliency map, and  $\hat{Z}$  denoted the ground truth that are normalized to  $[0,1]$ .

Moreover, we use  $F$ -measure which can be calculated by

$$F_{\beta} = \frac{(1 + \beta^2) Precision \times Recall}{\beta^2 Precision + Recall},$$

where  $\beta^2$  is set to be 0.3 to weight the importance of the precision value, and

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive},$$

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative}.$$

The definition of *True Positive*, *False Positive*, and *False Negative* is shown in Table 3.

Table 3 The confusion matrix for a binary classification problem

		Actual Target	
		Positive	Negative
Predicted Target	Positive	<i>True Positive</i>	<i>False Negative</i>
	Negative	<i>False Positive</i>	<i>True Negative</i>

Our modified DSS method is compared with the original DSS. The results are shown in Table 4.

Table 4 The performance comparison between DSS and modified DSS method

Methods	Full dataset		Simple case		Difficult case	
	<i>MAE</i>	<i>F<sub>β</sub></i>	<i>MAE</i>	<i>F<sub>β</sub></i>	<i>MAE</i>	<i>F<sub>β</sub></i>
DSS	0.0426	0.8936	0.0352	0.9172	0.0452	0.8853
<b>Modified DSS</b>	0.0389	<b>0.9110</b>	0.0349	<b>0.9234</b>	0.0402	<b>0.9067</b>

The results of our method are close to the original's results in the simple case. However, in the difficult case, ours can achieve 90.67% accuracy. Thus, our method can obtain more performance compared with the original method.

#### 4.2 Classification test

In the classification test, 1,140 remote sensing images of aircraft from The Multi-type Aircraft of Remote Sensing Images (MTARSI) by Wu, Zhize [11] are used as our dataset. The dataset consists of 6 different aircraft types: Boeing passenger aircraft, B-1 bombers, B-2 bomber, B-52 bomber, E-3 early warning aircraft, and F-22 fighter aircraft. All of the images are resized into 256x256 pixels and separated into a training set (660 images), validation set (240 images), and test set (240 images). The example images of each aircraft type are displayed in Figure 32.

For training, we select the keypoints from the ground truth label of our training set and extract the FD-CCD feature at each keypoint. Then the feature is used to train our RF model.

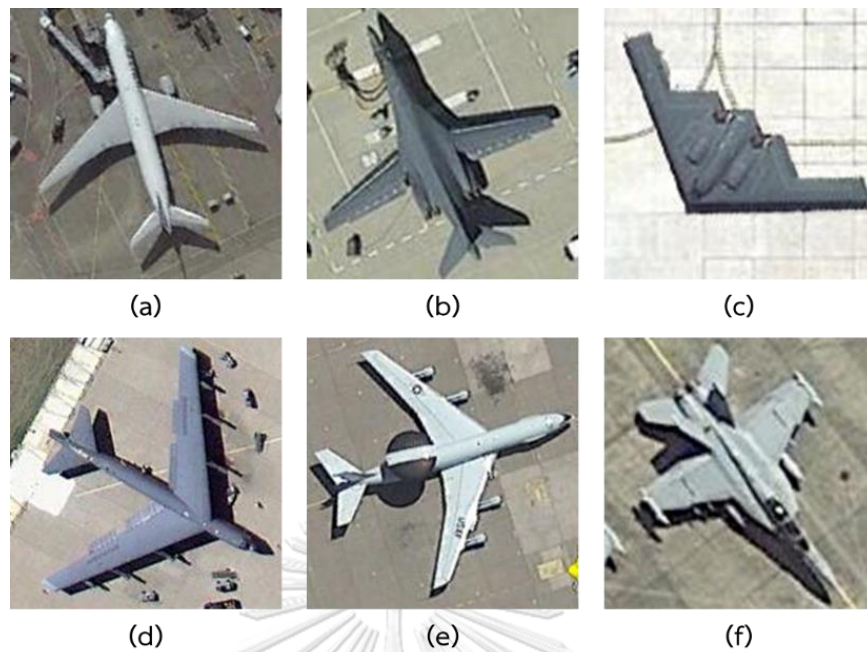


Figure 32 Example images of 6 aircraft types used in this work. (a) Boeing, (b) B-1, (c) B-2, (d) B-52, (e) E-3, and F-22

For testing, we select the keypoints from the segmented images obtained from the DSS and modified DSS algorithm. The FD-CCD descriptor is then extracted at each keypoint and used in the RF model. Tables 5 and 6 show the confusion matrix of the aircraft classification results using the DSS network and modified DSS network, respectively. These confusion matrices summarize the number of correct and incorrect decisions.

