Semi-Supervised Deep Learning with MaligNet for Bone Lesion Instance Segmentation Using Bone Scintigraphy

Mr. Terapap Apiparakoon

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Engineering in Computer Engineering
Department of Computer Engineering
FACULTY OF ENGINEERING
Chulalongkorn University
Academic Year 2019

การเรียนรู้เชิงลึกแบบกึ่งมีผู้สอนด้วยมาลิกเน็ตสำหรับการแบ่งส่วนตัวอย่างรอยโรคที่กระดูกโดยใช้ภาพถ่ายสแกนกระดูกด้วยวิธีทางเวชศาสตร์นิวเคลียร์

นายธีรภาพ อภิปาลกุล

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2562

Thesis Title	Semi-Supervised Deep Learning with MaligNet for Bone Lesion Instance Segmentation Using Bone Scintigraphy
By	Mr. Terapap Apiparakoon
Field of Study	Computer Engineering
Thesis Advisor	Dr. Ekapol Chuangsuwanich, Ph.D.
Thesis Co Advisor	Assistant Professor Dr. YOTHIN RAKVONGTHAI, Ph.D.

Accepted by the FACULTY OF ENGINEERING, Chulalongkorn University in Partial Fulfillment of the Requirement for the Master of Engineering

..... Dean of the FACULTY OF ENGINEERING
(Associate Professor SUPOT TEACHAVORASINSKUN, Ph.D.)

THESIS COMMITTEE

..... Chairman
(Dr. Duangdao Wichadakul, Ph.D.)

..... Thesis Advisor
(Dr. Ekapol Chuangsuwanich, Ph.D.)

..... Thesis Co-Advisor
(Assistant Professor Dr. YOTHIN RAKVONGTHAI, Ph.D.)

..... Examiner
(Associate Professor Dr. TAWATCHAI CHAIWATANARAT, M.D.)

..... External Examiner
(Dr. Supasorn Suwajanakorn, Ph.D.)

ธีรภาพ อภิปาลกุล : การเรียนรู้เชิงลึกแบบกึ่งมีผู้สอนด้วยมาลิกเน็ตสำหรับการแบ่งส่วนตัวอย่างรอยโรคที่กระดูกโดยใช้ภาพถ่ายสแกนกระดูกด้วยวิธีทางเวชศาสตร์นิวเคลียร์. (Semi-Supervised Deep Learning with MaligNet for Bone Lesion Instance Segmentation Using Bone Scintigraphy) อ.ที่ปรึกษาหลัก : อ. ดร.เอกพล ช่วงสุวนิช, อ.ที่ปรึกษาร่วม : ผศ. ดร.โยธิน รักวงษ์ไทย

หนึ่งในความท้าทายของการประยุกต์ใช้การเรียนรู้เชิงลึกกับภาพถ่ายทางการแพทย์คือการขาดข้อมูลที่มีผลเฉลยที่ชัดเจน แม้ว่าจะมีข้อมูลทางการแพทย์จำนวนมากก็ตามแต่ก็เป็นเรื่องที่ยากที่จะระบุผลเฉลยของข้อมูลนั้นโดยเฉพาะถ่ายสแกนกระดูกด้วยวิธีทางเวชศาสตร์นิวเคลียร์ซึ่งเป็นภาพถ่ายสแกนกระดูกแบบ 2 มิติ โดยปกติภาพถ่ายสแกนกระดูกด้วยวิธีทางเวชศาสตร์นิวเคลียร์จะมีสิ่งรบกวนและการบดบัง ผลเฉลยของข้อมูลและผลเฉลยที่ใช้อ้างอิง (gold standard) มักจะได้มาจากการผ่าตัดหรือการวินิจฉัยทางการแพทย์ซึ่งไม่สามารถหาได้จากการวิเคราะห์เบื้องต้น ผมจึงเสนอโมเดลโครงข่ายประสาทเทียมแบบใหม่ที่สามารถแบ่งส่วนบริเวณจุดสว่างที่ผิดปกติและแบ่งประเภทรอยโรคที่บริเวณหน้าอกด้วยวิธีการเรียนรู้แบบกึ่งมีผู้สอน โมเดลของผมเรียกว่า "มาลิกเน็ต" เป็นโมเดลสำหรับแบ่งส่วนตัวอย่างที่ผสมผสานโมเดลขั้นบันไดจัดการกับข้อมูลที่มีผลเฉลยและไม่มีผลเฉลย แตกต่างกับโมเดลการเรียนรู้เชิงลึกสำหรับแบ่งส่วนประเภทอื่นๆที่จะแบ่งประเภทแต่ละตัวอย่างอย่างอิสระ มาลิกเน็ตใช้ประโยชน์จากข้อมูลองค์รวมผ่านทางการเชื่อมต่อเพิ่มเติมจากแกนนำของโครงข่ายประสาทเทียม ในการประเมินประสิทธิภาพของโมเดล ผมสร้างชุดข้อมูลสำหรับแบ่งส่วนรอยโรคบนกระดูกโดยใช้ข้อมูลที่มีผลเฉลยและไม่มีผลเฉลยของคนไข้จำนวน 544 และ 9,280 คนตามลำดับ โมเดลที่ผมนำเสนอมีค่าเฉลี่ยพรีซิชัน (precision), ค่าเฉลี่ยรีคอล (recall), และค่าเฉลี่ยเอฟวัน (f1-score) เฉลี่ยอยู่ที่ 0.852, 0.856, และ 0.848 ตามลำดับ มีประสิทธิภาพเหนือกว่าโมเดลพื้นฐาน (Mask R-CNN) 3.92% โดยสัมพัทธ์ ผลการวิเคราะห์แสดงให้เห็นว่าการนำข้อมูลโดยรวมมาใช้นั้นช่วยให้โมเดลสามารถแบ่งประเภทตัวอย่างรอยโรคที่มีความเฉพาะเจาะจงที่ซึ่งต้องการข้อมูลจากบริเวณอื่นๆ สำหรับโจทย์การแบ่งประเภทการแพร่กระจายของมะเร็งกระดูก โมเดลบรรลุค่ารีคอล (recall) ที่ 0.657 และ ค่าเฉพาะเจาะจง (specificity) ที่ 0.857 แสดงให้เห็นถึงศักยภาพที่ยอดเยี่ยมสำหรับการตรวจวินิจฉัยโดยใช้ภาพถ่ายสแกนกระดูกด้วยวิธีทางเวชศาสตร์นิวเคลียร์ในทางปฏิบัติทางคลินิก

สาขาวิชา	วิศวกรรมคอมพิวเตอร์	ลายมือชื่อนิสิต .....
ปีการศึกษา	2562	ลายมือชื่อ อ.ที่ปรึกษาหลัก .....
		ลายมือชื่อ อ.ที่ปรึกษาร่วม .....

6170191421 : MAJOR COMPUTER ENGINEERING

KEYWORD: Bone scintigraphy, Bone cancer metastases, Semi-supervised learning, Lesion instance segmentation, Convolutional neural network (CNN)

Terapap Apiparakoon : Semi-Supervised Deep Learning with MaligNet for Bone Lesion Instance Segmentation Using Bone Scintigraphy. Advisor: Dr. Ekapol Chuangsuwanich, Ph.D. Co-advisor: Asst. Prof. Dr. YOTHIN RAKVONGTHAI, Ph.D.

One challenge in applying deep learning to medical imaging is the lack of labeled data. Although large amounts of clinical data are available, acquiring labeled image data is difficult, especially for bone scintigraphy (i.e., 2D bone imaging) images. Bone scintigraphy images are generally noisy, and ground-truth or gold standard information from surgical or pathological reports may not be available. We propose a novel neural network model that can segment abnormal hotspots and classify bone cancer metastases in the chest area in a semi-supervised manner. Our proposed model, called MaligNet, is an instance segmentation model that incorporates ladder networks to harness both labeled and unlabeled data. Unlike deep learning segmentation models that classify each instance independently, MaligNet utilizes global information via an additional connection from the core network. To evaluate the performance of our model, we created a dataset for bone lesion instance segmentation using labeled and unlabeled example data from 544 and 9,280 patients, respectively. Our proposed model achieved mean precision, mean sensitivity, and mean F1-score of 0.852, 0.856, and 0.848, respectively, and outperformed the baseline mask region-based convolutional neural network (Mask R-CNN) by 3.92%. Further analysis showed that incorporating global information also helps the model classify specific instances that require information from other regions. On the metastasis classification task, our model achieves a sensitivity of 0.657 and a specificity of 0.857, demonstrating its great potential for automated diagnosis using bone scintigraphy in clinical practice.

Field of Study:	Computer Engineering	Student's Signature .....
Academic Year:	2019	Advisor's Signature .....
		Co-advisor's Signature .....

ACKNOWLEDGEMENTS

I would like to thank my thesis supervisor Ekapol Chuangsuwanich, Ph.D. for his assistance with a suggestion, and Tawatchai Chaiwatanarat, M.D., Yothin Rakvongthai, Ph.D., Maythinee Chantadisai, M.D., Usanee Vutrapongwatana, M.D., Kanuangnit Kingpetch, M.D., and Sasitorn Sirisalipoch M.D. for labeling data and making this project possible. I also thank my project partner, Nutthapol Rakratchatakul from Faculty of Engineering, Chulalongkorn University for advising some methodology. Finally, I would like to thank Thanayut Wiriyatharakij for his help with data collection. We would also like to thank Vivattanachai Sangsa-nga, Thanyaporn Phinthuphan, Penpicha Sangsa-nga, and Panyawut Sri-iesaranusorn for their help with the early versions of our labeling tool.

Terapap Apiparakoon

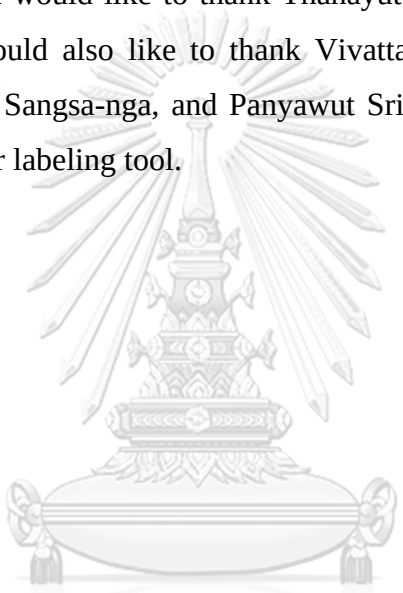


TABLE OF CONTENTS

Page

LIST OF TABLES

Page

LIST OF FIGURES

Page

1. Introduction

1.1 Motivation

At present, 1.7 million patients are diagnosed with cancer each year (Siegel, Miller et al. 2019), and cancer is commonly detected in multiple organs. Once cancer has spread to the bones, it can rarely be cured (Institute 2018). Therefore, bone cancer detection plays a key role in treatment decision making (Ibrahim, Mercatali et al. 2013). Bone scintigraphy is a nuclear medicine procedure that uses radioactivity for bone cancer imaging. Because the spread of cancer often manifests in the bones, clinicians usually request bone scintigraphy results before any type of treatment can be prescribed. The bone scintigraphy results are used in primary decision making as supporting information during screening and for identifying the positions of any abnormal regions, called lesions (Geng, Jia et al. 2015, Magee, Zachazewski et al. 2015).

However, abnormalities found in bone scans include not only cancer but also other bone abnormalities that can be considered benign. A malignant lesion is characterized as a cluster of dangerous tumor cells that can lead to bone cancer metastases (Confavreux, Pialat et al. 2019). To judge whether a lesion is malignant, the nuclear medicine physician must factor in several criteria, such as the pixel intensity reflecting the level of radioactive uptake, the lesion location, the number of lesions, etc. In cases where lesion categorization is difficult due to ambiguous characteristics, the physician might have to increase their time spent for diagnosis to up to an hour per patient to interpret the results. Using machine learning to support this task can help improve efficiency, resulting in better treatment for patients.

The difficulties in applying machine learning to medical imaging applications lie in manual labeling. In this case, labeling bone scintigraphy data requires nuclear medicine physicians. Consequently, the labeling task can be very expensive and time-consuming. It is very likely that only a small portion of the data will be labeled. Furthermore, when the physician is uncertain about the type of lesion, the nuclear medicine physician may label more than one class per lesion (multilabel data), which makes the data labeling more complex. Current instance segmentation methods designed for supervised learning use a large amount of labeled data for training and cannot use unlabeled data and poor results can be obtained when the labeled dataset is small.

Deep learning has become the predominant model for tasks related to medical images. Convolutional neural networks (CNNs) are usually used in such models due to their ability to handle spatial inputs well (Rajpurkar, Irvin et al. 2017). Our work focuses on the use of unlabeled data in addition to the labeled data to improve the model accuracy, a method often called semi-supervised learning. Specifically, our model uses the feature pyramid network (FPN) architecture (Lin, Dollár et al. 2017) as a basis and incorporates the autoencoder structure used in the ladder network (Rasmus, Berglund et al. 2015) to make use of unlabeled data.

Lesion instance segmentation is a type of segmentation task that is responsible for dividing pixels into parts depending on the characteristics of lesions. A segmentation task can be separated into two types: semantic segmentation, which aims to group the pixels in a semantically meaningful way through a pixel-wise classification, and instance segmentation, which is the task that not only segments pixels into groups but also identifies the groups in instances. Generally, region-based approaches (Girshick, Donahue et al. 2014) for object detection are applied in instance segmentation in the first stage. Each region is categorized and segmented into a binary mask (He, Gkioxari et al. 2017).

Normally, classifying the type of objects in the instance segmentation task relies first on the object detection process to identify the regions of interest (ROIs). Each object is classified independently, which might be appropriate in certain tasks. However, for bone scintigraphy, this method cannot be used because categorizing the type of lesion must rely on other lesions in the images. For example, if most lesions are considered malignant, then lesions that are not yet classified are likely to also be malignant. We use global features from the core network to support this line of reasoning. The model utilizes global features by using the overall composition to help determine the type of lesion (Ibrahim, Mercatali et al. 2013).

1.2. Objective

This thesis studies the method to utilize the unlabeled data for increasing the efficiency of the model in a semi-supervised manner for lesion instance segmentation task. The main hypothesis of the thesis is:

The semi-supervised learning method can be applied with instance segmentation to make the model can learn the properties and representation of the lesion better from labeled and unlabeled data. Moreover, knowing the specific properties of the data will give us utilize a particular feature from the data.

In this thesis, there are two main objectives:

1. Ladder network may be an appropriate semi-supervised approach to apply with the feature pyramid network that makes our model can learn labeled and unlabeled data simultaneously.

2. Using the global feature may help the model to categorize the type of lesion by taking into account of the overall structure of the image.

To address both aspects, we do the experiments, analyze, and visualize the results to show the effect of applying each technique.

1.3. Scope

The study provides an alternative approach in the semi-supervised approach for lesion instance segmentation on the chest area in bone scintigraphy. We will focus on utilizing the unlabeled data to make the model more efficient. However, we also give the details in the chest detection process to understand the overview of the model workflow.

2. Related work

There has been a trend of using deep neural networks for medical image analysis. ChexNet (Rajpurkar, Irvin et al. 2017) uses DenseNet (Huang, Liu et al. 2017), a variant of CNNs, for detecting pneumonia from chest X-rays. RIANet (Tong, Li et al. 2019) is an encoder-decoder that can efficiently reuse parameters to encode richer representative features for cardiac MRI segmentation. Three-dimensional roto-translation group convolutions have been applied to detect pulmonary nodules in CT scan images rather than standard translational convolutions to reduce false-positive errors (Winkels and Cohen 2019). A combination of three CNNs is used to automatically localize anatomical ROIs of CT scan images (de Vos, Wolterink et al. 2016).

For landmark detection to locate points of interest, a CNN (Yang, Zhang et al. 2015) was used for the localization of geometric landmarks on the femur surface in 3D MRI. SpatialConfiguration-Net (Payer, Štern et al. 2016) is used to localize multiple landmarks in the hand image using regression heatmaps.

Semantic segmentation is widely applied in the medical image field to group pixels into semantically meaningful segments. For example, the pixels in the same tissue or lesion should be grouped in the same segment. Micro-Net (Raza, Cheung et al. 2019) and DCNet (Küstner, Müller et al. 2018) use CNNs to perform semantic segmentation on microscopy images and multi-contrast MRI, respectively. However, semantic segmentation has difficulties in separating different instances in the same class, which affects the counting and classification of objects.

To solve the problem of separating instances of the same class, an instance segmentation task was introduced both to identify objects and their regions and to segment pixels within such regions. However, there is some overhead in the object detection phase, which takes time in the training process and requires more memory.

Compared to instance segmentation, which is rare in medical image analysis (Xu, Li et al. 2017), the related task of image segmentation is more common. Spine-GAN (Han, Wei et al. 2018) performed semantic segmentation on the spinal region from MRIs. Fully convolutional networks (FCNs) were used for male pelvic organ segmentation on CT scans by (Wang, He et al. 2019). (Ambellan, Tack et al. 2019) used CNNs with a priori knowledge of anatomical shape for knee bone and cartilage segmentation. (Graham, Chen et al. 2019) performed gland segmentation using a modified CNN that reintroduces the original image at multiple points within the network to help reduce the loss of information caused by max-pooling. (Heinrich, Oktay et al. 2019) applied a 3D CNN with 3D CT multi-organ medical images. (Kamnitsas, Ledig et al. 2017) proposed a dual-pathway CNN for brain lesion segmentation. (Bustamante, Gupta et al. 2018) performed four-dimensional segmentation from cardiac 4D flow MRI.

All of the aforementioned works used supervised labels. ChexNet used more than 100,000 chest X-ray images with labels obtained from medical records. This is a high-level classification task where the labels can be relatively easy to acquire. However, for complex tasks such as segmentation, the number of training data samples can be as low as a couple of hundred due to the difficulty of data acquisition and labeling. One way to reduce the effort of data annotation is to use coarse annotation schemes. For example, (Kervadec, Dolz et al. 2019) proposed a constrained-CNN loss for image segmentation on the left ventricle (MRI), vertebral body (MR-T2), and prostate (MR-T2) using segmentation labels that did not cover the entire region.

Another popular approach is to use unlabeled training data to improve the model, a method often called semi-supervised learning. (Cheplygina, de Bruijne et al. 2019) provided a comprehensive overview of semi-supervised methods applied to medical image analysis. Self-training uses a model that is previously trained on labeled data to estimate labels for unlabeled data. (Azmi, Norozi et al. 2011) proposed a self-training approach for breast lesion segmentation using MRI. This simple approach is surprisingly effective when the starting model is sufficiently robust.

Another strategy, called graph-based methods in (Cheplygina, de Bruijne et al. 2019), employs unlabeled data to better learn about the distribution of the data. Our work falls under this category but is based on the deep learning framework. By modifying the loss function to include an unsupervised loss, the training process is simplified because we treat labeled and unlabeled data almost identically.

For the task of bone scintigraphy, which is the domain application for our work, (Dang 2016, Belcher 2017) used CNNs to classify hotspot regions for prostate cancer metastases. This work used approximately 2,000 images due to the difficulty in labeling. (Geng, Jia et al. 2015) used a sparse autoencoder to automatically learn good features for metastasis classification and then used multiple instance learning (MIL) for a patch-level classifier to perform segmentation. EM-MILBoost was later proposed by (Geng, Ma et al. 2016), which further applied expectation-maximization (EM) to MIL to achieve additional improvements in performance.

(Kang, Choi et al.) performed unsupervised lesion detection on bone scintigraphy images using unsupervised learning on normal images in an autoencoder-like manner instead of using semi-supervised learning. Our method uses the supervised and unsupervised data to jointly train our model, which should perform better than a completely unsupervised model in terms of detection capability. Moreover, (Kang, Choi et al.) can only detect lesions but cannot perform classification or segmentation due to the limitation of unsupervised data.

Our work also differs from previous bone scintigraphy-related works in that our task is instance segmentation, which means that we classify the lesion type and perform segmentation of each lesion. While both semantic segmentation and instance segmentation can identify the location of lesions, when two lesions are overlapping or adjacent to each other, instance segmentation can detect the two lesions as two separate entities. Moreover, previous works classified lesions only as metastatic or

nonmetastatic, whereas our work can classify each lesion into finer classes, i.e., malignant, degenerative change, post-trauma, and inflection/inflammation. This classification is closer to the current clinical practice for bone scintigraphy. In some cases, instance segmentation can help the model better differentiate malignant from nonmalignant lesions. For example, if the model understands that lesions on different ribs that form in an orderly manner into a straight line should be classified as post-trauma, then it will be easier for instance segmentation models than for semantic segmentation models to consider this correlation.

3. Background

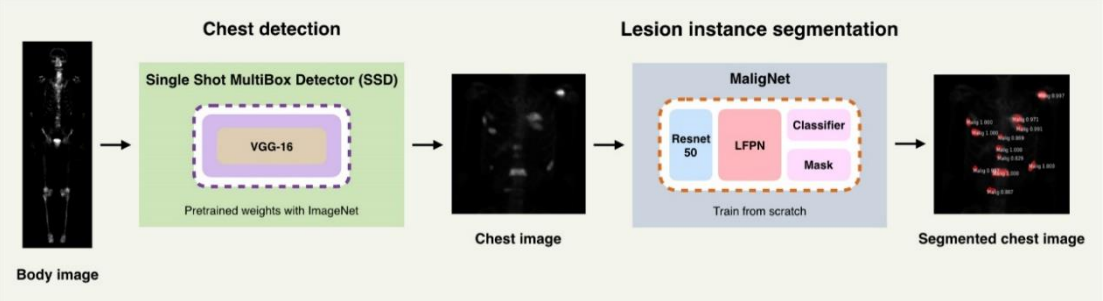


Figure 1: An overview of our model workflow. The whole-body bone scintigram (left image) was passed into the Single Shot MultiBox Detector (SSD) to detect the chest area (middle image) and sent to the MaligNet model for lesion instance segmentation (right image).

3.1. The overview of the model workflow

Our overall system consists of two parts: the chest localization model, which localizes the chest area, and the instance segmentation model, which segments and classifies each lesion, as shown in Figure 1. In this paper, we focus on MaligNet, a model for lesion instance segmentation on the chest area in bone scintigraphy that has various internal components. We provide some background for each component and works related to them. In the field of machine learning systems, there are many approaches to build a model from learning the data patterns. Machine learning systems can be classified according to the amount and type of supervision whether or not they are trained with human supervision.