Table 5 The confusion matrix of classification result using the DSS network with FD-CCD and RF classifier





























Actual	Predicted						Total
	Boeing	B-1	B-2	B-52	E-3	F-22	
Boeing	24	7	0	4	2	3	40
B-1	1	38	0	0	1	0	40
B-2	0	0	40	0	0	0	40
B-52	0	0	0	38	0	2	40
E-3	0	0	0	0	40	0	40
F-22	0	0	0	0	0	40	40

Table 6 The confusion matrix of classification result using the modified DSS network with FD-CCD and RF classifier

Actual	Predict						Total
	Boeing	B-1	B-2	B-52	E-3	F-22	
Boeing	28	2	0	3	3	4	40
B-1	0	40	0	0	0	0	40
B-2	0	0	40	0	0	0	40
B-52	1	0	0	37	1	1	40
E-3	0	0	0	0	40	0	40
F-22	0	0	0	0	0	40	40

From the results that are shown in Table 5 and Table 6, we found that our method can perform well on this dataset. This method can accurately predict four classes: B-1, B-2, E-3, and F-22. Moreover, it can correctly predict more than the original DSS method in Boeing and B-52 class. However, some images from class Boeing and B-52 are still mispredicted, as shown in Table 7.

Table 7 Example of incorrect prediction results

Actual Class	Input image	Ground truth	Segmented image		Predicted class	
			DSS	ours	DSS	ours
Boeing					B-1	B-1
Boeing					B-52	B-52
Boeing					E-3	E-3
Boeing					F-22	F-22
B-52					F-22	Boeing
B-52					B-52	E-3
B-52					F-22	F-22

CHULALONGKORN UNIVERSITY

As shown in Table 7, Boeing is confused with B-1, B-52, E-3, and F-22 because the segmented image obtained from the segmentation process is not clear and imperfect. Besides, B-52 is also confused with Boeing, E-3, and F-22. The sixth row of this table shows that our method mispredicts B-52 with E-3 while DSS is correct. It seems like our method decides that the white pixel at the fuselage area is similar to the rotodome in E-3.

To evaluate the model, the accuracy can be approximated by

$$Accuracy = \frac{\#True\ prediction}{\#All\ predictions}$$

The comparison of accuracy between our method and the DSS method are represented in Table 8.

*Table 8 The Accuracy of the DSS and modified DSS method with FD-CCD and RF classifier*

Methods	Accuracy
DSS+FD-CCD+RF	0.9042
Modified DSS+FD-CCD+RF	0.9375

The results show that our method better classifies the different types of aircraft with shadow compared with the original method. Its accuracy achieves 93.75% on this dataset.

## CHAPTER V

### Conclusions

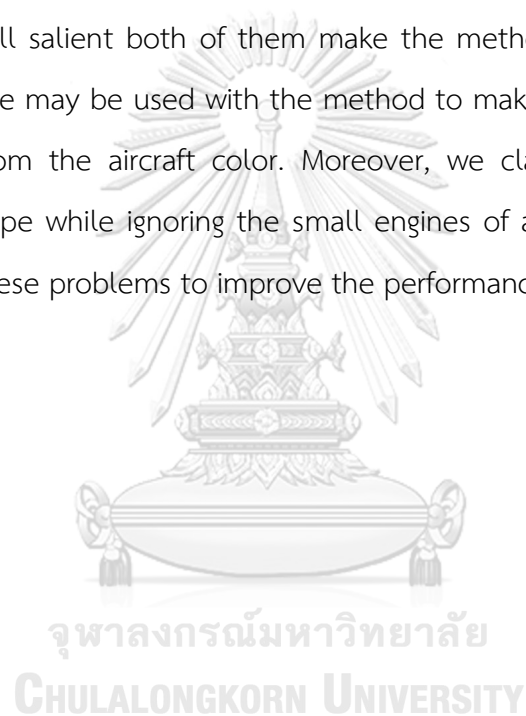
This work introduces a method that intends to categorize the type of aircraft with shadow in remote sensing images. There are three steps in this method, segmentation, extraction, and classification. The segmentation part uses the modified DSS to segment only the aircraft body out of the background. This modified DSS algorithm adjusts the primitive DSS algorithm by adding L2-Normalization in the middle of the shallowest side-connection between convolutional and side-output layers to increase the performance of the result saliency map. So, the aircraft body can stand out from the shadow and other objects. Besides, the post-processing is presented to handle the noise problem. In the extraction part, we first use the keypoints selection method to choose the representative boundary point that invariant to noise. The FD-CCD methods are then adopted to extract the scale and rotation invariance features at each keypoint. The final part, classification, the RF is used to classify types of aircraft images via the FD-CCD features.