3.2. Supervised learning

Supervised learning is the most popular approach that people always use in various tasks. This learning method required the solution (always called labels or ground truth) to teach the model. When you feed the data into your model, the model will learn to solve the tasks based on the labels of each data. For this reason, the labels must be clean and correct, to make the model understand the correlation between data and labels. The examples of the most important supervised learning algorithms are:

- Linear Regression
- Logistic Regression
- Decision Trees and Random Forests
- k-Nearest Neighbors
- Support Vector Machines (SVMs)
- Neural networks

3.3. Unsupervised learning

For unsupervised learning, this learning system does not require the labels of data (often called unlabeled data). Unsupervised learning tries to learn without a teacher. This approach can provide useful clues for how to group examples in representation space. The examples of unsupervised learning algorithms that are:

- Clustering
  - K-Means
  - DBSCAN
  - Hierarchical Cluster Analysis (HCA)
- Anomaly detection and novelty detection
  - One-class SVM
  - Isolation Forest
- Visualization and dimensionality reduction
  - Principle Component Analysis (PCA)
  - Kernel PCA
  - Locally-Linear Embedding (LLE)
  - t-distributed Stochastic Neighbor Embedding (t-SNE)
- Association rule learning
  - Apriori
  - Eclat

3.4. Semi-supervised learning

Supervised learning requires the ground truth for the model to learn correlations between data and labels. Unsupervised learning learns the representation of the group of data. Both algorithms can deal with only labeled or unlabeled training data. The semi-supervised learning can use with partially labeled training data, usually a lot of unlabeled data and a small amount of labeled data. Semi-supervised learning is usually integrations of supervised and unsupervised algorithms. For example:

- Restricted Boltzmann machines (RBMs)
- Deep belief network (DBNs)
- Self-training

3.4.1. Self-training approach

Self-training was used in this thesis to compare with our model. Therefore, we will give some details about the self-training. Self-training produce labels from labeled to the unlabeled data and then using the larger, newly labeled set for training. Self-training will only be effective when we have enough labeled data to give the model high confidence predictions are correct.

3.5. Activation functions

Activation functions or Transfer functions are designed to convert an input signal of a node in a neural network to an output signal. Activation function giving the neuron know the bounds of the value whether the neuron should activate or not. Moreover, these functions aim at the mapping between the inputs and response variables. The activation function can be divided into two types which are:

3.5.1. Linear activation function

A linear activation function is a straight line function where activation is proportional to the input following this equation.

$$F(\boldsymbol{x}) = \boldsymbol{x} \quad (1)$$

The linear function is designed for a non-complex model because it has a polynomial of one degree. For this reason, a linear activation function is limited in its complexity and no ability to learn the complex neuron. The output of this function can be any value without boundary.

3.5.2. Non-linear activation function

The non-linear activation functions are the most used activation function. These functions make the model to generalize or to deal with complexity between the output. In this thesis, we will focus on popular non-linear activation functions that are Sigmoid, Tanh, ReLU, and Softmax activation functions.

3.5.2.1. Sigmoid or Logistic activation function

In the case of choosing sigmoid function as an activation function, the output value after passed the sigmoid function is between zero to one. Thus, it is used to predict probability as an output. Due to this, the probability exists only between the range of 0 and 1. The sigmoid function has the equation as follows

$$\sigma(\boldsymbol{x}) = \frac{1}{1 + e^{-x}} \quad (2)$$

3.5.2.2. Tanh activation function

Tanh or hyperbolic tangent activation function is similar to sigmoid function but tanh function is more wide-ranging from -1 to 1. The advantage is that this function is supported by negative input and it can also produce a negative output. The tanh function has the equation as follows

$$Tanh(\boldsymbol{x}) = tanh\, \boldsymbol{x} \quad (3)$$

Notice that the sigmoid and tanh activation function is used in a feed-forward neural network.

3.5.2.3. ReLU activation function

ReLU or Rectified Linear Unit is the most used activation function. ReLU is a half-rectified activation function that look like a linear function but it is a non-linear function. The output value of this function will be zero if the input value is negative. For the positive input value, the output will be obtained from a linear function. The range of ReLU is from 0 to infinite which has the equation as follows

$$ReLU(\boldsymbol{x}) = max(0, \boldsymbol{x}) \quad (4)$$

3.5.2.4. Softmax activation function

Softmax function always used in a classification task. Softmax converts the input logits into probabilities that sum to one. The output of this function is a vector that represents the probability distributions of a list of potential outcomes. The softmax function has the equation as follows

$$\boldsymbol{S(y_i)} = \frac{\boldsymbol{e^{y_i}}}{\sum_{\boldsymbol{k=1}}^{\boldsymbol{K}} \boldsymbol{e^{z_k}}} \quad (5)$$

3.6. Cost function

The cost function, or lost function, or objective function, or criterion is the function that we want to minimize or maximize to measure the performance of the machine learning model. Cost function estimates the error between the prediction result and the actual values. The value of cost function is present in the form of a single real number. There are many types of cost function in machine learning but I will inform you of some details about the cost function used in this thesis.

3.6.1. Mean square error (MSE)

Mean square error or MSE is a cost function to measure the average of the squares of the errors. MSE is the average squared difference between the prediction output and the ground truth which define as:

$$MSE = \frac{1}{N}\sum_{n=1}^{N} (y_i - \tilde{y}_i)^2 \quad (6)$$

3.6.2. L1 norm

The L1 norm or Manhattan Distance is the sum of the magnitudes of the vector's space. It always used to measure the distance between vectors. The distance is calculated from the sum absolute difference components vectors.

$$\|\boldsymbol{x}\|_1 = |\boldsymbol{x}_1| + |\boldsymbol{x}_2| + \ldots + |\boldsymbol{x}_n| \quad (7)$$

3.6.3. L2 norm

The L2 norm or Euclidean Distance is similar to L1 but it was calculated as the square root of the sum of the squared vector values. The equation of the L2 norm is as follow:

$$\|\boldsymbol{x}\|_2 = \sqrt{|\boldsymbol{x}_1|^2 + |\boldsymbol{x}_2|^2 + \ldots + |\boldsymbol{x}_n|^2} \quad (8)$$

Notice that, both L1 and L2 norms are used in machine learning to avoid the model overfit and make the model more generalization to the data.

3.6.4. Smoothed L1

Smooth L1 is a combination of L1 and L2 norms. The output values were calculated by the L1 norm in case of the absolute value of arguments is higher than α. On the other hand, the L2 norm is used, if the absolute value of arguments is lower than α. The advantages of combining between both norms are L1 norm has steady gradients for a large value of x and the L2 norm has fewer oscillations during updates when x is small. The equation of Smoothed L1 norm is as follow:

$$s(x) = \begin{cases} x - 0.5 & if\ |x| > \alpha; \\ 0.5x^2 & if\ |x| \leq \alpha; \end{cases} \quad (9)$$

3.6.5. Cross-entropy

Cross-Entropy or Log loss is a cost function that always found in a classification task. This cost function measures the performance of a classification model that the output is a probability of range between 0 and 1. Cross-entropy cost increases the predicted probability diverges from the actual values.

3.6.5.1. Categorical cross-entropy

Categorical Cross-Entropy or Softmax loss is a combination of softmax and the cross-entropy activation function. In a multi-class classification task, the labels are one-hot that means only the positive class is used to calculate the cost. So the equation of categorical cross-entropy is as follow:

$$CE = -\sum_{i=1}^{C} t_i \, log(f(s)_i) \quad (10)$$

After combining with Softmax activation function the equation will be as:

$$CE = -\sum_{i=1}^{C} t_i \, log(\frac{e^{s_i}}{\sum_j^C e^{s_j}}) \quad (11)$$

3.6.5.2. Binary cross-entropy

Binary Cross-Entropy is also called Sigmoid Cross Entropy. This cost function is a combination of the sigmoid and the cross-entropy like categorical cross-entropy. Binary Cross-Entropy always in a binary classification task and a multi-label classification task. The cost computed for every output vector component is independent which not affect the other values. Binary cross-entropy was defined as:

$$CE = -\sum_{i=1}^{C=2} t_i \log(f(s_i))$$

$$CE = -t_1 \log(f(s_1)) - (1 - t_1) \log(1 - f(s_1)) \quad (12)$$

3.7. Regularization

Regularization is used in the machine learning model to give the model avoid to be overfitting or more generalized. Generally, regularization is adapted from the norm or distance function which is L1 and L2 norm to apply with the weight of the model. The aim of regularization is to discourage the complexity of the model. It also combines within the cost function term.

3.8. Metrics

Our experiments were divided into two parts: lesion instance segmentation and bone cancer metastases prediction. In the bone cancer metastases prediction task, we evaluate in a different metric include precision, recall, f1-score, accuracy, and specificity to be an indicator of the effectiveness of models. The details of each type of measurement are described separately in each section.

Table 1: The confusion matrix between cluster labels true positive (TP), false positive (FP), true negative (TN), and false-negative (FN).

Actual classes	Predicted classes	
	Positive	Negative
Positive	True Positive (TP)	False Negative (FN)
Negative	False Positive (FP)	True Negative (TN)

Accuracy is the most intuitive performance measure and it is simply a ratio of correctly predicted observation to the total observations which was defined as:

$$Accuracy = \frac{(TN + TP)}{(TN + TP + FN + FP)} \quad (13)$$

Precision is the ratio of positive observations correctly predicted to the positive total predicted observations :

$$Precision = \frac{TP}{(TP + FP)} \quad (14)$$

Recall or Sensitivity is the ratio of positive observations correctly predicted to the all actual class - yes in observations:

$$Recall = \frac{TP}{(TP + FN)} \quad (15)$$

Specificity is the proportion of images that tested negative and are negative of all the images that actually are negative :

$$Specificity = \frac{TN}{(TN + FP)} \quad (16)$$

F1-score is the weighted average of Precision and Recall. consequently, this score takes into consideration both false positives and false negatives :

$$F1 = 2 \cdot \frac{(Recall * Precision)}{(Recall + Precision)} \quad (17)$$

Our experiments were divided into two parts: lesion instance segmentation and bone cancer metastases prediction. Instead of using the mean average precision (mAP), which is a relative score metric that is used to evaluate object detection in natural images such as Pascal VOC (Everingham, Van Gool et al. 2010), COCO (Lin, Maire et al. 2014) and Open Images, we use the mean precision, the mean sensitivity, and the mean f1 to measure the performance of our model in lesion instance segmentation. In the context of instance segmentation, we must not only correctly identify the object but also correctly locate its position. Thus, to calculate the mean precision and the mean sensitivity, the Jaccard index (Intersection over Union) is also used to measure the overlapping region between the ground truth and the predicted area. The Jaccard index is defined as follows:

$$J(A_1, A_2) = \frac{|A_1 \cap A_2|}{|A_1 \cup A_2|} \quad (18)$$

where A_1 is the area of the ground truth and A_2 is the area of the prediction.

A prediction is considered correct if the Jaccard index is above a predetermined threshold. We chose a threshold of 0.5 since this is sufficient for locating the lesion. For multiclass detection tasks such as ours, we can calculate mean precision and mean sensitivity by taking the weighted average of the precision and sensitivity values, respectively.

MaligNet is designed for lesion instance segmentation. Therefore, we cannot directly evaluate the performance of the model in the metastasis classification task. To do so, we convert the instance segmentation predictions to a binary prediction. If the model predicts malignancy for at least one lesion in the chest area, the image will be classified as metastasis. However, we will only consider the case where at least one malignant prediction matches the ground truth as a true positive sample.

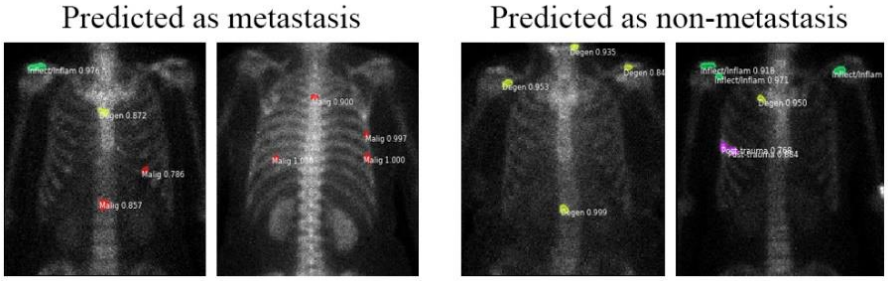


Figure 2: Example predictions for the bone cancer metastasis prediction task. If there is at least one malignant lesion predicted, the image will be classified as metastasis.

In other words, a metastasis prediction that is caused by a false alarm in the instance segmentation task will not be counted as a correct classification. Examples of interpretation in bone cancer metastasis prediction are shown in Figure 2.

In cases in which the model does not find any malignant lesions in the image or finds another lesion type, such as degenerative change, inflection/inflammation, or post-traumatic, we assume that the model predicts non-metastasis status or a negative sample. We also evaluated our model of bone cancer metastasis prediction in terms of various metrics, namely accuracy, precision, recall (sensitivity), specificity, and f1-score. The details of each type of measurement are described separately in each section.

3.9. Convolutional neural network (CNN)

Convolutional neural networks (CNNs) is one kind of a neural network, which is designed for extracting the data in a grid-cells including time-series data. Convolution is a special kind linear operation that uses convolution in place of general matrix multiplication in at least of their layers.

3.10. Object detection task

In this thesis, the main task of our work is an instance segmentation task, which is applied to the object detection task. Object detection task is a task, which aims to detect the instance of semantic objects in the images. In the object detection task, there are two types of procedures for detecting the object in the images which are one-stage object detection and two-stage object detection.

3.10.1. One-Stage object detection

In one-stage object detection, the number of predictions on the grid will be fixed. That means the model directly predict the object bounding and classes simultaneously. In other words, there is no intermediate task. For this reason, a one-stage detector is more simple and faster than a two-stage detector. However, one-stage detection often has lower average precision than two-stage detection. The examples of popular one-stage detectors are YOLO (You Only Look Once) (Redmon, Divvala et al. 2016), SSD (Single Shot MultiBox Detector) (Liu, Anguelov et al. 2016), and RetinaNet (Lin, Goyal et al. 2017).

3.10.2. Two-Stages object detection

The region proposal network is proposed in two-stages detection. The proposal network aims to find the object proposal in the first stage and fine-tune the proposal and output prediction in the second stage. The two-stages is often slower than one stage because it takes time in finding proposal but often more accuracy than a one-stage detector.

3.10.2.1. R-CNN

R-CNN (Girshick, Donahue et al. 2014, Ren, He et al. 2015) is the first of object detection using that use CNNs in region base family. It uses a selective search algorithm to generate 2,000 region proposals on the image. After that, the model classifies each proposal using CNNs for feature extraction and fed into Support Vector Machine(SVM) to classify the presence of the object. The problem of R-CNN is it takes time in the training process due to 2,000 region proposal would have to classify per images. Moreover, the selective search algorithm can not learn to generate the proposal from the image since it is a fixed algorithm.

3.10.2.2. Fast R-CNN

For solving the problem with training time due to a huge region proposal. The Fast R-CNN (Girshick 2015) was proposed to solve the training time problem in R-CNN. Instead of feed each proposal into CNNs, they fed the input image to the CNNs, using a feature map to identify the region of the object, and also wrapping them in a fixed size by ROI pooling. Finally, each ROI feature was passed into the softmax layer to classify the class of that object. For this reason, Fast R-CNN is faster than R-CNN significantly. However, it also uses a selective search algorithm which makes the model still slow in finding the region proposal.

3.10.2.3. Faster R-CNN

To solve the problem of R-CNN and Fast R-CNN about using selective search for finding region proposal, that makes the model slow and time-consuming. Faster R-CNN (Ren, He et al. 2015) uses a neural network to learn the region proposals called Region Proposal Network. Region proposal network (RPN) is a kind of fully convolutional neural network. The RPN is a module that generates such region proposals. The RPN is designed for detecting objects on the convolutional feature maps from the backbone. It predicts region proposals of various scales and aspect

ratios using multiple anchor boxes. Specifically, for each location and scaling factor on a regular grid, the RPN outputs object region boundaries and their associated objectless scores which specify how likely each region proposal will contain an object of interest. Due to RPN takes a few overheads to learn to generate the proposals instead of generating 2,000 proposals that make the detector faster than both of the above algorithms.

3.10.2.3.1 Region proposal network (RPN)

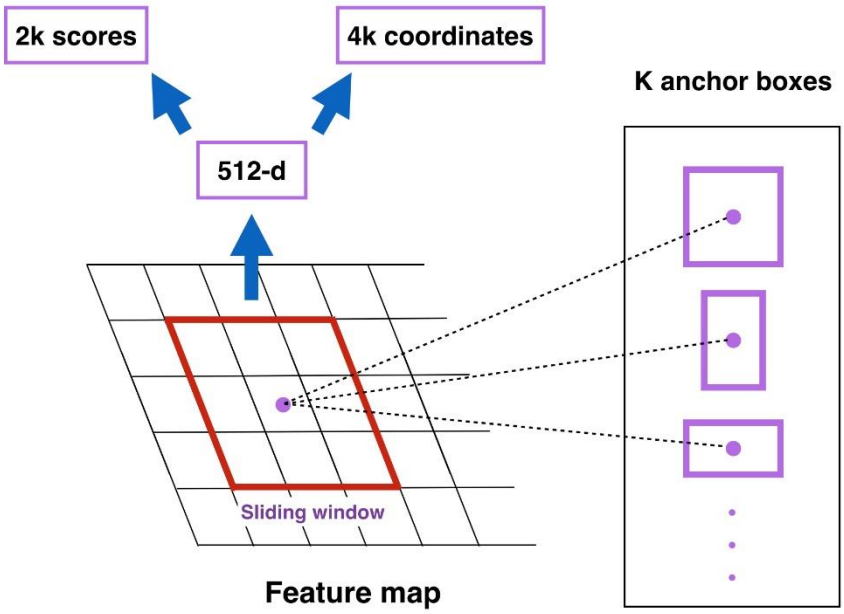


Figure 3: The illustration of the region proposal network (RPN) in which the input is a feature map. The RPN produces 2k anchor scores and 4k bounding box coordinates per pixel in the feature map, where k is the number of anchor boxes.

The region proposal network (RPN) is a type of fully convolutional network that is used in Faster R-CNN (Ren, He et al. 2015). This model is in the region-based family, which includes R-CNN (Girshick, Donahue et al. 2014), Fast R-CNN (Girshick 2015) and Mask R-CNN (He, Gkioxari et al. 2017). Region-based object detectors first identify potential regions for objects and then classify each region into object classes. The RPN is a module that generates such region proposals. It is designed for detecting objects on the convolutional feature maps from the core network. The module predicts region proposals of various scales and aspect ratios using multiple anchor boxes.

Specifically, for each location and scaling factor on a regular grid, the RPN outputs object region boundaries and their associated objectless scores, which specify how likely each proposed region is to contain an object of interest, as shown in Figure 3. The cost function for classifying each region proposal R_c is a categorical cross-entropy loss which was defined as:

$$R_c = -\frac{1}{N}\sum_{n=1}^{N}\sum_{a=1}^{A} log\ P\ (\tilde{y}\ =\ y_{n,a} \mid x_{n,a}) \qquad (19)$$

where N is the minibatch size, and A is the number of anchors per image.
For the region bounding box, the smoothed-L1 loss was used as the cost function for the bounding box prediction, as shown below:

$$d_{n,a} = \left\| b_{n,a}{}^{*} - b_{n,a} \right\|_1 \tag{20}$$

$$s(x) = \begin{cases} x - 0.5 & if\ |x| > 1 \\ 0.5x^2 & otherwise, \end{cases} \tag{21}$$

$$R_b = -\frac{1}{N} \sum_{n=1}^{N} \sum_{a=1}^{A} s(d_{n,a}) \tag{22}$$

where $\|\ \ \|$ denotes the L1 norm; $b_{n,a}{}^{*}$ and $b_{n,a}$ are vectors containing the coordinates of the predicted bounding box and labeled bounding box, respectively; s(x) is the smoothed L1 loss; and R_b is the sum of region proposal bounding box loss in all anchors.

3.11. Segmentation task

Image segmentation is the process of segmenting the image into multiple regions of pixels. That makes the image more simple and changes the representation into more meaningful to understand.

3.11.1. Semantic segmentation

Semantic segmentation is the high-level task to understand the image. The goal of semantic segmentation is to assign pixels to their classes without separate the individual instances in the image. In this task, the pixels in all regions would have been assigned to a class that treats thing classes as stuff.

3.11.2. Instance segmentation

Instance segmentation was used in our thesis. It takes two steps in the segmentation process. First, it takes object detection to detect the instance in the image. After that, each instance would have to identify which pixels belong to that instance. To give the details about instance segmentation, we will give the example of instance segmentation model that we used as the baseline in the experiment of this thesis

3.11.2.1. Mask R-CNN

Mask R-CNN (He, Gkioxari et al. 2017) is an instance segmentation model that uses the Faster R-CNN in the first step for feature extraction, extending with a mask frontend for segmentation. Mask R-CNN is designed to solve the problem of a multi-scale object by using a feature pyramid network (FPN) (Lin, Dollár et al. 2017) to generate multiple-scale feature maps.

3.11.2.1.1. Feature Pyramid Network (FPN)

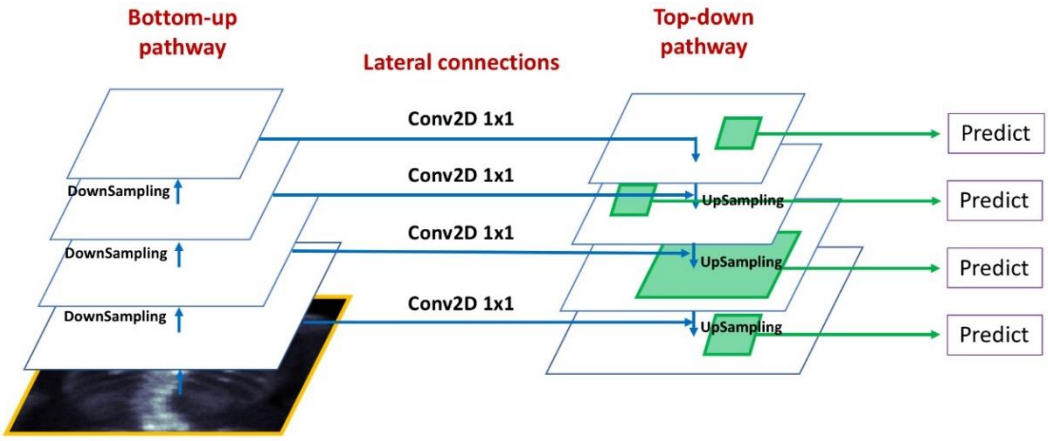


Figure 4: The illustration of the feature pyramid network (FPN). The FPN consists of the bottom-up pathway and top-down pathway. The bottom-up pathway is the feed-forward neural network of the core. The top-down pathway is the ConvNet which upsamples spatial coarser high-level features combining with low-level features through lateral connections.

The feature pyramid network (Lin, Dollár et al. 2017) is chosen as the core component in MaligNet for instance segmentation, as shown in Figure 4. We chose FPN because it was designed to detect objects of different scales, which is the case for lesions in a chest image. FPN consists of two main parts: bottom-up and top-down pathways. The bottom-up pathway is the feedforward neural network. The top-down pathway, which is connected by a bottom-up pathway through lateral connections, is designed for building semantic feature maps at all scales by double upscaling to enhance the feature maps from the bottom-up pathway. Combining high-resolution but semantically weak features with low-resolution but semantically strong features via a lateral connection and top-down pathway impart rich semantics at all levels of the FPN.

Another difference thing between Mask R-CNN and Faster R-CNN is the Mask frontend. Mask frontend is a convolutional neural network designed for object segmentation. The segmented proposals were separated into foreground and background.

3.12. Autoencoder

Autoencoder is an unsupervised neural network that aims to learn to compress the information of data. The compressed features keep important information about the data. Furthermore, it learns to reconstruct the reduced encoded representation that is close to the original input.

3.12.1. Denoising autoencoder

Denoising autoencoder (DAE) (Vincent, Larochelle et al. 2008) is an autoencoder that is mostly used in denoising the data. Furthermore, DAE can solve the problem of overfitting in autoencoder in case of more nodes in the hidden layers than the input layers. Denoising autoencoder aims to corrupt or add the noise to the data randomly about 50%. To minimize the cost function, DAE would have learned the important features of the data that can reconstruct the corrupted features as close as the original input.

3.12.2. Ladder network

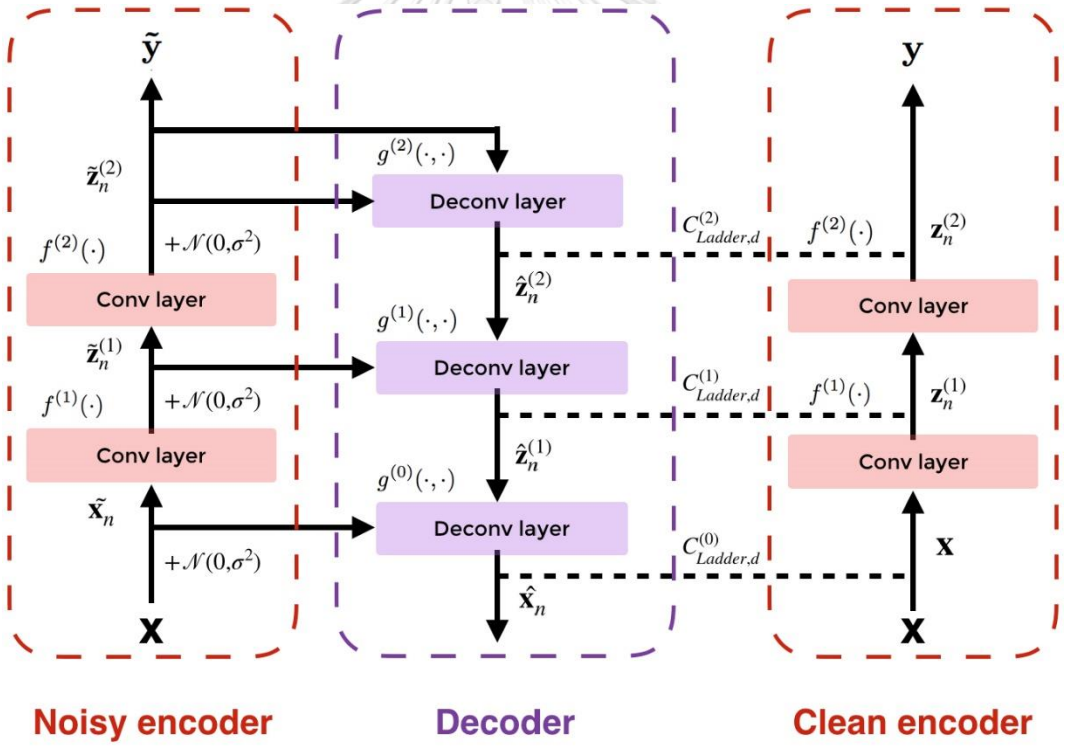


Figure 5: The structure of the Ladder network which is a convolutional neural network consists of two parts of neural networks: encoder and decoder. The encoder includes a clean encoder $(x \to z_{(i)} \to y)$ *and a noisy encoder* $(x \to \tilde{z}_{(i)} \to \tilde{y})$*. Both clean and noisy encoder will share the same mapping function f. The decoder* $(\tilde{z}^{(m)} \to \hat{z}^{(m)} \to \hat{x})$ *will perform reconstructs the information from noisy encoder compare with the clean encoder in lateral connections with function g, which is denoising function. Considering in verticle connections, the cost is caused by*

supervised learning ($C_{Ladder,c}$) as shown in equation (26). For lateral connections, $C_{Ladder,d}$ is costs from unsupervised learning of all layers as shown in equation (27).

A ladder network is a semi-supervised learning method (Rasmus, Berglund et al. 2015) that can utilize labeled and unlabeled data simultaneously. The ladder network is similar in concept to the denoising autoencoder (DAE). DAE is a type of an autoencoder that receives a corrupted data as input and is trained to remove the noise that is introduced to the input to uncover the uncorrupted input data at the output layer (Goodfellow, Bengio et al. 2016). Ladder network takes this a step further by introducing noise at every layer, not just the input. Figure 5 illustrates a simple Ladder network. The network consists of three parts: the corrupted encoder (the leftmost stack), the denoising decoder (the middle stack), and the original network (the rightmost stack). Let's first consider the original network. Let z(m) be the output of the m-th layer of the clean encoder. The function $f_{(m+1)}(\cdot)$ represents the $(m+1)$-th layer which in our case is a convolutional layer. The relationship between each layer can be given as:

$$\boldsymbol{z}^{(m+1)} = f^{(m+1)}\left(\boldsymbol{z}^{(m)}\right) \quad (23)$$

Note that $\boldsymbol{z}^{(0)}$ refers to the network input, $\boldsymbol{x}$. The network will yield the final prediction, $\boldsymbol{y}$.

The corrupted network uses the same weights and layers as the original network. However, we corrupt the inputs preceding each layer by adding Gaussian noise.

$$\tilde{\boldsymbol{z}}^{(m+1)} = f^{(m+1)}(\tilde{\boldsymbol{z}}^{(m)} + \mathcal{N}) \quad (24)$$

Where $\mathcal{N}$ refers to Gaussian noise with zero mean and variance σ^2 variance. $\tilde{\boldsymbol{z}}^{(m)}$ is the output of layer the m-th layer in the corrupted network.

Finally, the denoising decoder wants to recover the original output at each layer by using only the information from the corrupted network. Let $g^{(m)}$ be the inverse mapping function for the m-th layer which in this case is a transposed convolutional layer which outputs $\hat{\boldsymbol{z}}^{(m)}$ and takes $\tilde{\boldsymbol{z}}^{(m)}$ and $\hat{\boldsymbol{z}}^{(m+1)}$ as inputs.

$$\hat{\boldsymbol{z}}^{(m)} = g^{(m)}(\tilde{\boldsymbol{z}}^{(m)}, \hat{\boldsymbol{z}}^{(m+1)}) \quad (25)$$

The supervised cost, $C_{Ladder,c}$, is the average negative log probability of the noisy output $\tilde{y}$ matching the ground truth target t given input $\boldsymbol{x}$.

$$C_{Ladder,c} = -\frac{1}{N}\sum_{n=1}^{N} log\, P(\tilde{y} = t_n|\boldsymbol{x}_n) \quad (26)$$

Where N is the mini-batch size, n is the index of the training data.

The goal of the denoising decoder is to make $\hat{\boldsymbol{z}}^{(1)}$ matches with $\boldsymbol{z}^{(m)}$. This is accomplished by adding an unsupervised cost function, $C_{Ladder,d}$, which tries to minimize the mean square error as shown below:

$$C_{Ladder,d} = \sum_{m=0}^{M} {C^{(m)}}_{Ladder,d} = \sum_{m=0}^{M} \frac{1}{N w_l} \sum_{n=1}^{N} \left\| {z_n}^{(m)} - {\hat{z}_n}^{(m)} \right\|^2 \quad (27)$$

Where w_l is the layer's width (for the dense layer), L the number of layers in the ladder network.

The Ladder network uses this unsupervised loss to learn important information about the data. The two losses ($C_{Ladder,c}$ and $C_{Ladder,d}$) can be combined and trained jointly.

4. Proposed method

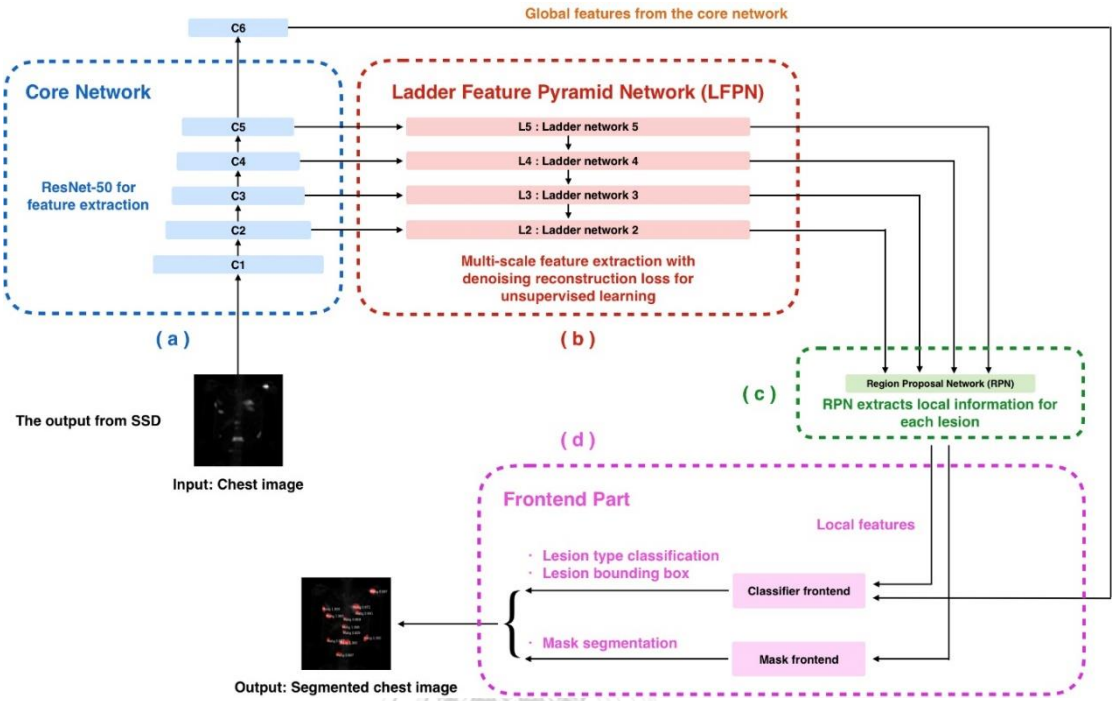


Figure 6: An overview of MaligNet. Figure 6a (in blue) contains the core network of MaligNet, a ResNet-50, extracts the features at different scales to feed to the ladder feature pyramid network (LFPN). Figure 6b (in red) is the LFPN, a feature pyramid network combined with a ladder network to facilitate semi-supervised learning. Figure 6c (in green) is the region proposal network (RPN) that selects regions of interest for object classification and regression. Figure 6d (in pink) contains the frontend part, which consists of a classifier frontend (Figure 7) and a mask frontend (Figure 8). The classifier frontend performs lesion classification and refines the bounding box. The mask frontend outputs the segmentation masks. Unlike Mask R-CNN, the classifier frontend also receives global features from the core network.

Our proposed model is a neural network for lesion instance segmentation called MaligNet. Although lesions can occur anywhere throughout the body, they are often found in the chest area. This area is often the hardest to diagnose due to the complexity and overlap of ribs in the chest area, which consists of small bones. Therefore, we focused only on finding the lesions in the chest area. Figure 1 shows an overview of our system. We start by locating the chest area using the Single Shot MultiBox Detector (SSD) (Liu, Anguelov et al. 2016) in both anterior and posterior views of the whole body in the bone scintigram. Then, the chest area is used in the lesion instance segmentation process.

4.1. Chest detection

The chest detector is the first part of our pipeline used to detect the chest area. Because it is relatively straightforward to detect the chest area, we use a standard SSD to detect both the anterior and posterior chest areas. We choose the SSD because it is a one-stage detection model that has high speed in training and inference. Moreover, it maintains good accuracy compared to other object detection models and is easy to adapt and apply to our task. We use VGG-16 (Simonyan and Zisserman 2014) as the backbone in the SSD. VGG-16 was pretrained with ImageNet (Deng, Dong et al. 2009) and fine-tuned using our data. The hyperparameters of retraining SSD are shown in Table 9.

4.2. Lesion instance segmentation

MaligNet is a CNN model based on FPN with modifications. More specifically, we add the ladder feature pyramid network to the top-down pathway to allow for semi-supervised training. We also add an additional layer that extracts the global features from the core network to the classifier frontend. As shown in Figure 6, MaligNet consists of four parts. Similar to the FPN, the first part is an image classification core model that is used to extract features. We have tried several standard architectures for the choice of the core model (see Table 10) and ultimately settled on ResNet-50 (He, Zhang et al. 2016). ResNets have the nice property of using a stride of two for every scale reduction. This makes incorporating ResNet-50 into the FPN straightforward when we have to upscale the feature maps in the top-down pathway. Moreover, ResNet-50 is a relatively small network based on modern standards and is thus appropriate for our limited labeled data.

The second part is the ladder feature pyramid network (LFPN), which corresponds to the top-down pathway of the FPN but with additional denoising decoder components inspired by the ladder network (Rasmus, Berglund et al. 2015). This allows MaligNet to utilize the training data from both labeled and unlabeled data simultaneously. The features from the top-down pathway are used by the third part, which detects and localizes the lesions using the RPN. The frontend part is designed for lesion instance segmentation as the final part of our model and has two output results. The classifier frontend adjusts the bounding box and categorizes each lesion into different classes. The mask frontend is used for lesion segmentation. However, unlike the traditional FPN, which focuses on local information in each region, the classifier frontend also exploits the global information taken from the topmost level of the core network.

4.2.1. Ladder feature pyramid network (LFPN)

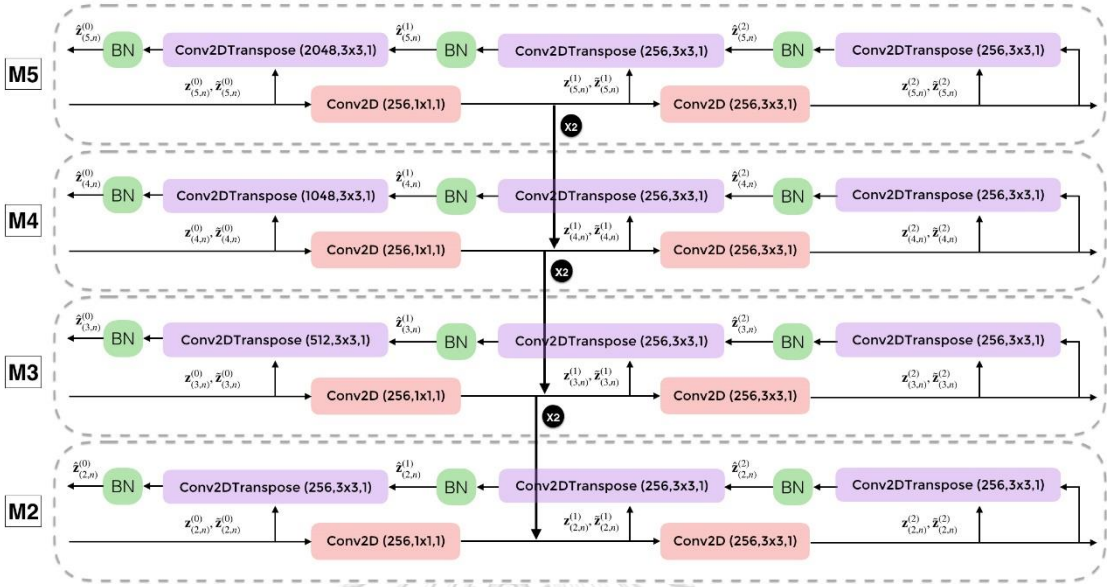


Figure 7: Illustration of the ladder feature pyramid network (LFPN), the feature pyramid network combined with the ladder network to enhance the features by unsupervised learning. In the figure, we have four lines of convolutional neural networks (red blocks) in the feature pyramid network, and each line represents an encoder. We add a decoder to invert the mapping on each layer of the encoder (purple blocks); thus, we have a total of four ladder networks, represented as L2, L3, L4, and L5. The upsampling layer (shown as the x2 symbol) in the LFPN is a bicubic algorithm for scaling-up the feature map.

To allow semi-supervised learning, we incorporated ladder-network-like structures into each level of the FPN. The new structure is referred to as the LFPN. As shown in Figure 7, each lateral connection will be similar to an encoder part in the ladder network. The lateral connections that do not add noise into the features are considered a clean encoder $\mathbf{z}^{(m)}{}_{(l,n)}$ which is defined as:

$$\mathbf{z}^{(m+1)}{}_{(l,n)} = \begin{cases} g(f^{(m+1)}{}_{l}(\mathbf{z}^{(m)}{}_{(l,n)}), \mathbf{z}^{(m+1)}{}_{(l+1,n)}) & if\ m = 0 \\ f^{(m+1)}{}_{l}(\mathbf{z}^{(m)}{}_{(l,n)}) & if\ m = 1 \end{cases} \quad (28)$$

The main difference here from the regular ladder network is the additional connection from the upsampling layer (denoted as x2 in Figure 7).

For the noisy encoder, the features in the LFPN are as follows: $\tilde{\mathbf{z}}^{(m)}{}_{(l,n)}$.

$$\tilde{\boldsymbol{z}}^{(m+1)}{}_{(l,n)} = \begin{cases} g(f^{(m+1)}{}_{l}(\tilde{\boldsymbol{z}}^{(m)}{}_{(l,n)}), \tilde{\boldsymbol{z}}^{(m+1)}{}_{(l+1,n)}) + \boldsymbol{h}^{(m+1)}{}_{(l,n)} & if\ m = 0 \\ f^{(m+1)}{}_{l}(\tilde{\boldsymbol{z}}^{(m)}{}_{(l,n)}) + \boldsymbol{h}^{(m+1)}{}_{(l,n)} & if\ m = 1 \end{cases} \quad (29)$$

where *f* is the convolution function and *k* is the feature combination function. Note that $\mathbf{z}^{(0)}{}_{(l,n)}$refers to $\mathbf{x}^{(0)}{}_{(l,n)}$, and $\tilde{\boldsymbol{z}}^{(0)}{}_{(l,n)}$ refers to $\mathbf{x}^{(0)}{}_{(l,n)} + \boldsymbol{h}^{(m+1)}{}_{(l,n)}$.

Here, the noise for all layers is sampled from a Gaussian distribution with zero mean at a fixed variance level, which is a hyperparameter that is tuned (see Table 10). Even though bone scintigraphy has a Poisson noise distribution (Tsui, Beck et al. 1981) in the raw image, the aim of adding noise in the LFPN is to augment the feature space, not the raw image. The weight distribution in the network is Gaussian. Thus, the noise injection in the LFPN is chosen to be Gaussian instead of Poisson. We also tried to inject Poisson noise in the LFPN, but the resulting model performed worse than the baseline FPN model.

To denoise the noisy features, a transposed convolution layer is used in the denoising decoder because it is an inverse function of the convolutional layer that (Rasmus, Berglund et al. 2015) (Zeng, Yu et al. 2017) used in CNN-Ladder. The reconstruction $\hat{\boldsymbol{z}}^{(m)}{}_{(l,n)}$ is the output of its upper layer $\hat{\boldsymbol{z}}^{(m+1)}{}_{(l,n)}$ and the noisy lateral layer $\tilde{\boldsymbol{z}}^{(m)}{}_{(l,n)}$ by the Conv2DTranspose layer and batch normalization. We added a denoising decoder (the purple blocks in Figure 7) to each lateral connection of the FPN such that we can incorporate the unsupervised loss. For each lateral connection (M2 to M5 in the figure), there are three targets to perform denoising, which correspond to the outputs from different layers on that level. Thus, the unsupervised loss from equation (27) becomes the following:

$$\begin{aligned} C_{LFPN,d} &= \sum_{l=2}^{L=5} \sum_{m=0}^{M=2} C^{(m)}{}_{LFPN,d} \\ &= \frac{1}{NLM} \sum_{l=2}^{L=5} \sum_{m=0}^{M=2} \sum_{n=1}^{N} \left\| z_{l,n}{}^{(m)} - \hat{z}_{l,n}{}^{(m)} \right\|^2 \end{aligned} \quad (30)$$

where *N* is the minibatch size, *L* is the number of levels in the LFPN, and *M* is the number of layers in each lateral connection.

During the training process, the RPN takes the noisy output rather than the clean output. Because noise is added to the features, the model will capture the important information from the features as augmented information. This makes the model more generalizable and avoids overfitting.

Finally, we would like to comment on the choice of location for adding the ladder network structure to the model. We add the ladder network to the FPN rather than the backbone because the noise introduced in the ladder network can accumulate, as there are more noising layers. Adding the ladder network to the backbone, which has many layers, greatly reduces the model performance. However, adding the noise to the FPN only adds three noise terms per lateral connection.

4.2.2. Classifier frontend

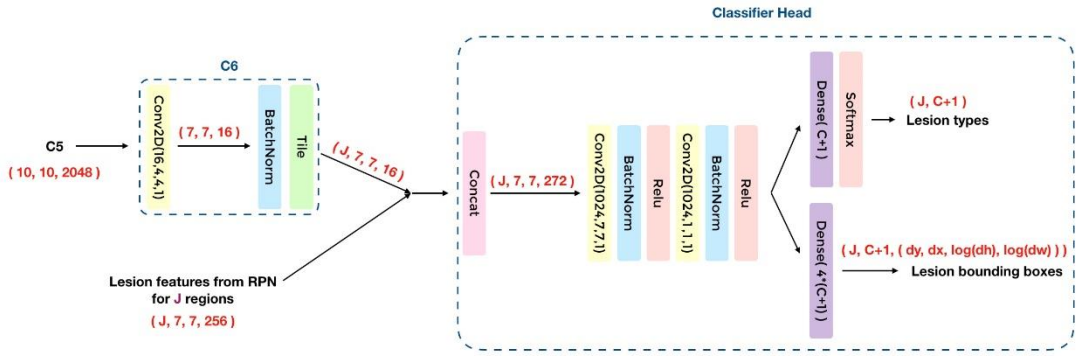


Figure 8: Illustration of the classifier frontend. Global features are applied with lesion features by concatenation. The classifier frontend separates into two branches for lesion type classification and bounding box prediction. Each lesion prediction of both sub-frontends has C+1 outputs.