Our experiment is divided into two parts: the segmentation test and classification test. In the first part, we apply the original DSS and the modified DSS algorithm with two cases of aircraft images, simple and difficult. The background of images from the difficult case is more complicated than the simple one. It has aircraft and other objects, including the shadow in the same frame. The result shows that the modified DSS works well with both cases, while the result of the original DSS gains more noise from the complex background. However, some problems occur when the aircraft has more than one color on their body and has dramatically high or low contrast. The second experiment aims to test the method in classification performance. The proposed method is used to classify six types of aircraft by using

the RF classifier. The experimental result shows that our method with the FD-CCD and RF classifier can classify different types of aircraft with shadow effectively.

### 5.1 Future work

In this work, we proposed the modified DSS, which can address the shadow and noise problem. However, if the contrast between shadow and aircraft is considerably low or high, the method cannot distinguish between the aircraft body and shadow. It will salient both of them make the method performance decrease. So, the color space may be used with the method to make the color of the shadow more different from the aircraft color. Moreover, we classify by focusing on the aircraft body's shape while ignoring the small engines of aircraft. In future work, we intend to solve these problems to improve the performance of aircraft segmentation.





## REFERENCES

1. Gurselkaracor, A. and Erdaltorun, *AIRCRAFT CLASSIFICATION USING IMAGE PROCESSING TECHNIQUES AND ARTIFICIAL NEURAL NETWORKS*. International Journal of Pattern Recognition and Artificial Intelligence, 2012. **25**.
2. Zhu, X., et al., *Aircraft type classification based on an optimized Bag of Words Model*. 2016. 434-437.
3. Xie, S. and Z. Tu *Holistically-Nested Edge Detection*. 2015. arXiv:1504.06375.
4. Hou, Q., et al. *Deeply supervised salient object detection with short connections*. 2016. arXiv:1611.04849.
5. Dai, Z., et al., *CNN Descriptor Improvement Based on L2-Normalization and Feature Pooling for Patch Classification*. 2018. 144-149.
6. Simonyan, K. and A. Zisserman *Very Deep Convolutional Networks for Large-Scale Image Recognition*. 2014. arXiv:1409.1556.
7. Yang, D. and H. Park, *A New Shape Feature for Vehicle Classification in Thermal Video Sequences*. IEEE Transactions on Circuits and Systems for Video Technology, 2016. **26**(7): p. 1363-1375.
8. Chang, C.C., S.M. Hwang, and D.J. Buehrer, *A shape recognition scheme based on relative distances of feature points from the centroid*. Pattern Recognition, 1991. **24**(11): p. 1053-1063.
9. Zhang, D. and G. Lu, *Study and evaluation of different Fourier methods for image retrieval*. Image and Vision Computing, 2005. **23**(1): p. 33-49.
10. Quinlan, J.R., *Induction of decision trees*. Machine Learning, 1986. **1**(1): p. 81-106.
11. Wu, Z.-Z., et al., *A benchmark data set for aircraft type recognition from remote sensing images*. Applied Soft Computing, 2020. **89**: p. 106132.



จุฬาลงกรณ์มหาวิทยาลัย  
**CHULALONGKORN UNIVERSITY**

## VITA

NAME Natnicha Meeboonmak

DATE OF BIRTH 25 September 2537

PLACE OF BIRTH Bangkok

INSTITUTIONS ATTENDED Kasetsart University

HOME ADDRESS 303/060101 Village No.7 Khao Phra Ngam Sub-District,  
Mueang Lop Buri District, Lop Buri Province