The classifier frontend consists of two sets of convolutional and batch normalization layers with rectified linear unit (ReLU) activation. For classification, we used a dense layer with softmax normalization to obtain the probabilities for each lesion type. The dense layer has *C+1* neurons, where *C* is the number of lesion types with one additional class for non-lesions.

The bounding box prediction is designed to refine the proposal region, which is treated as a regression task using *4*(C+1)* outputs from a dense layer that predicts the position x and y coordinates, log(height), and log(width) for each class. Bounding box regression can be difficult for tasks that have objects that vary greatly in size. Therefore, the height and width of the bounding box are converted to the log scale, which usually can be easier to regress. This preprocessing method is considered a standard practice in object detection tasks (Girshick, Donahue et al. 2014).
We use the categorical cross-entropy loss as the classification cost as follows:

$$C_c = -\frac{1}{N}\sum_{n=1}^{N}\sum_{j=1}^{J} log\, P(\tilde{\boldsymbol{y}} = t_{n,j} \mid \boldsymbol{x}_{n,j}) \qquad (31)$$

Where J is the maximum number of region proposals in the image.

We used the smoothed L1 loss as the cost function for the bounding box prediction as shown below:

$$C_b = -\frac{1}{N}\sum_{n=1}^{N}\sum_{j=1}^{J} s(d_{n,j}) \quad (32)$$

where $d_{n,j}$ is L1 norm, which is the same as equation 20; $s(x)$ is the smoothed L1 loss, as shown in equation 21; and C_b is the sum of bounding box lost in all lesions.

4.2.3. Applying global features for lesion classification

Typically, object detection for natural images is performed by detecting each object independently regardless of the other objects in the same image. However, physicians usually take other lesions and other cues in addition to the lesion itself into account when deciding the lesion types. For example, if a lesion appears by itself without other lesions nearby, it is difficult to say that the lesion is malignant. However, if there are multiple lesions in the same region, they are usually malignant. Thus, we incorporate global features that summarize the information of the image to support the prediction of each individual lesion.

The construction of the classifier frontend is shown in Figure 8. The output from the core network (C5) is embedded into a lower-dimensional space using a convolutional layer and then tiled replicated (tile layer in TensorFlow) *J* times such that it can be concatenated with the features from each region proposal. The concatenated features are passed to the classifier frontend to classify the lesion type and adjust the bounding boxes.

4.2.4. Mask frontend

Figure 9: The illustration of the mask frontend for mask prediction in instance segmentation task.

Lesion segmentation is performed by the mask frontend. The mask frontend, given a proposed lesion region from the RPN, outputs C+1 foreground-background segmentation masks for each class. The mask that corresponds to the lesion classification is used as the segmentation output. Our architecture, shown in Figure 9, consists of sets of convolutional and batch normalization layers with ReLU activation. The output after passing sigmoid activation has a mask size of 14x14 pixels, *C+1* sets.

We use the binary cross-entropy loss as the cost function:

$$C_m = -\frac{1}{N}\sum_{n=1}^{N}\sum_{j=1}^{J} t_{n,j}\log s_{n,j} + (1 - t_{n,j})\log(1 - s_{n,j}) \quad (33)$$

where $s_{n,j}$ is the score for the foreground class after the sigmoid function and $\boldsymbol{y}_{n,j}$ is the ground-truth mask.

4.2.5. Unified loss

We have many components in our model, such as the LFPN (Figure 7), the RPN (Figure 3), the classifier and bounding box frontend (Figure 8), and the mask frontend (Figure 9). Each component has a different objective function from supervised and unsupervised loss. For this reason, the loss calculation must be weighted to avoid loss values of each term that are too different. The weight multiplier λ_k is the hyperparameter of each loss.

The combination of supervised loss C_s is the summation of the weight multiplier with their loss *k*, which is shown in the following equation:

$$C_s = \lambda_{rc}R_c + \lambda_{rb}R_b + \lambda_{cc}C_c + \lambda_{cb}C_b + \lambda_{cm}C_m \quad (33)$$

where R_c and R_b are the costs of the class and bounding box in the RPN and C_c, C_b, and C_m are the costs of the lesion class, lesion bounding box, and lesion mask in the classifier frontend and mask frontend; each λ_k is a weight multiplier of each loss *k*.

For unsupervised data, the cost function as shown below:

$$C_{us} = \lambda_{us}C_{LFPN,d} \quad (34)$$

where λ_{us} is the weight of the unsupervised loss term and $C_{LFPN,d}$ is the cost function in the LFPN, which is shown in equation 30.

We add an L2 regularization term to avoid overfitting and to make it more generalized. Therefore, the total cost is the sum of all cost functions with the L2 regularization term.

$$C_{total} = \lambda_{us}C_{LFPN,d} + C_s + \lambda_{L2}\sum_{k=0}^{K}\omega_k{}^2 \quad (35)$$

where λ_{us} is the weight of the unsupervised loss term, λ_{L2} is the weight of regularization, and *K* is the number of trainable layers.

Thus, our model is able to learn both supervised and unsupervised learning jointly, a form of semi-supervised learning.

4.3. Implementation details

We chose the original ResNet-50 as our core network due to the limited amount of labeled data. Moreover, ResNet-50 makes it easy to upsample in an FPN. Because the output images from the chest detection stage have different sizes, we scale and resize both the image and mask to match the GPU memory. During training, each minibatch contains both supervised and unsupervised data. We also tested a different setup that alternates between mini-batches of supervised and unsupervised data, and the same results were obtained. We used two sets of NVIDIA GeForce 1080 Ti for each batch size, equal to eight per GPU. The hyperparameters are detailed in Appendix B and C.

5. Experimental setup

In this section, we provide information about the dataset, data collection, and data labeling including the description of the lesion types. There are two main tasks in our pipeline: lesion instance segmentation and bone cancer metastasis prediction.

5.1 Dataset

Table 2: The amount of labeled and unlabeled data in training of lesion instance segmentation separated into training, validation, and testing data

	Number of data	Training data	Validation data	Testing data
Labeled data	1,088	741	231	116
Unlabeled data	18,560	14,786	3,774	0
Total data	19,648	15,527	4,005	116

Table 3: The total number of lesions per type.

Lesion types	Number of lesions
Malignant	3,500
Inflection/Inflammation	290
Degenerative change	805
Post-trauma	415
Total lesions	5,010

We included a total of 9,824 patients. The details of the patients' genders and ages are shown in Appendix A. The injection dose of 20 mCi/70 kg varied according to the patient's weight, and the uptake time was approximately 5 hours. The images are in DICOM format with 16-bit depth. For chest detection, we used 680 images of the whole body for training, 200 for validation, and 240 for testing. For lesion instance segmentation, the dataset contained 19,648 chest images separated into 1,088 labeled images and 18,560 unlabeled images.

The dataset was separated into training, validation, and testing data as detailed in Table 2. The physician mainly focuses on four types of lesions: malignant (or cancerous), inflection/inflammation, degenerative change (bone deterioration), and posttrauma (broken regions caused by accidents). The details about the number of each type of lesion are shown in Table 3.

5.2 Data collection and data labeling

Figure 10: The user interface of the bone labeling tool.

For the convenience of labeling, we created a labeling tool to reduce the effort of the physician. The program was developed using python language. In the application, the user can import an image into the program by clicking the load button. In case of user want to resume the previous job, the physician can click the resume button as shown in Figure 10.

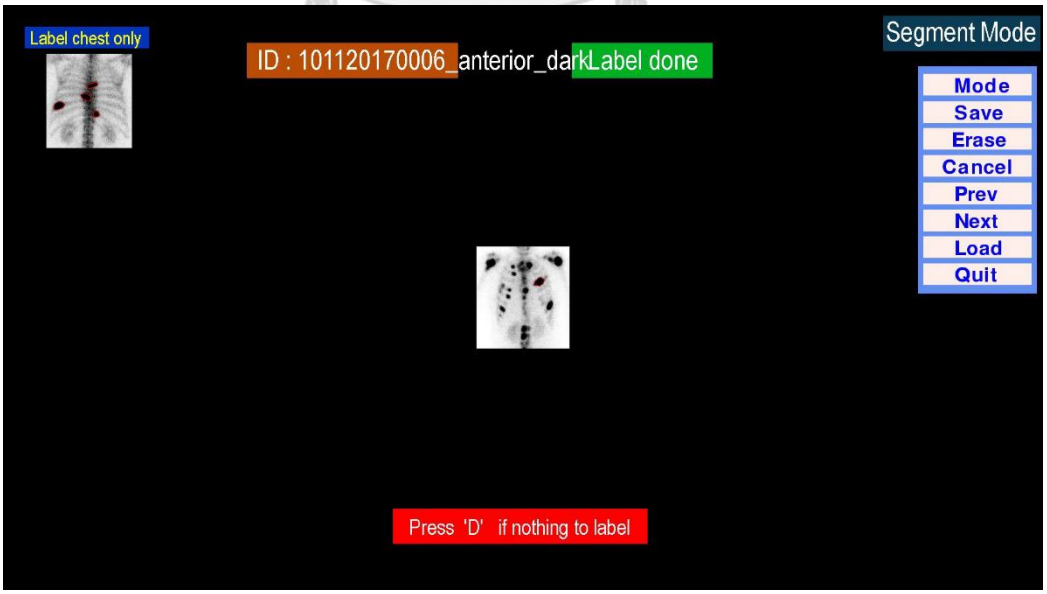


Figure 11: The main user interface of the bone labeling tool.

After clicking the “Load” button the system. The system will display images to prepare for data labeling from users as shown in Figure 11. For the main functions that the user can use include:

- “Mode” button: to change the labeling mode which are segment mode, zoom mode, and manual mode.
- “Save” button: To save the data after finished labeling.
- “Erase” button: To erase the region of labeling in the image.
- “Cancel” button: To cancel the selected region in the image.
- “Prev” button: To change the previous image.
- “Next” button: To change the next image.
- “Load” button: To import the image.
- “Quit” button: To exit the program.
- “D” button (in the keyboard): To specify whether there is no lesion in the image.
- “R” button (in the keyboard): To select the region for editing.
- “C” button (in the keyboard): To change the type of lesion.

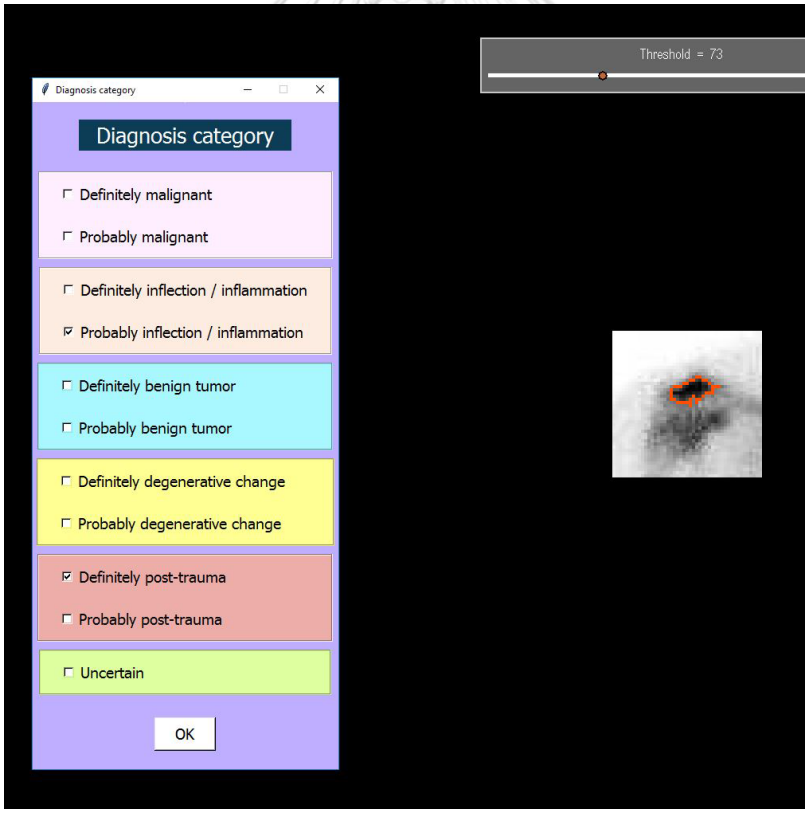


Figure 12: The user interface of creating and saving the type of lesion in the bone labeling tool.

Figure 12, represents an example of creating the ground-truth by using the zoom mode to select the area of interest. Recorded with the selection of the type of lesion in that area. The type of lesion can be divided into 5 main types which are
- Malignant tumor
- Benign tumor
- Degenerative change
- Inflection/Inflammation
- Post-traumatic cystic bone lesions

and can be divided into two confident levels which are:
- Definitely
- Probably

The data collection was approved by the Institutional Review Board (IRB) of the Faculty of Medicine, Chulalongkorn University. The labeling was performed by five nuclear medicine physicians with 31, 28, 21, 11, and 8 years, respectively, of diagnostic experience. Note that all labeled data were labeled by nuclear medicine physicians without the use of medical records. Thus, the type of lesion might not reflect the true lesion type. Thus, supervised learning was applied with a test that was not the gold standard and may not reflect the true metastasis value of the hotspots.

In the training process, we augment the data with an affine transformation. Normally, bone scintigraphy requires adjusting the light and contrast such that the physician can observe the hotspot before labeling. For this reason, we also augment the data by increasing and decreasing the light, contrast, and brightness for consistency with the physician's process.

6. Experimental results

In this section, we report the experimental results with a comparison of the performances of each model. Our experiments were divided into three subtasks: chest detection, lesion instance segmentation, and bone cancer metastasis prediction. We also conducted ablation studies that evaluated the impact of semi-supervised training in comparison with other semi-supervised methods.

6.1 Results of chest detection

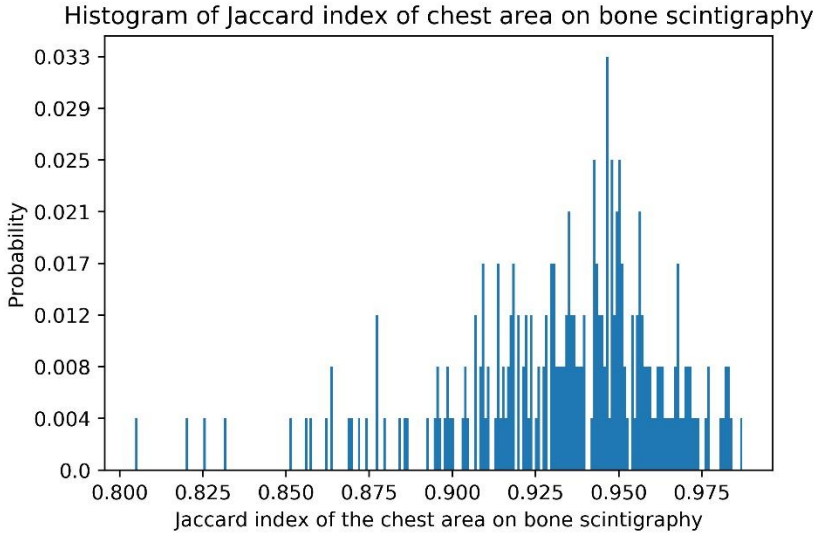


Figure 13: A histogram of the Jaccard index in chest detection for bone scintigraphy. The horizontal axis represents the Jaccard index of the chest area compare between the ground truth and prediction of the bounding box. The vertical axis represents the frequency of each Jaccard index in each bin of the histogram.

Anterior (frontside) and posterior (backside) images are available for each patient. Because detecting the chest area in the whole image from bone scintigraphy is a simple task, the model provides accurate results with min, mean, and max Jaccard indexes of 0.804, 0.933, and 0.987, respectively. From Figure 13, the histogram shows that SSD provides excellent chest detection results with a Jaccard index of at least on every testing data. Examples of chest detection results are shown in Figure 14.

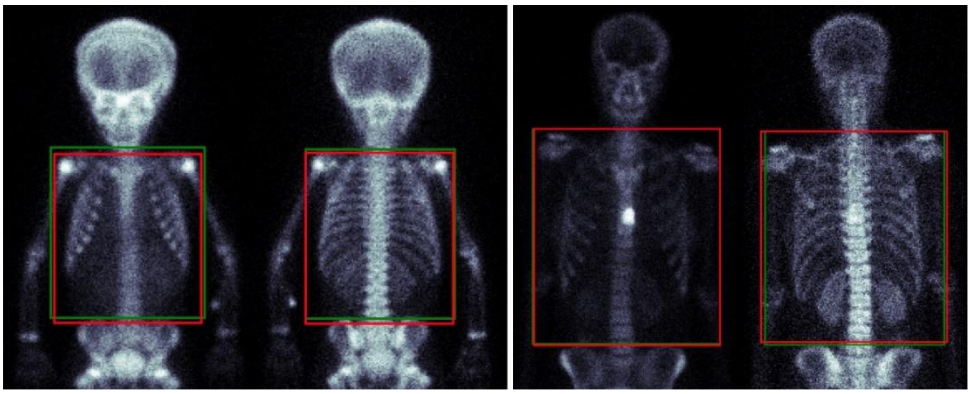


Figure 14: Some examples from the chest detection model. The first bone scintigram (the leftmost side) is an anterior view (front view), and the second bone scintigram is a posterior view (back view) of a pediatric patient. The third skeleton is an anterior view, and the last skeleton (the rightmost side) is a posterior view of an adult patient. Ground-truth boxes are indicated in green, while the outputs of the SSD model are indicated in red. The Jaccard indices are 0.895, 0.943, 0.987, and 0.914 from left to right.

6.2. Results of the lesion instance segmentation task

In this section, we provide the results of lesion instance segmentation. The chest images from bone scintigraphy, which are the output results from chest detection, were used as the data in this task. Data cleaning and augmentation are applied before performing the experiments. We evaluated our model in lesion instance segmentation on four lesion types and compared the results with the baseline model (Mask R-CNN). Examples of the results are shown in Figure 15 (hand-picked examples) and Figure 16 (random examples). We also studied the impact of how each technique of the model affects the overall performance. We conducted the experiments with the techniques separately and combined, as shown in Table 4. Moreover, we applied self-training with the baseline and our model to compare the effect of each technique.

Table 4: Comparison between each model and technique for the lesion instance segmentation task. The global features in this table are the output features from layer C6 in Figure 6.

Approach	*Mean Precison*	Mean sensitivity	Mean f1-score
Resnet50 + FPN (Mask R-CNN)	0.827	0.811	0.816
Resnet50 + FPN w/ global features	0.829	0.826	0.824
Resnet50 + LFPN w/o global features	0.838	0.839	0.835
Resnet50 + LFPN w global features (MaligNet)	**0.852**	**0.856**	**0.848**
Resnet50 + FPN + self-training	0.849	0.843	0.840
Resnet50 + LFPN w/ global features + self-training	**0.867**	**0.844**	**0.851**

We evaluated our model in lesion instance segmentation on four lesion types and compared the results with the baseline model (Mask R-CNN). Examples of the results are shown in Figure 15 and Figure 16. We also studied the impact of how each technique of the model affects the overall performance. We conducted the experiments with the techniques separately and combined, as shown in Table 4.

Table 5: Comparison between MaligNet and baseline for lesion localization in the lesion segmentation task.

Model	Precision	Sensitivity	F1-score
Mask R-CNN	0.691	0.736	0.713
MaligNet	0.678	0.781	0.726

Table 6: Comparison between MaligNet and baseline for lesion classification in the lesion segmentation task.

Lesion types	Model	Accuracy	Precision	Sensitivity	Specificity	F1-score
Malignant	MaligNet	**0.886**	**0.912**	**0.941**	**0.710**	**0.926**
	Mask R-CNN	0.839	0.864	0.937	0.519	0.899
Inflecton/ Inflammation	MaligNet	0.950	0.432	0.667	**0.962**	0.525
	Mask R-CNN	**0.953**	**0.447**	**0.739**	**0.962**	**0.557**
Degenerative change	MaligNet	**0.905**	0.662	**0.584**	0.954	**0.621**
	Mask R-CNN	0.891	**0.667**	0.342	**0.974**	0.452
Post-trauma	MaligNet	**0.952**	**0.737**	**0.378**	**0.991**	**0.500**
	Mask R-CNN	0.936	0.476	0.278	0.980	0.351

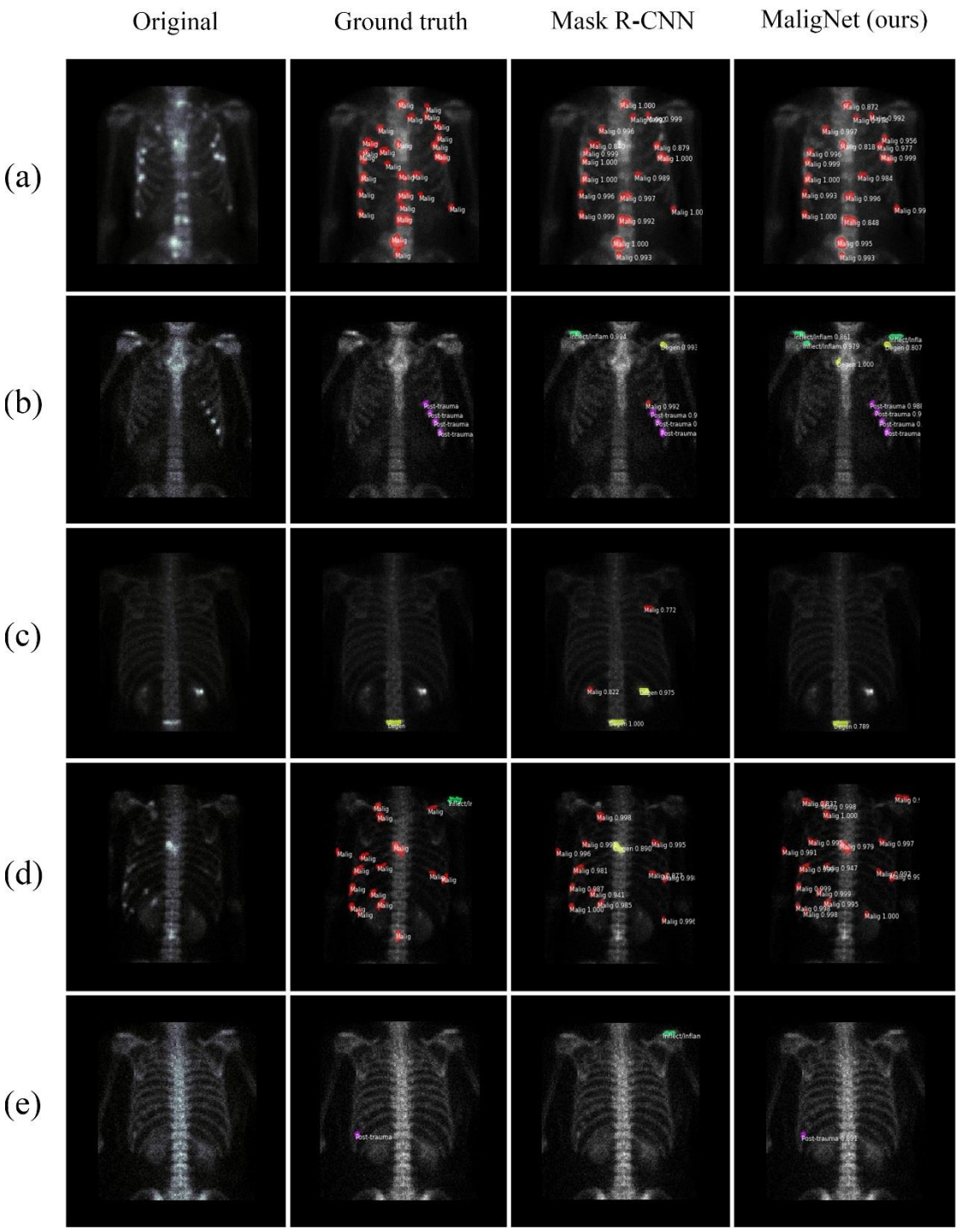


Figure 15: Hand-picked examples of the comparison results: the leftmost image is the original bone scintigram, the second image is the ground-truth image, the third image is the result of Mask R-CNN, and the rightmost image is the result of MaligNet (ours). Each row refers to a different subject. Each column refers to different image sources.

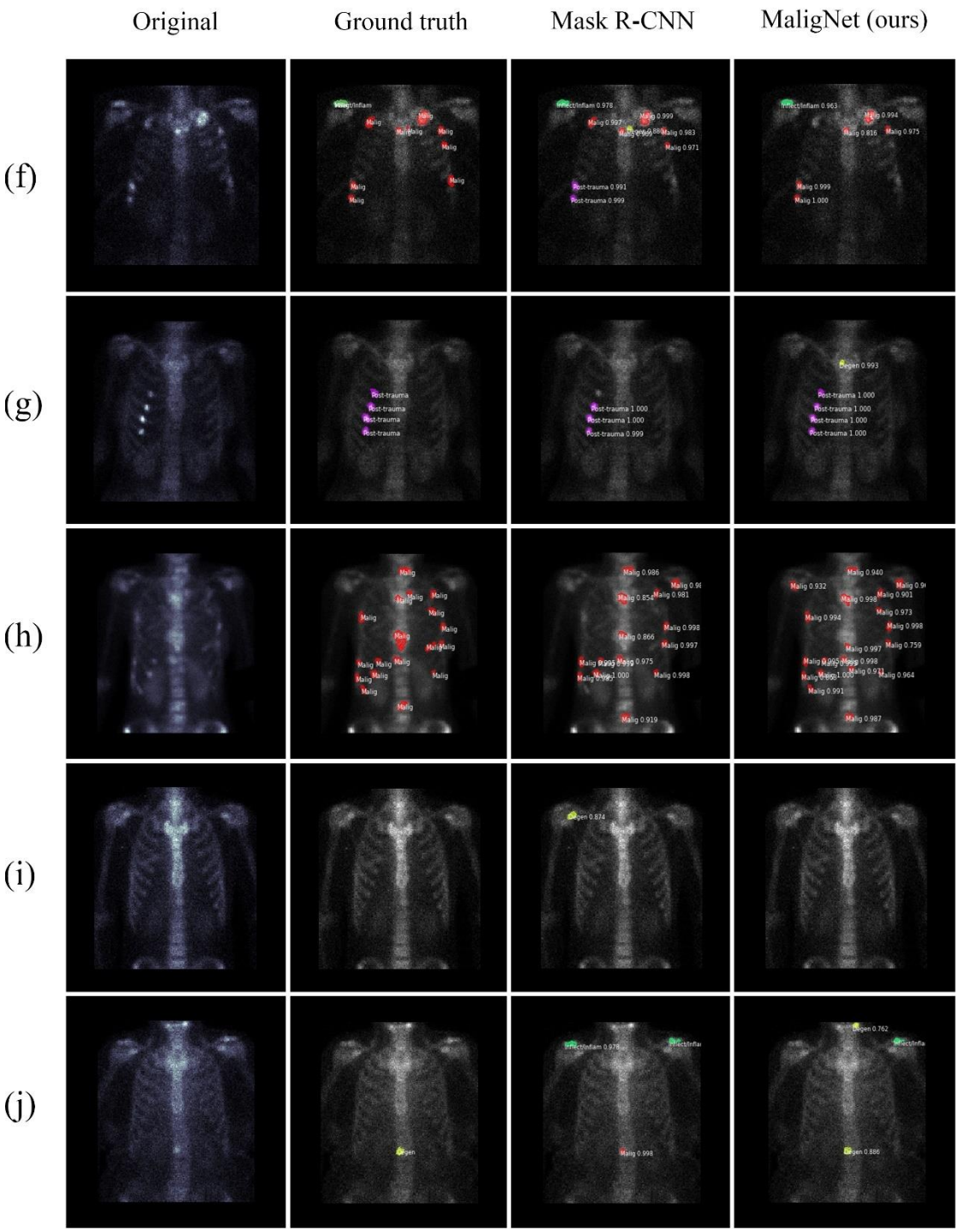


Figure 16: Random examples of the comparison result: the leftmost image is the original bone scintigram, the second image is the ground-truth image, the third image is the result of Mask R-CNN, and the rightmost image is the result of MaligNet (ours). Each row refers to a different subject. Each column refers to different image sources.

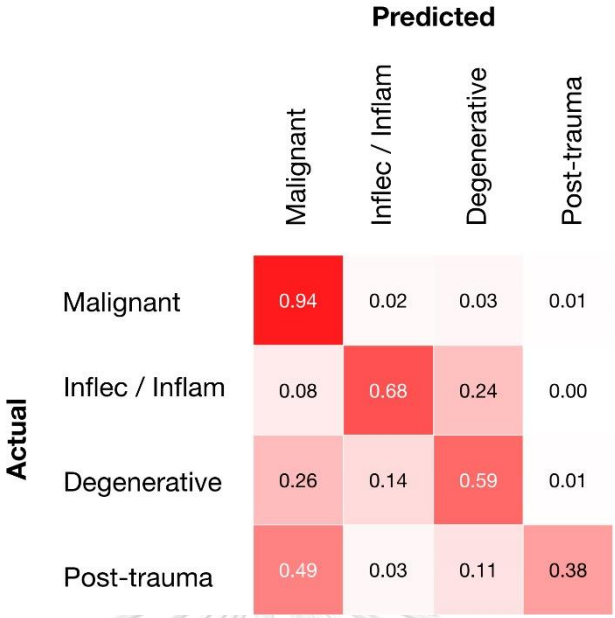


Figure 17: The normalized confusion matrix of the lesion classification task using MaligNet without self-training. The rows represent the true labels (ground truth), and the columns represent the predicted label.

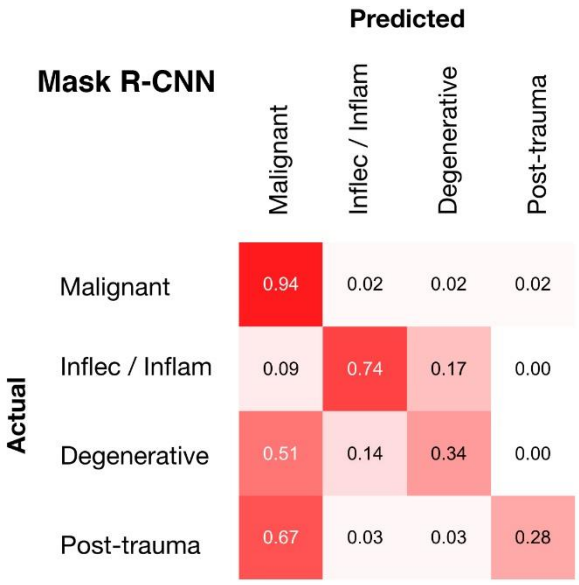


Figure 18: The normalized confusion matrix of the lesion classification task using Mask R-CNN. The rows represent the true labels (ground truth), and the columns represent the predicted label.

Using global features allows the model to use high-level features and semantically strong features to make prediction decisions, which increases the accuracy of lesion classification. Moreover, applying the ladder network in the FPN makes the model capable of learning the representation of the images in unsupervised learning, thus improving the model in every comparable configuration. Utilizing LFPN for semi-supervised training over the standard Mask R-CNN, MaligNet can take advantage of the unlabeled data (14,786 images), increasing the performance of the model significantly and reaching an f1-score of 0.835. Furthermore, combining global features allows the f1-score of MaligNet to be improved even further to 0.848. We also show the results of lesion classification using MaligNet compared with the baseline model, as shown in the confusion matrix in Figure 17 and Figure 18. Furthermore, we compare the results of both models in lesion localization and lesion classification on each lesion class as shown in Table 5 and Table 6 respectively.

The results show that MaligNet tends to predict malignancy extremely well. Other classes are rarer in the training data, making it less accurate. Moreover, post-trauma is similar to malignancy, and it can be difficult to distinguish this class from malignant lesions, resulting in lower accuracy.

6.3. Results of the bone cancer metastasis prediction task.

Table 7: The results of MaligNet on bone cancer metastases prediction.

Model	Backbone	Accuracy	Precision	Recall (Sensitivity)	Specificity	F1-score
Mask R-CNN	Resnet50	0.707	**0.941**	0.608	**0.919**	0.739
MaligNet	Resnet50	**0.741**	0.863	**0.657**	0.857	**0.746**

Bone cancer metastasis prediction from lesion instance segmentation is more difficult than direct classification. Rather than distinguishing between metastases and non-metastases, the model must locate the position of a malignant lesion. The results in Table 7 show that MaligNet has higher accuracy, sensitivity, and f1-score than the baseline model. Although our model has lower precision and specificity, for our application, sensitivity is preferred over other metrics.

Our model slightly takes longer inference time (0.76 ms for Mask R-CNN vs 0.87 ms for Malignet). Even though the f1-score only increases slightly, the sensitivity, which is the main metric for screening, improves by an absolute 5% without requiring more labeled data.

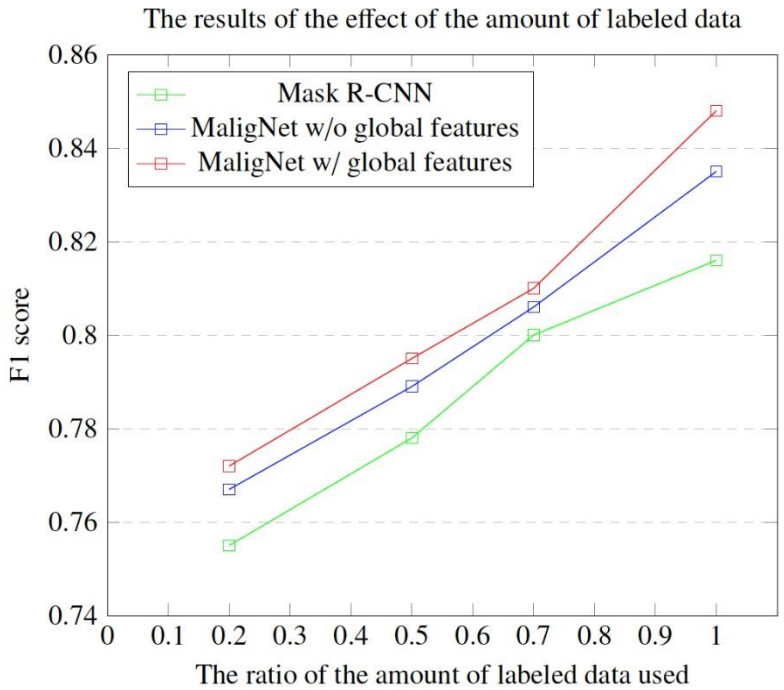
Note that we can also apply model optimization and compression techniques, such as network pruning (Han, Pool et al. 2015), weight quantization (Hubara, Courbariaux et al. 2017), binarized neural networks (Courbariaux, Hubara et al. 2016), and deep compression (Han, Mao et al. 2015), to reduce the inference time; however, model accuracy is the main concern of this work.

6.4. The impact of data

6.4.1 Effect of the amount of labeled data

In this experiment, we varied the amount of labeled training data while keeping the amount of unlabeled data fixed and measuring the f1-score. The results are shown in Figure 19. Using unsupervised data, MaligNet w/o global features improve every time the amount of training data is increased, thus improving the performance over the Mask R-CNN baseline model by an average of 1.51%. Adding global features improves the performance even further, reaching a relative f1-score average improvement of 2.40%. At the same f1-score level, our proposed method reduces the amount of labeled data required by an average of 20.11%. The result can be used as an anecdotal reference when deciding the trade-off between spending more time to label the data and making use of semi-supervised methods.

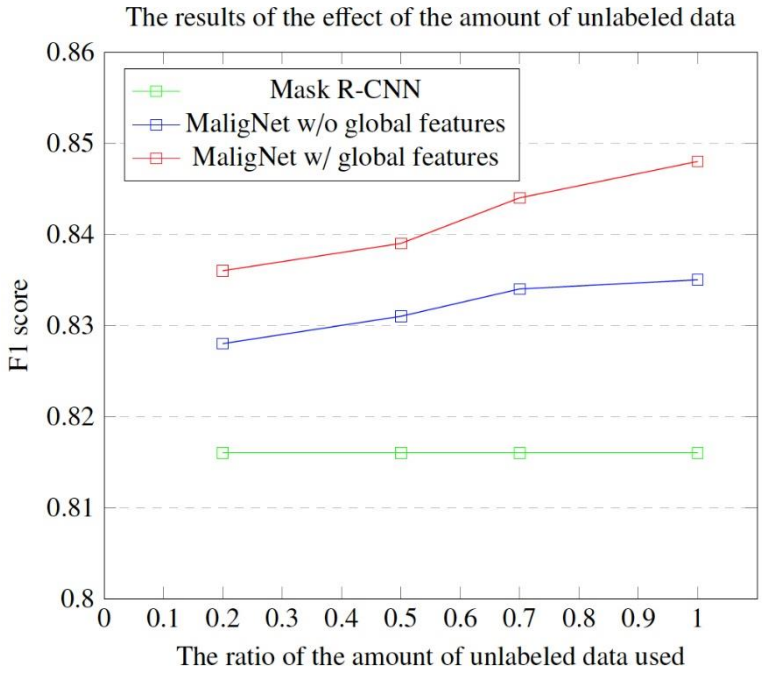
Figure 19: The effect of the amount of labeled data in lesion instance segmentation measured by the f1-score. We perform the experiments while increasing the amount of labeled data at various ratios. MaligNet can also use unlabeled data, whereas Mask R-CNN is fully supervised.



6.4.2 Effect of the amount of unlabeled data

We also studied the effect of varying the amount of unlabeled data. As shown in Figure 20, the performance increases as we include more unlabeled training data. However, at higher amounts, the gain from adding more data decreases. This is expected because the unlabeled data are used to learn better representations. The model captures enough variation from the unsupervised data, and adding more unsupervised data should have little to no effect.

Figure 20: The effect of the amount of unlabeled data in lesion instance segmentation measured by the f1-score. We conduct the experiment by increasing the amount of unlabeled data while keeping the amount of labeled data fixed. Because Mask R-CNN cannot use unlabeled data, the performance remains constant.



6.5. Comparison with the self-training method

Table 8: The results of the self-training approach with a different confidence threshold.

Confidence threshold	Filter out	No. of images	No. of lesions	*Mean precision*	Mean sensitivity	Mean f1-score
All	-	18,560	52,756	0.815	0.826	0.818
>0.8	Lesions level	18.560	47,140	0.834	0.819	0.822
	Images level	15,423	33,572	0.837	0.836	0.831
>0.85	Lesions level	18,560	42,998	0.827	0.837	0.828
	Images level	12,761	21,964	0.849	0.843	**0.840**
>0.90	Lesions level	18,560	38,018	0.829	0.825	0.823
	Images level	10,443	13,871	0.836	0.833	0.830

One popular semi-supervised approach is *self-training*. Self-training produces virtual labels for unlabeled data by treating the model predictions as the ground-truth label. The original labeled data are then combined with the unlabeled data (with labels produced by the model) to train a better model.

A confidence threshold value can be used to filter unlabeled data that the model is not certain about. We can treat the softmax output probability from the model as the confidence level and only use data that are above a certain confidence level. We set the minimum confidence threshold for Mask R-CNN postprocessing (which removes clutter and merges overlapping regions) to 0.7 which results in confidence values ranging only from 0.7 to 1.0. We tried different confidence levels with 0.05 increments and report only values that show a local maximum in Table 11.

We filtered out the data in two ways: lesion level and image level. For the lesion level, we filter lesions that are lower than the threshold value. For the image level, we filter out any images that have at least one lesion with a score lower than the threshold. This filters out approximately 30% of the images. The results are shown in Table 8. The best results are from a threshold value of 0.85 and image-level filtering. This improves the Mask R-CNN baseline results from 0.816 to 0.840.

6.6 Results of model visualization

In this section, we conduct the mask results of the experiment to explain how do model classifies each lesion type. Grad-CAM (Selvaraju, Cogswell et al. 2017) is one of the techniques to explain the model results that use the gradients of the target concept to make visual explanations in the classifier model. Since Malignet is the instance segmentation model so that we can't directly apply Grad-CAM to visualize our model.

For our case, we visualize the results through the mask frontend. Our mask frontend is the binary classification in all classes which can visualize the difference between each class compare with the ground-truth. To do this, we can't explain which part of the image that the model looks to classify. But, we will show whether how the model chooses to mask in the same lesion of different classes. The pattern of mask prediction of each class is the representation to visualize whether why each class answers differently and what is the difference in the lesion mask.

We show the results of the mask prediction and the difference between each mask class in Figure 21-36.

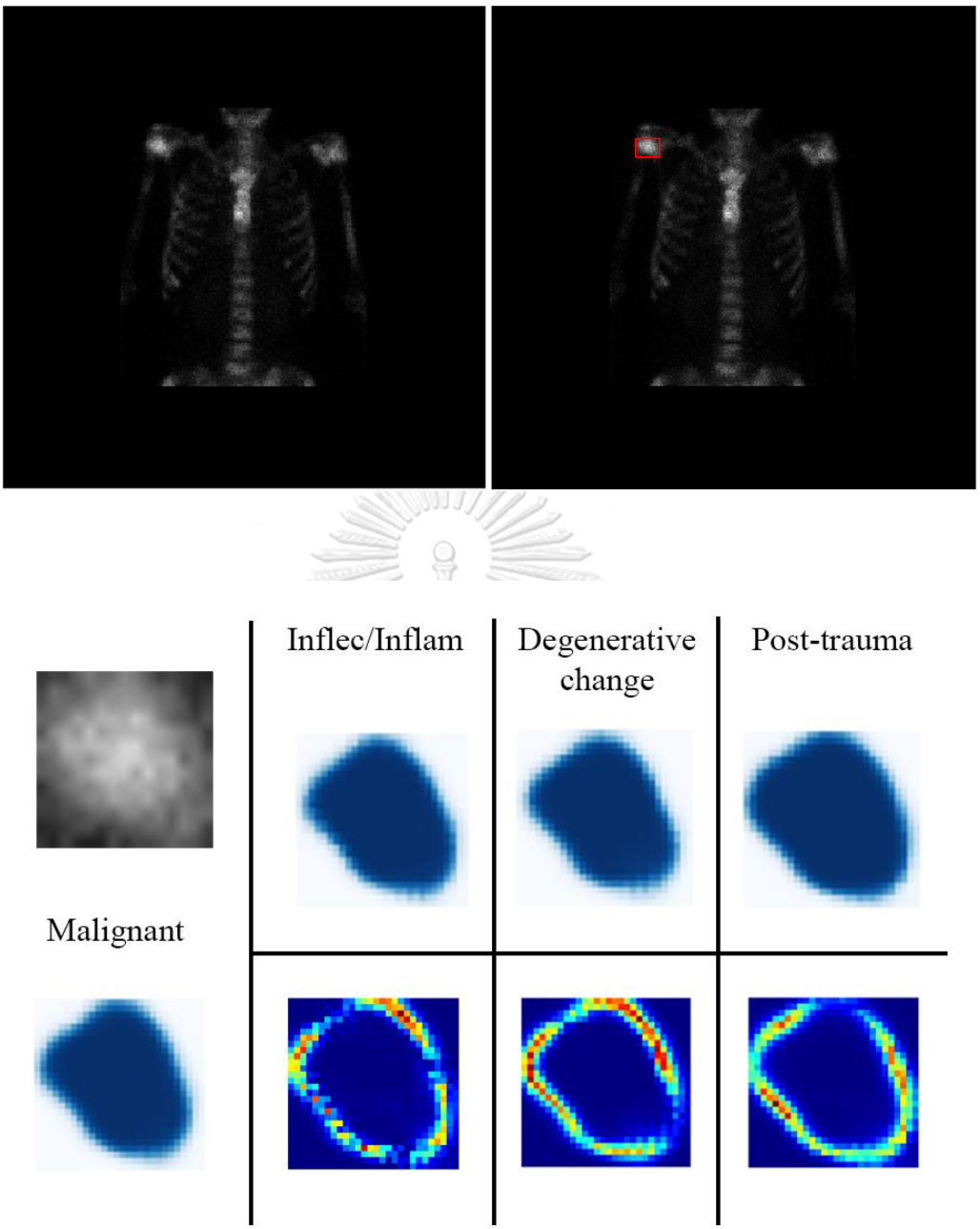


Figure 21: The visualization of mask prediction of each class. The most left side is the mask prediction of malignant. The difference between the ground truth class and other classes are shown in the bottom.

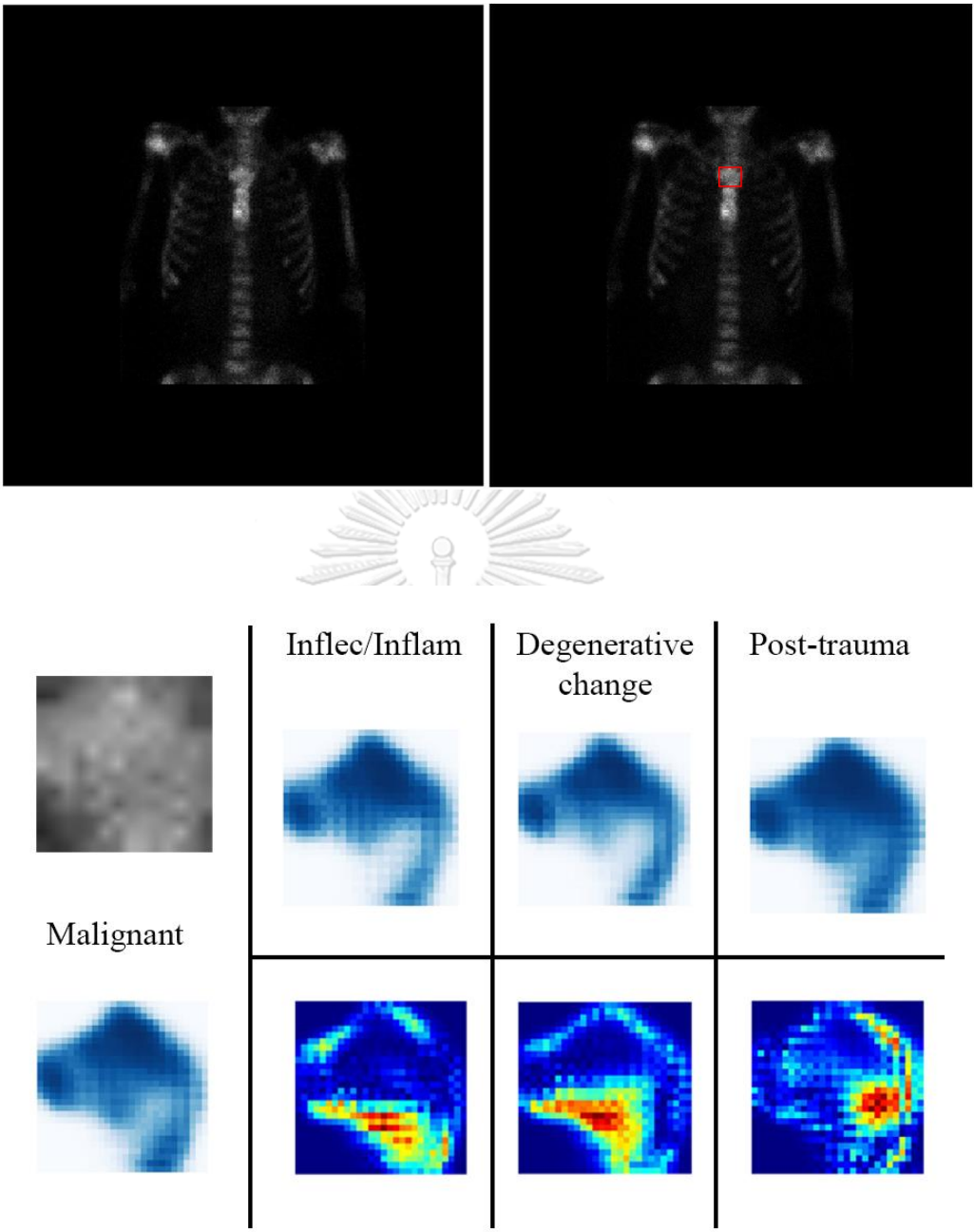


Figure 22: The visualization of mask prediction of each class. The most left side is the mask prediction of malignant. The difference between the ground truth class and other classes are shown in the bottom.

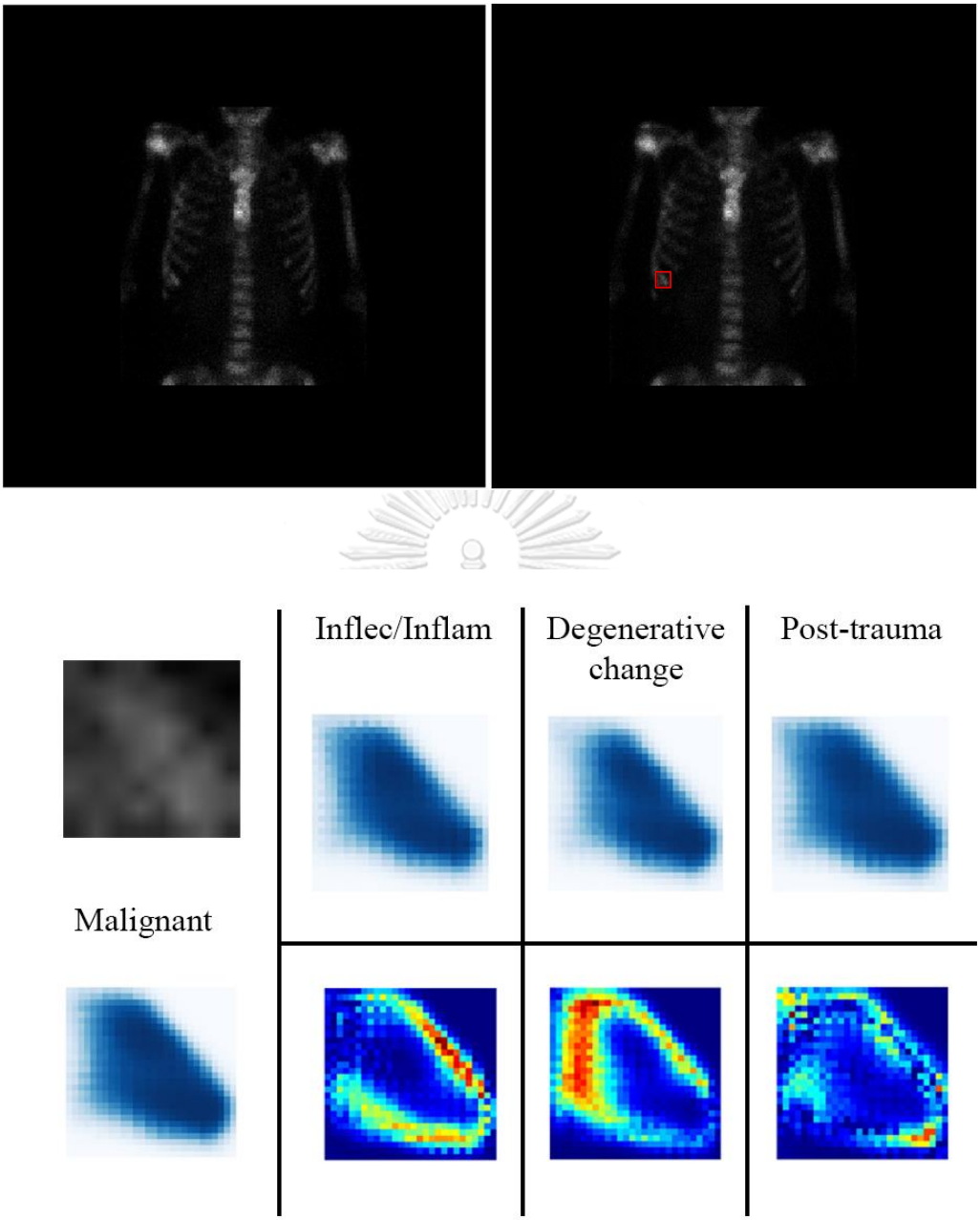


Figure 23: The visualization of mask prediction of each class. The most left side is the mask prediction of malignant. The difference between the ground truth class and other classes are shown in the bottom.

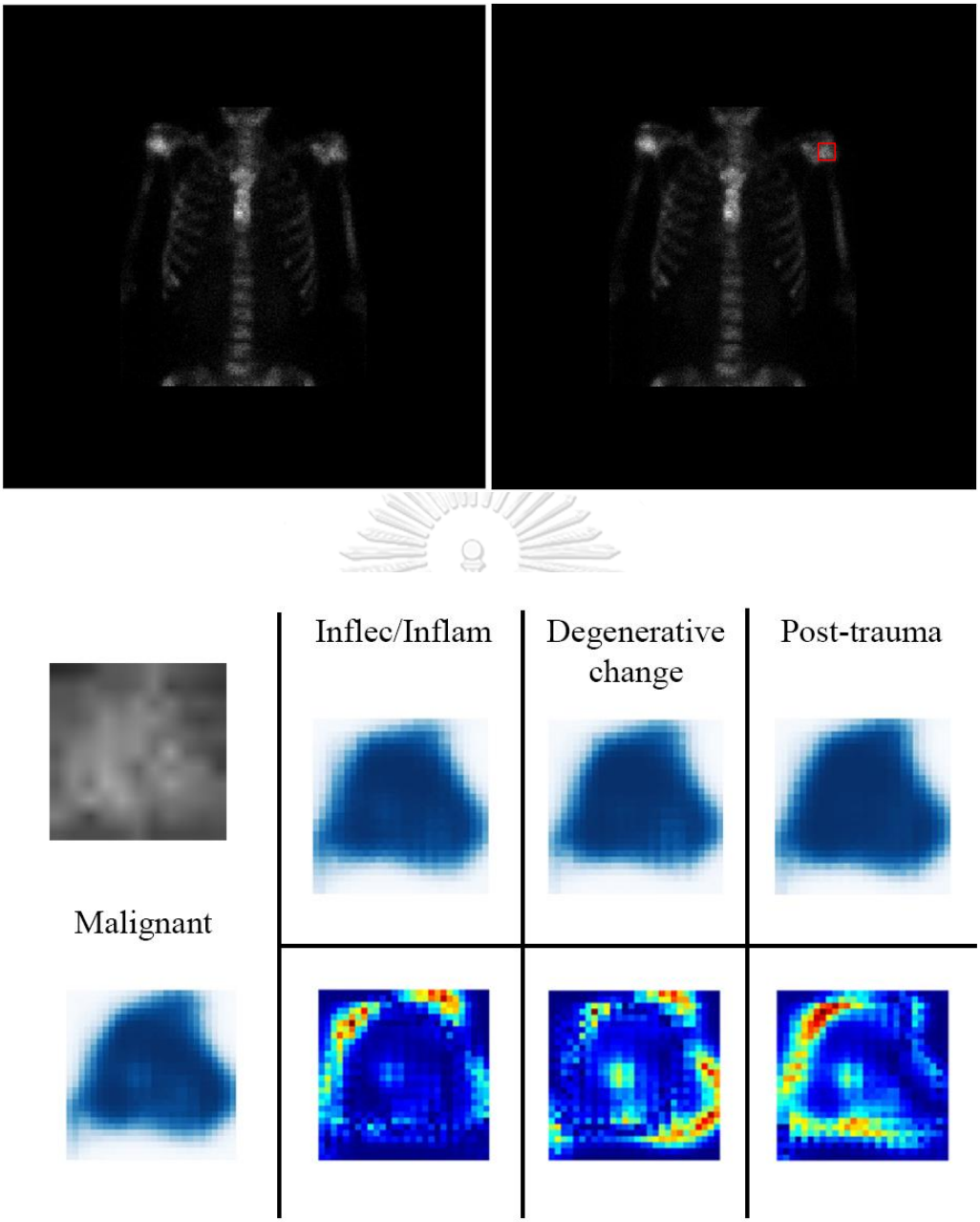


Figure 24: The visualization of mask prediction of each class. The most left side is the mask prediction of malignant. The difference between the ground truth class and other classes are shown in the bottom.

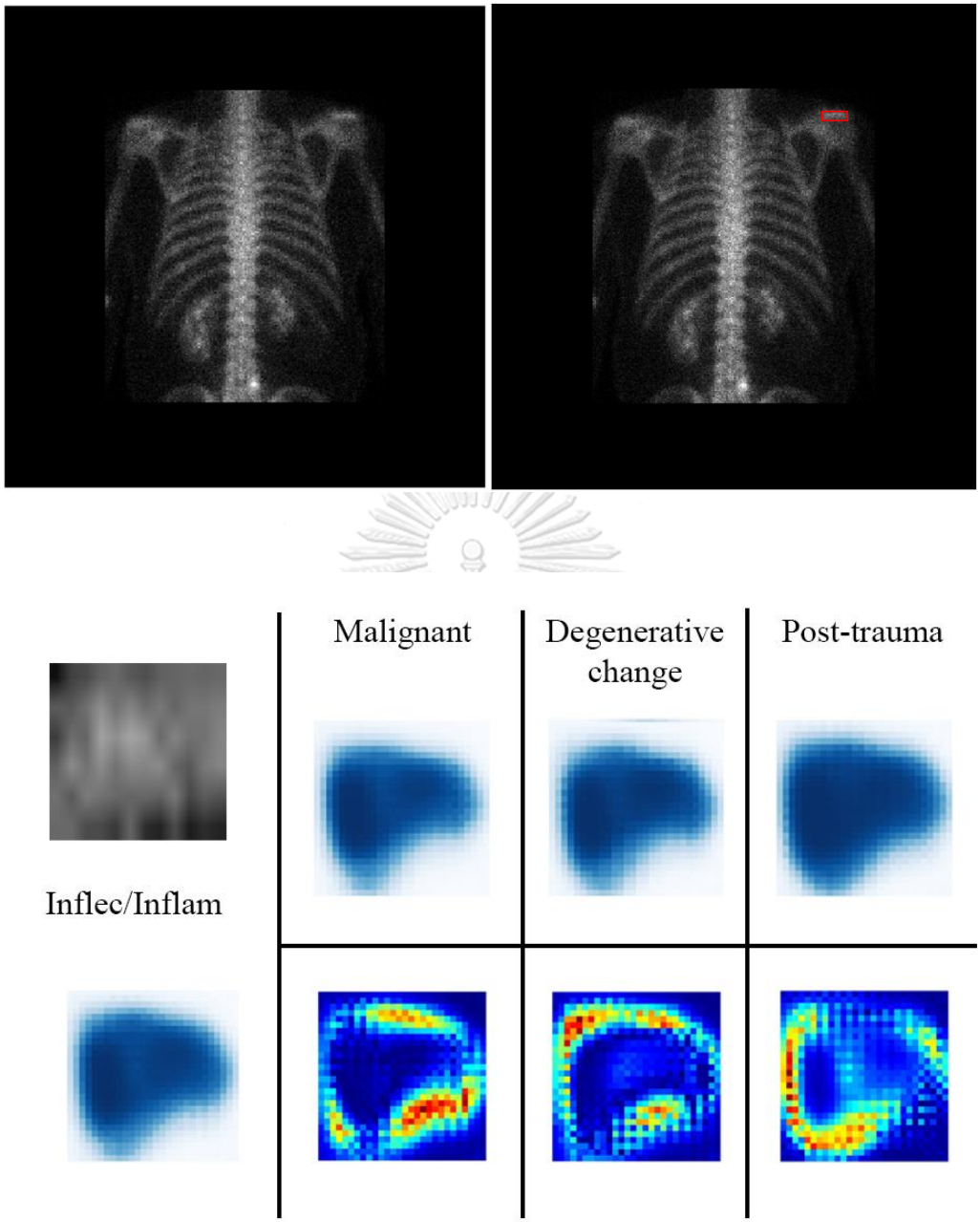


Figure 25: The visualization of mask prediction of each class. The most left side is the mask prediction of inflection/inflammation. The difference between the ground truth class and other classes are shown in the bottom.

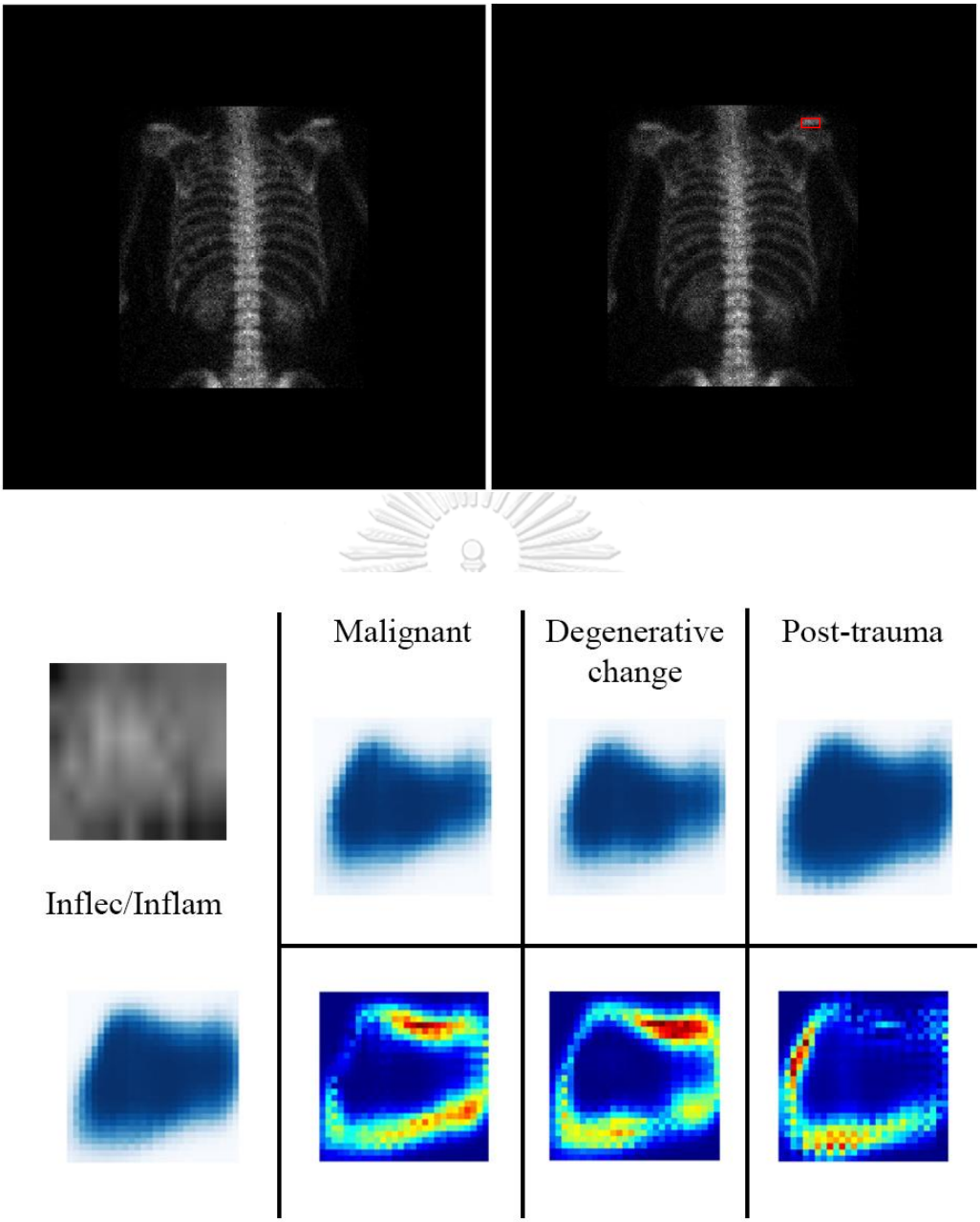


Figure 26: The visualization of mask prediction of each class. The most left side is the mask prediction of inflection/inflammation. The difference between the ground truth class and other classes are shown in the bottom.

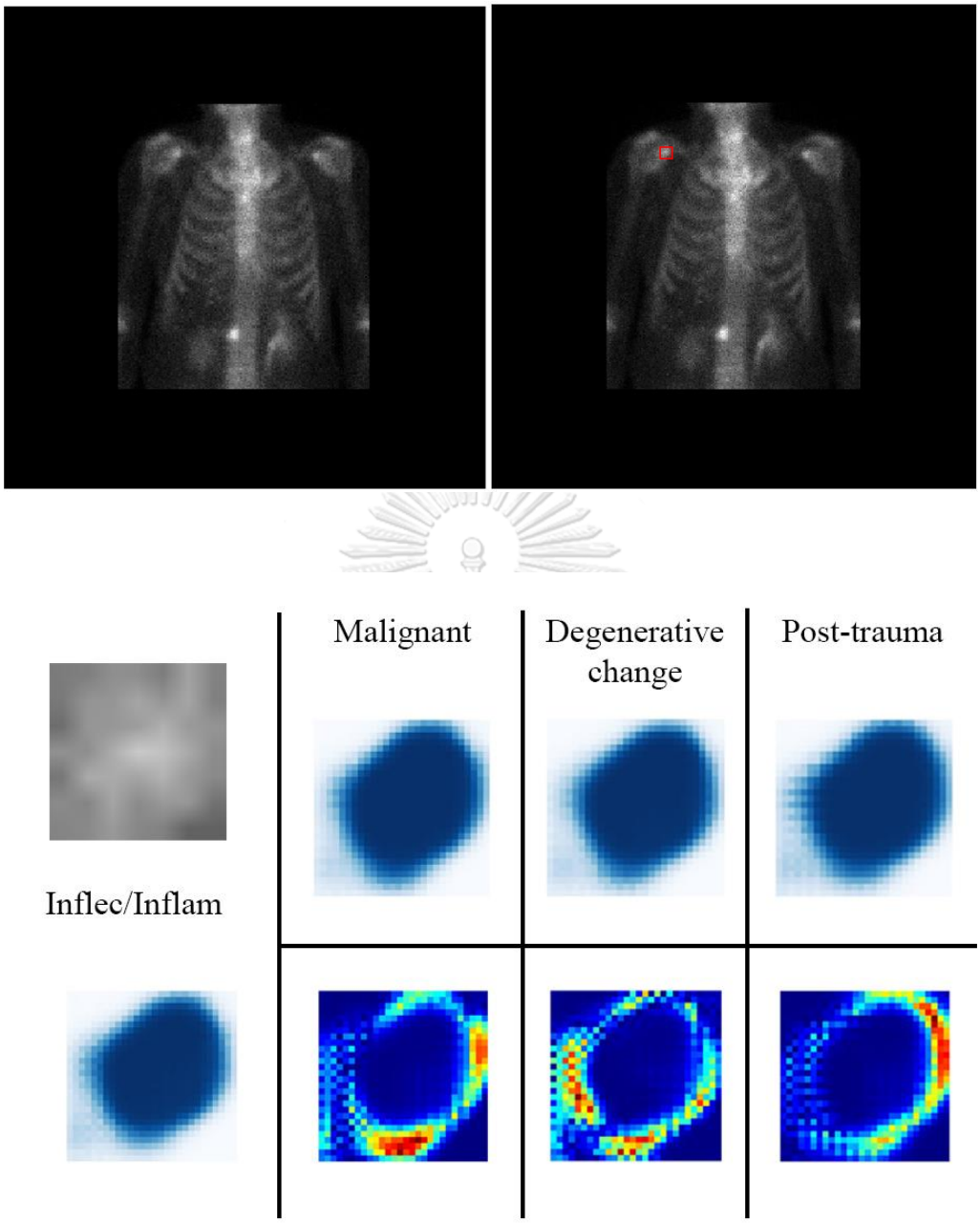


Figure 27: The visualization of mask prediction of each class. The most left side is the mask prediction of inflection/inflammation. The difference between the ground truth class and other classes are shown in the bottom.

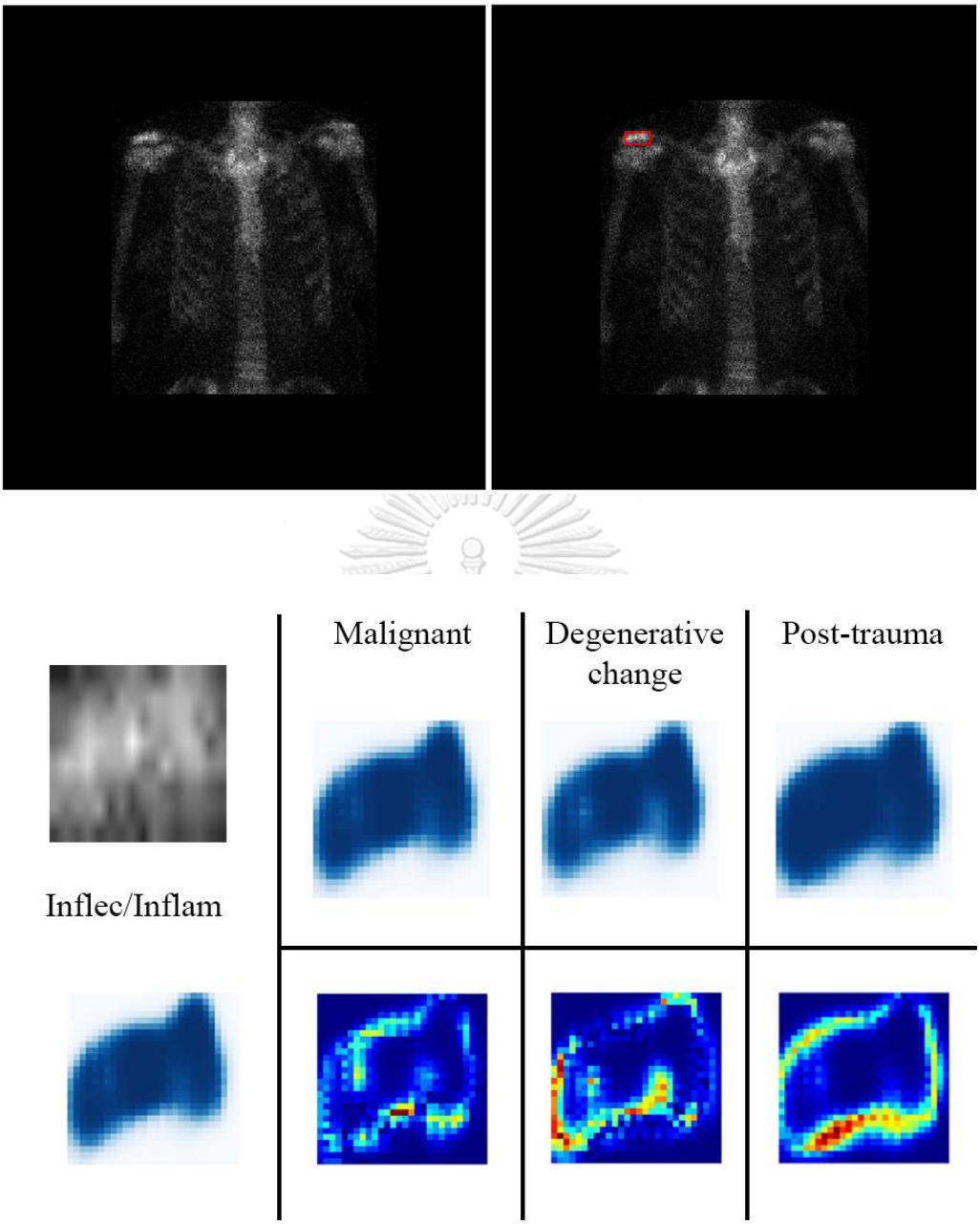


Figure 28: The visualization of mask prediction of each class. The most left side is the mask prediction of inflection/inflammation. The difference between the ground truth class and other classes are shown in the bottom.

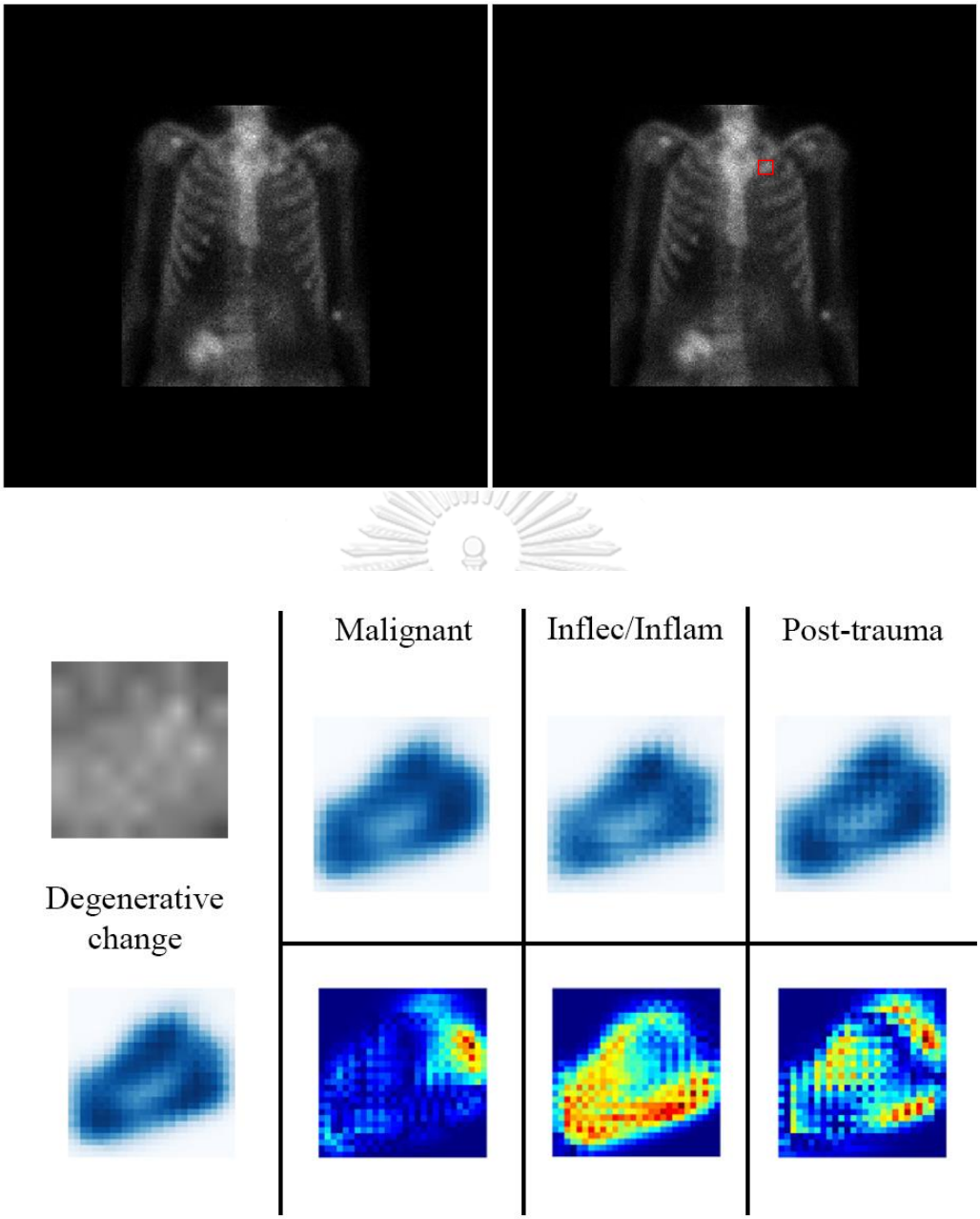


Figure 29: The visualization of mask prediction of each class. The most left side is the mask prediction of degenerative change. The difference between the ground truth class and other classes are shown in the bottom.

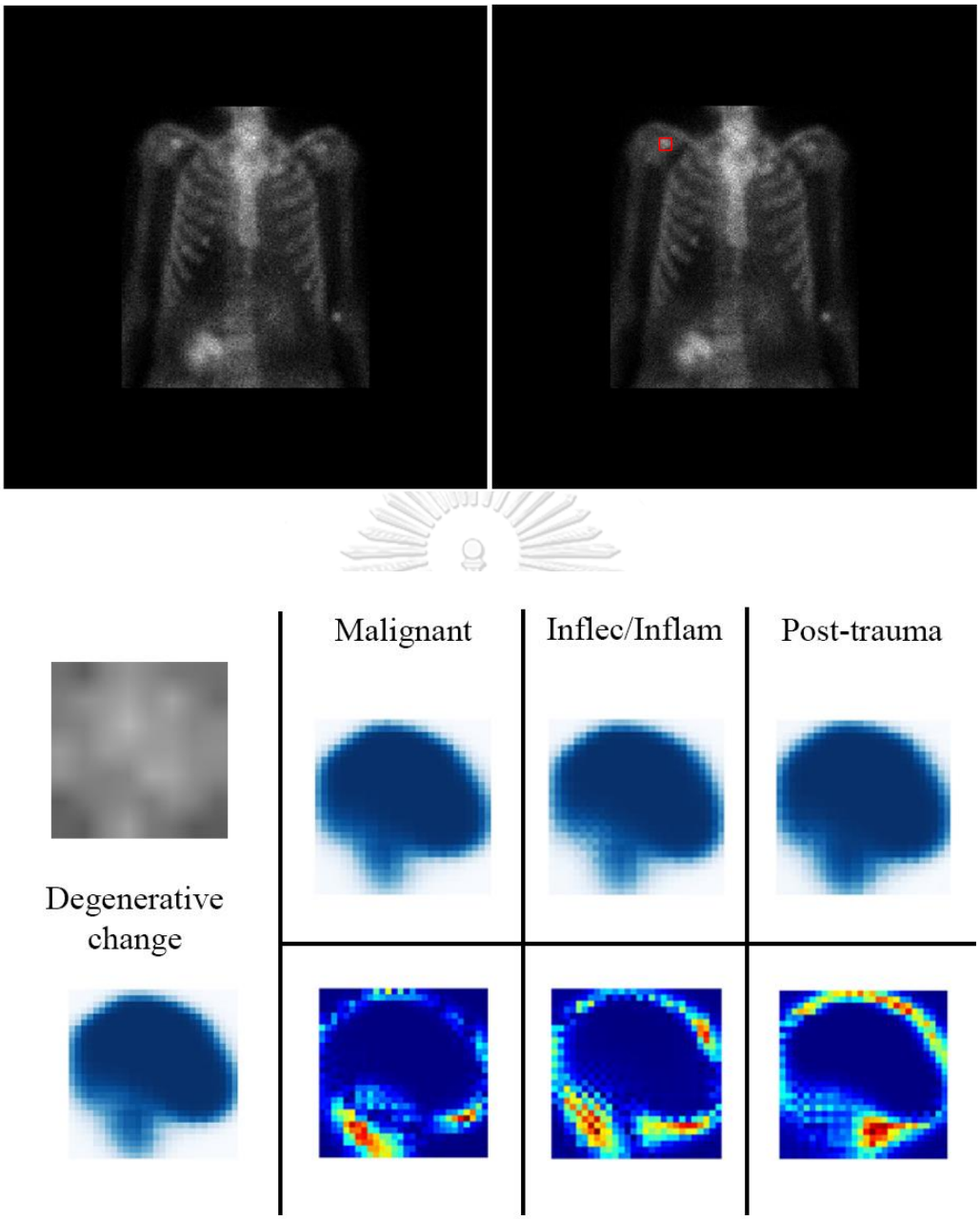


Figure 30: The visualization of mask prediction of each class. The most left side is the mask prediction of degenerative change. The difference between the ground truth class and other classes are shown in the bottom.

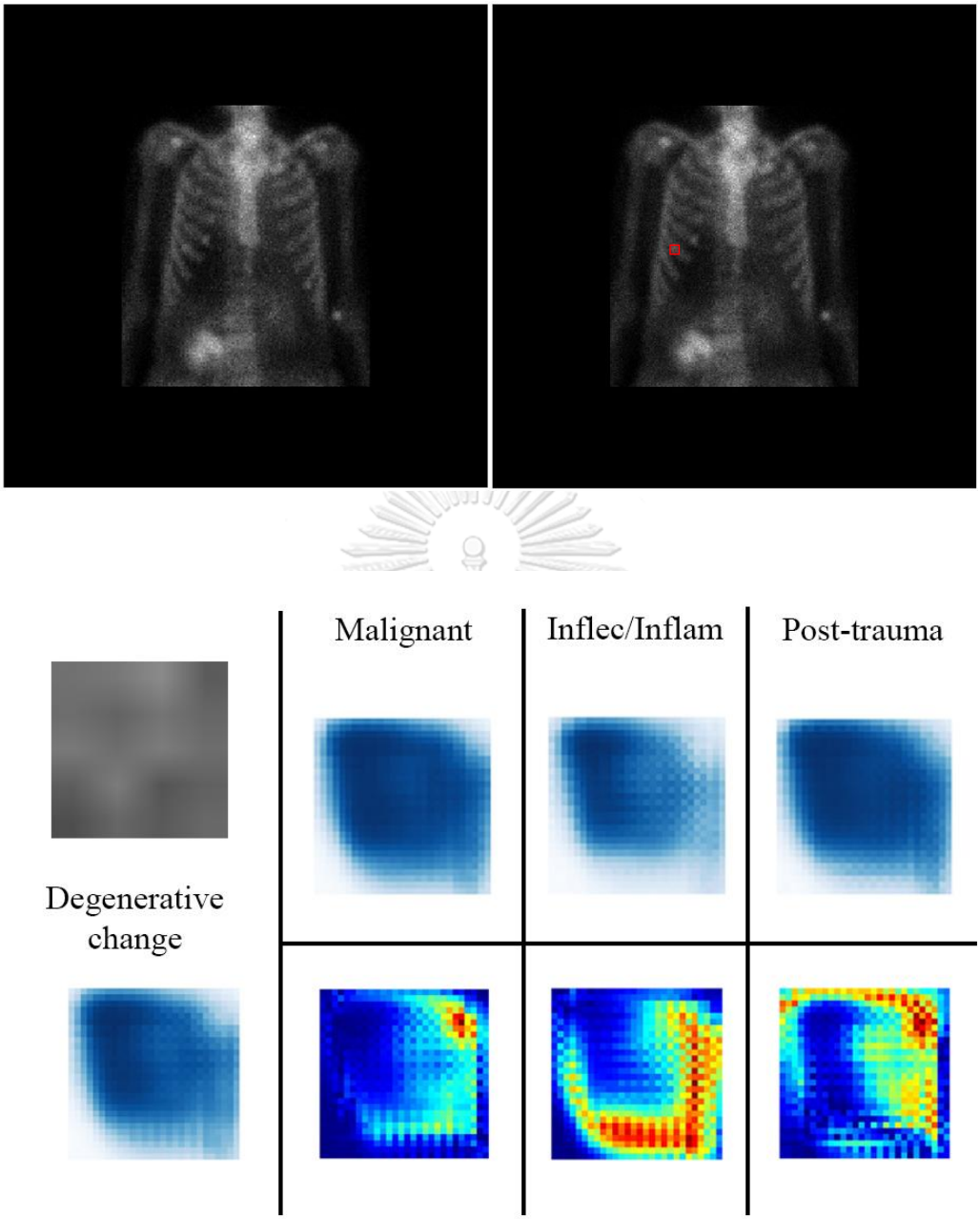


Figure 31: The visualization of mask prediction of each class. The most left side is the mask prediction of degenerative change. The difference between the ground truth class and other classes are shown in the bottom.

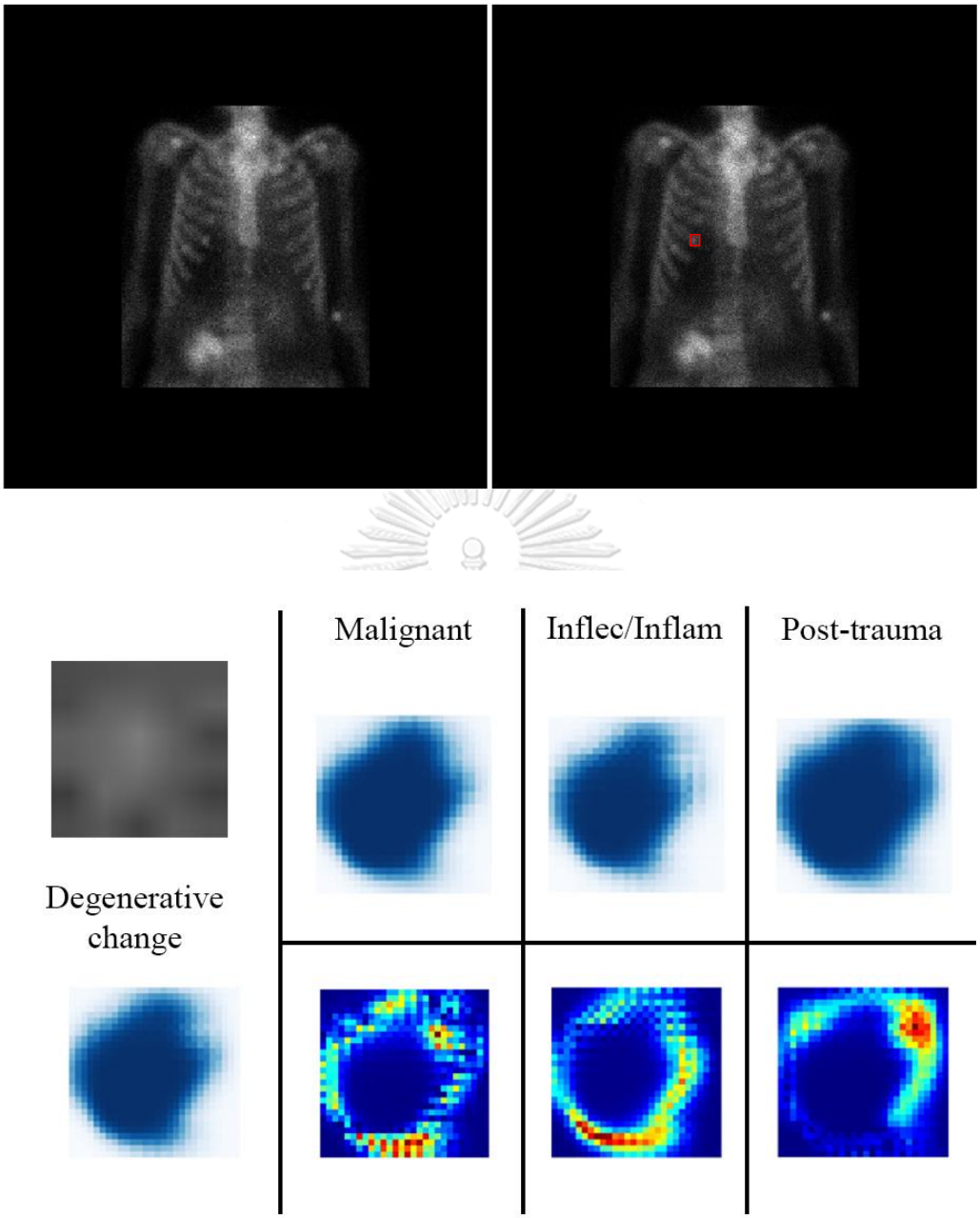


Figure 32: The visualization of mask prediction of each class. The most left side is the mask prediction of degenerative change. The difference between the ground truth class and other classes are shown in the bottom.

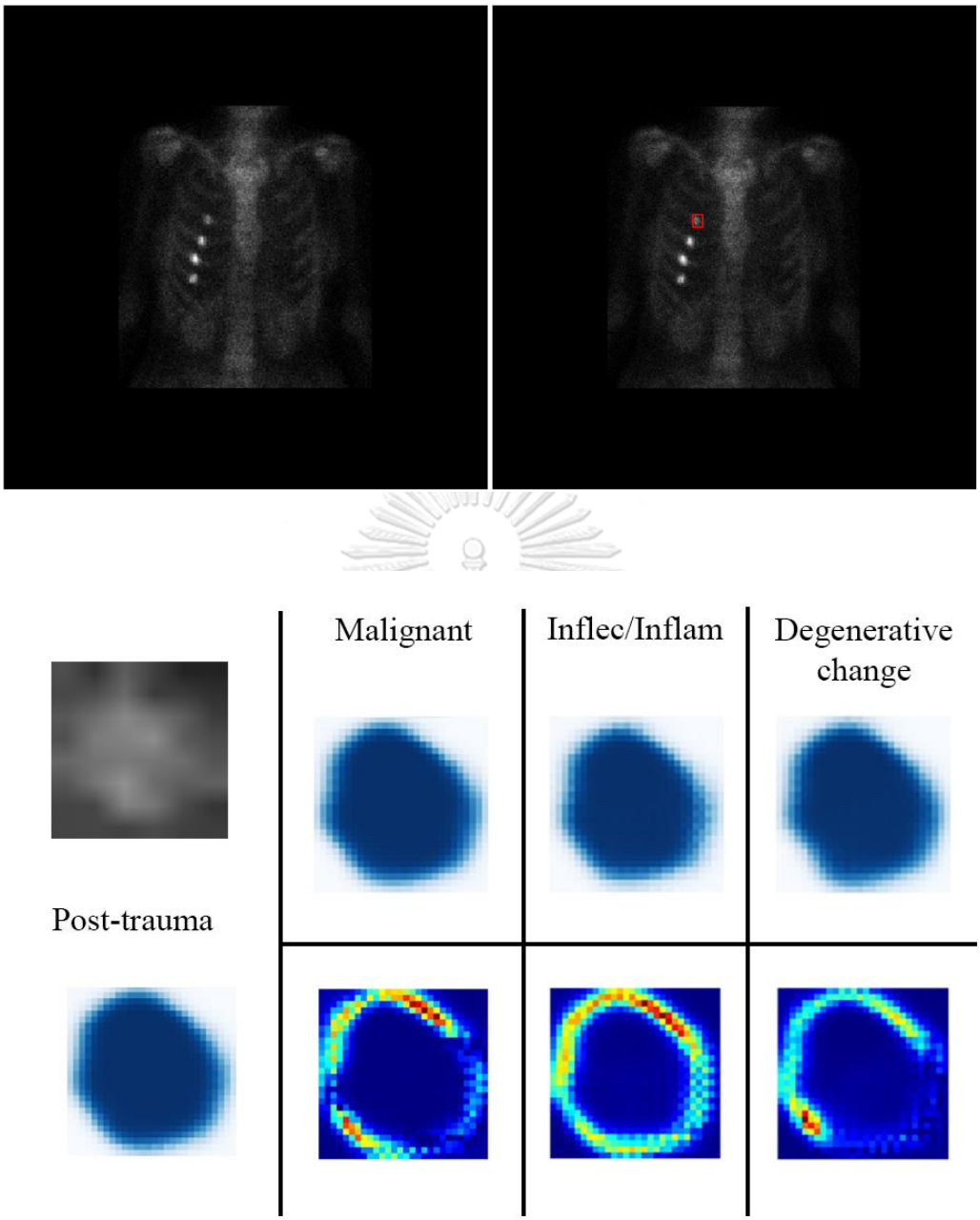


Figure 33: The visualization of mask prediction of each class. The most left side is the mask prediction of post-trauma. The difference between the ground truth class and other classes are shown in the bottom.

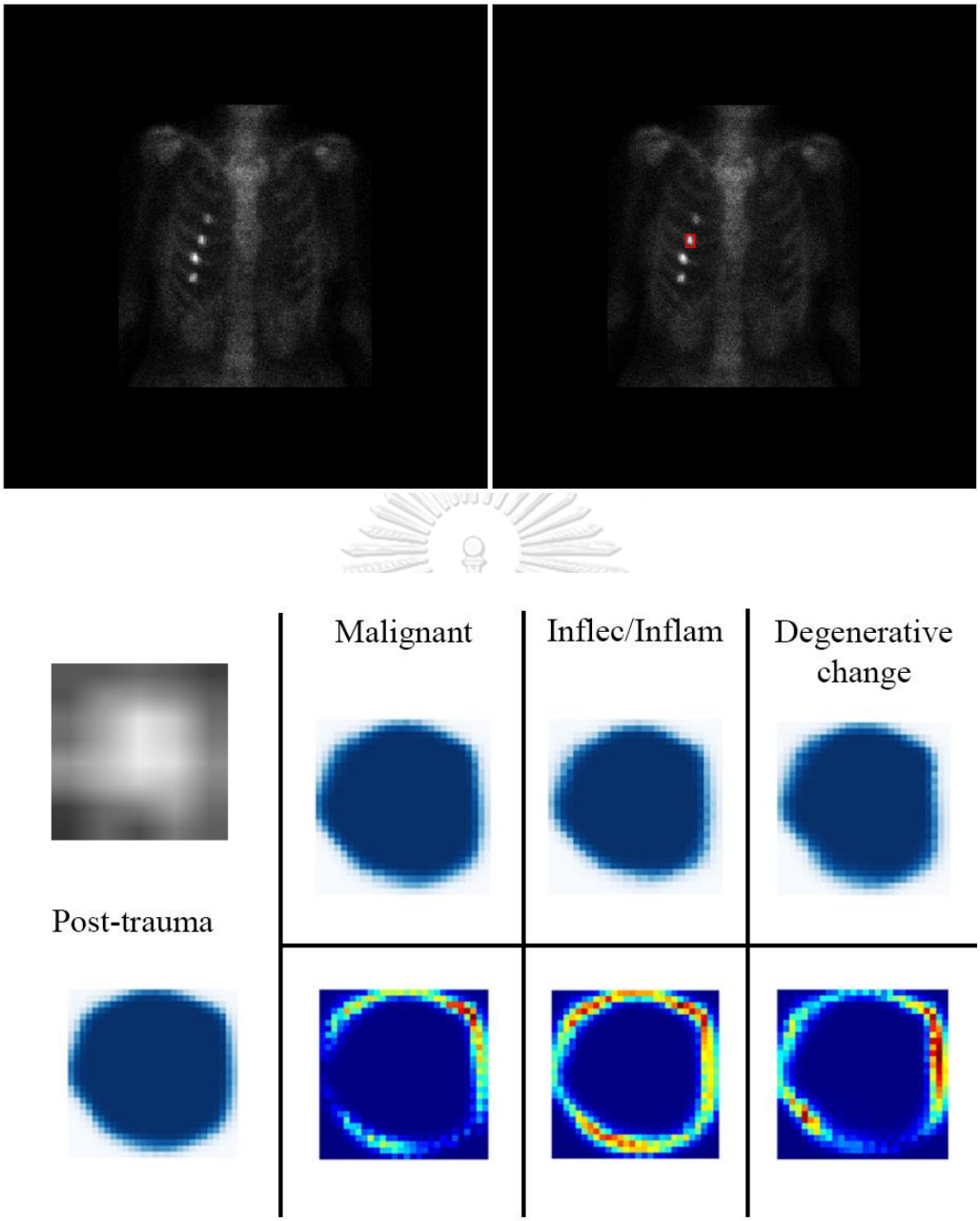


Figure 34: The visualization of mask prediction of each class. The most left side is the mask prediction of post-trauma. The difference between the ground truth class and other classes are shown in the bottom.

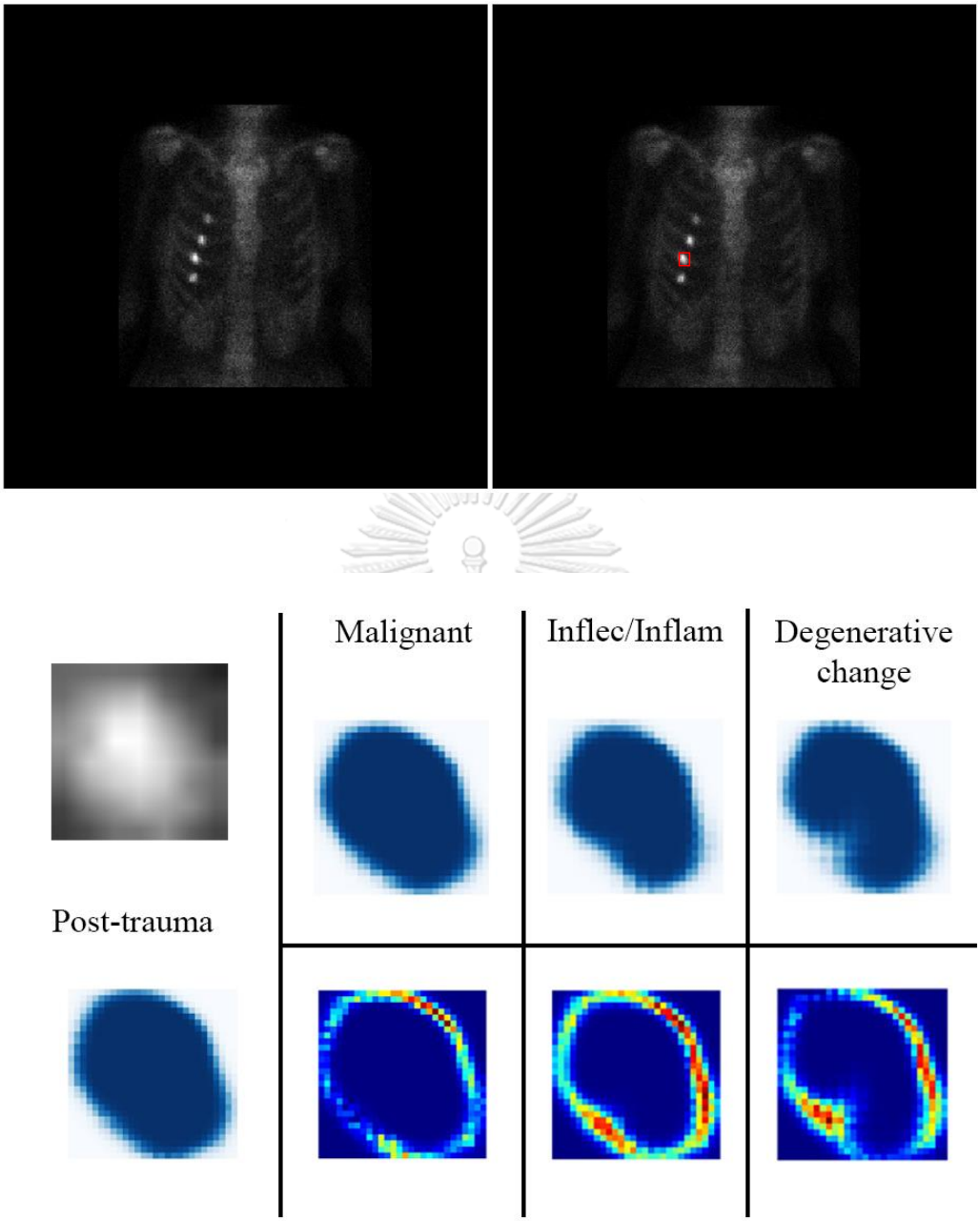


Figure 35: The visualization of mask prediction of each class. The most left side is the mask prediction of post-trauma. The difference between the ground truth class and other classes are shown in the bottom.

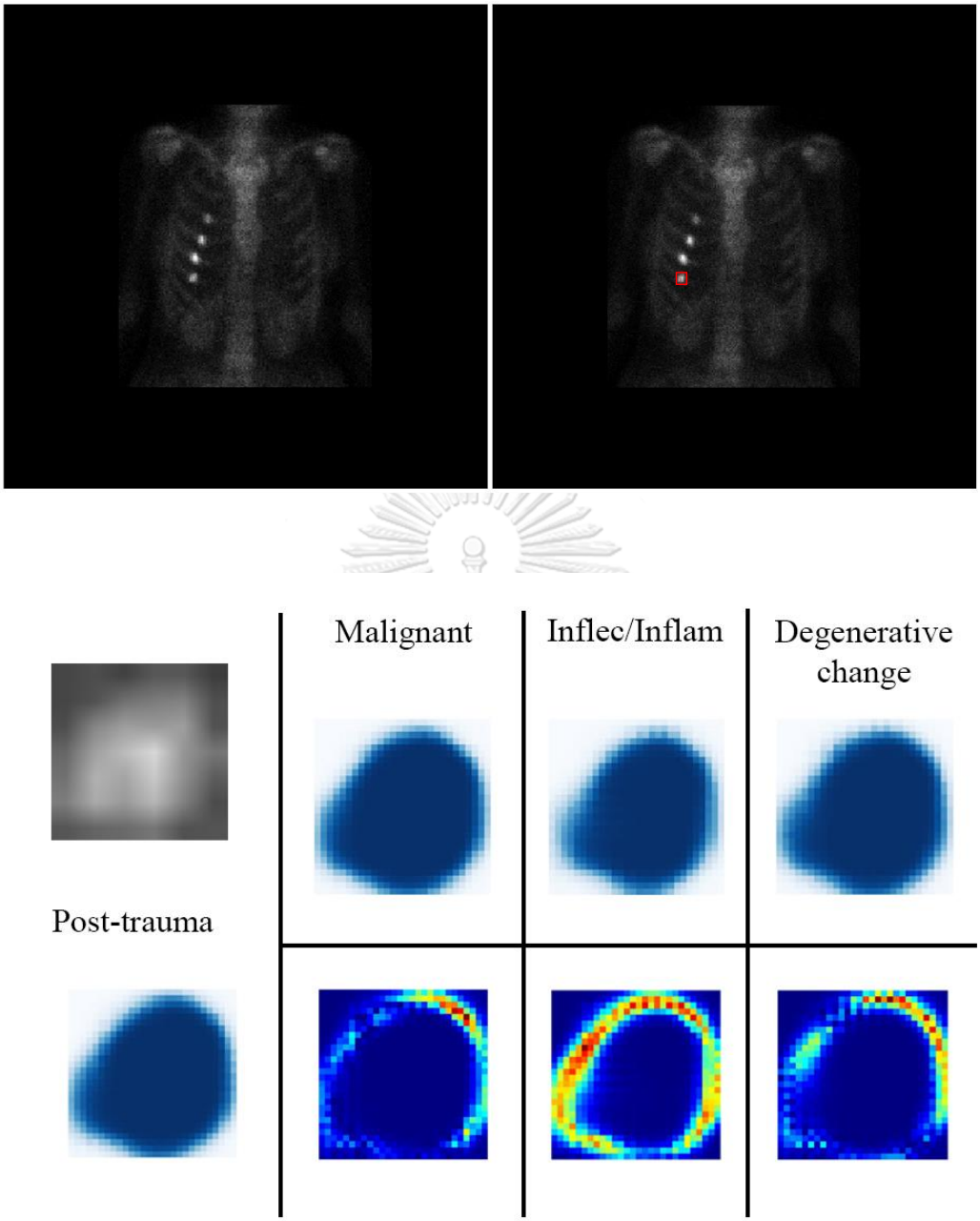


Figure 36: The visualization of mask prediction of each class. The most left side is the mask prediction of post-trauma. The difference between the ground truth class and other classes are shown in the bottom.

Form the following results, the Intensity, curvature, and shape of the lesion mask are affected by the different types of mask predictions. But, the mask prediction of each class tends to be very different at the edges of the lesion, especially in the curved area. If the lesion is round, the difference of the mask between malignant and post-trauma is small as shown in Figure 34-36, which makes it difficult to classify both types when looking at just one lesion. Degenerative change and inflection/inflammation are also quite different from the mask at the edges and inside the lesion, as shown in Figures 29 and 31. Furthermore, post-trauma and inflection/inflammation have clearly different masks at the edges of the lesion as shown in Figure 34-36.

The above interpretation is the interpretation of experimental results, which is different from the physician analysis because the interpretation from mask visualization is only focused on one lesion. On the contrary, the physician will analyze from the whole image, which makes the mask visualization may not reflect the true differences.

6.7 Results of global features visualization

In addition to visualizing the model, we have also visualized the global features of each image. We applied PHATE (Moon, van Dijk et al. 2017), a dimensionality reduction method for visualizing trajectory structures in high-dimensional biological data, with the global features of bone scintigraphy. The results were compared with the ground-truth of the lesion types that appeared in that image.

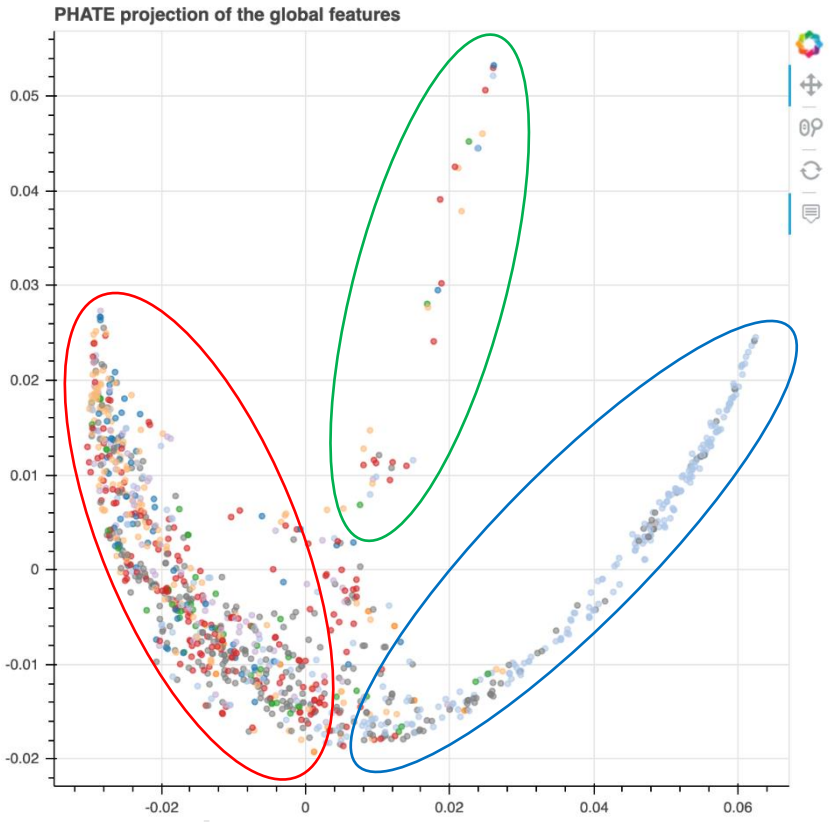


Figure 37: The results of global features of bone scintigraphy visualization using PHATE

From the results in Figure 37, we can separate the global features into three groups which are the malignant group (the blue circle group on the right side in the figure), non-malignant (the red circle group on the left side), outlier group (the green group circle on the middle in figure).

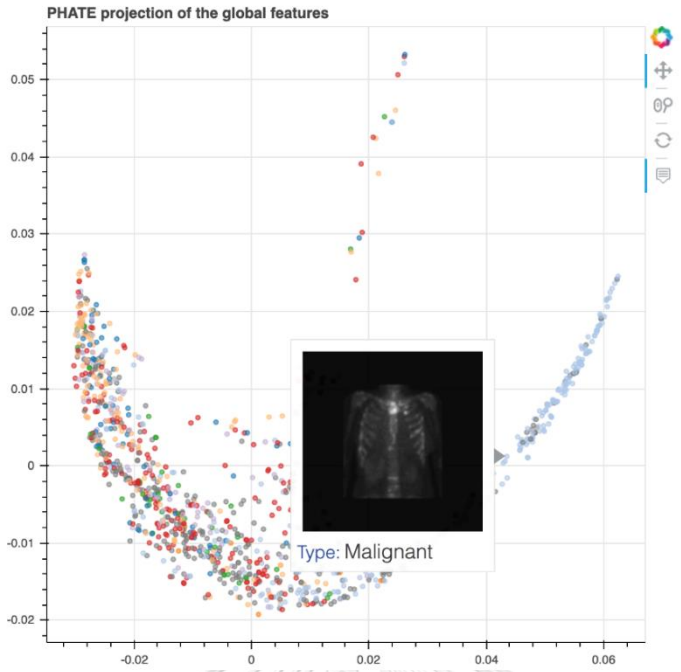


Figure 38: The sample of bone scintigraphy in the malignant group (blue circle in figure 37).

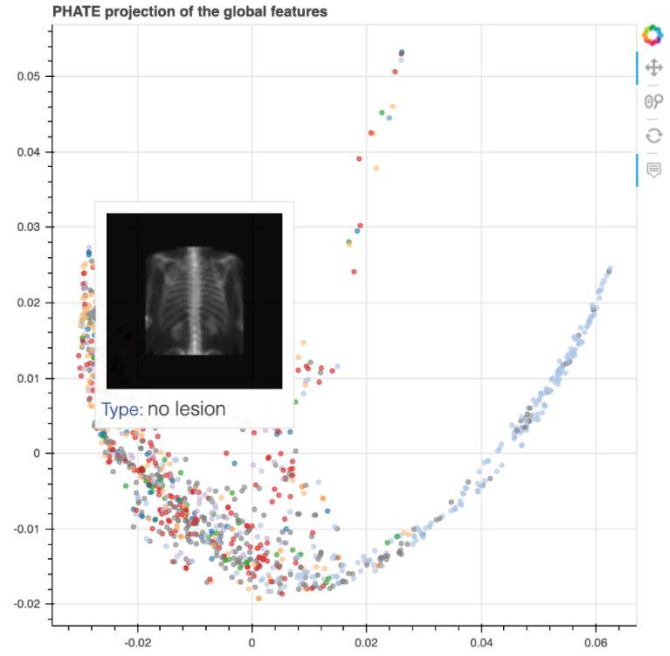


Figure 39: The sample of bone scintigraphy in the non-malignant group (red circle in figure 37).

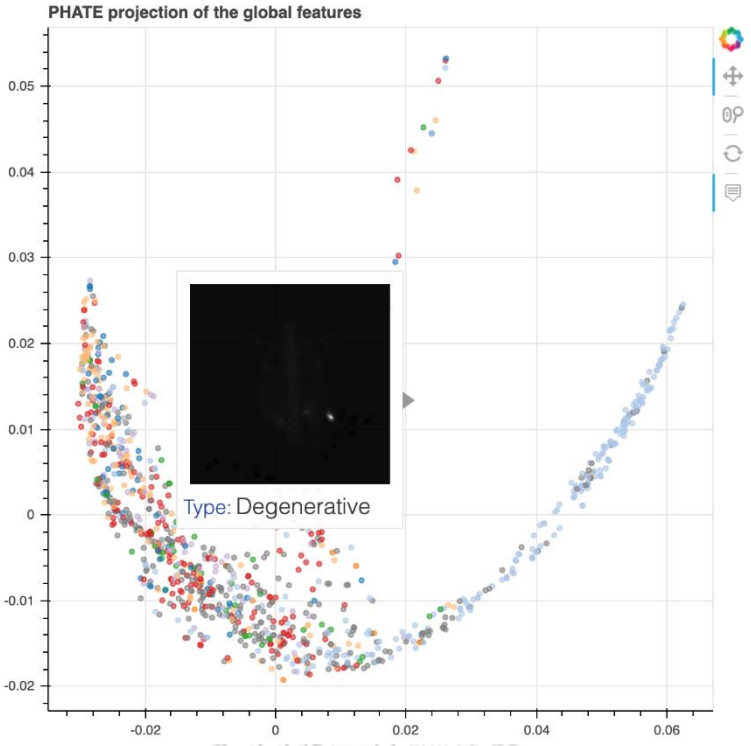


Figure 40: The sample of bone scintigraphy in the outlier group (green circle in figure 37).

We sampled the bone scintigraphy in each group to analyze the pattern in each group. The results show that the images which consist of "only malignant lesions" have similar features (blue points in figure 38). For multi-lesion types, the global feature tends to be close together such as "no lesion" which shown in Figure 39. Moreover, we also analyzed the outlier of the global feature which is shown in Figure 40. The images in the outlier group seem to be a dark image that may not give the information to classify the lesion types.

7. Discussion

7.1. The effect of applying each technique in Malignet

MaligNet reached an f1-score of up to 0.848 over the baseline of 0.816. Based on the results in Table 4, we studied the effectiveness of LFPN and global features. The results show that the effectiveness of LFPN makes our model more accurate than using global features. However, applying both techniques outperforms using only one technique.

7.2. The limitation of using unlabeled data

The usefulness of unlabeled data is limited. At some point, when the increase in the amount of unlabeled data reaches a saturation point, the efficiency of the model does not greatly increase. By contrast, increasing the labeled data still improves the f1-score. However, labeling bone scintigraphy data is a time-consuming task. Thus, MaligNet is a good choice for utilizing unlabeled data with significant time and resource savings.

7.3. Analysis of the prediction results

As shown in Figure 15.b and Figure 16.f, applying global features appears to help the model in categorizing lesions more accurately. MaligNet uses not only the lesion features but also global information to categorize the lesion types. However, caution is needed in applying global features to avoid relying too much on global features rather than lesion features. As a result, MaligNet tends to predict malignant lesions more often than other types. For this reason, MaligNet has higher sensitivity than the baseline, as shown in Figures 15.a and 15.d. This occasionally causes a false positive, as shown in the examples presented in Figure 16.h.

7.4. Difference between the LFPN and self-training

Self-training is a useful approach to semi-supervised learning. We also trained MaligNet using the self-training method, which improves the model even further. As shown in Table 4, the f1-score improves from 0.848 to 0.851 after self-training. Self-training and the LFPN can be considered different ways to learn from unlabeled data. The LFPN, which is similar to an autoencoder, tries to learn better data representation, while self-training provides discriminative information that helps the classification task. In the case of using only the LFPN, our method has a slightly lower f1-score than self-training. However, MaligNet can be trained in one step on both types of data simultaneously, which takes less time in training than the self-training. The training and inference time without self-training of the Mask R-CNN was 19.6 hours and 0.76 milliseconds, respectively; on the other hand, Malignet took 23 hours and 0.87 milliseconds, respectively. Models with self-training required twice the amount of time to train. Moreover, when we combined both techniques, our method is more accurate.

8. Conclusions and Future work

Almost all object detection or instance segmentation models are designed for supervised learning, which requires a large amount of labeled data in the training process. However, our medical image dataset has a small amount of labeled data, which can lead to model overfitting. We focused on using unlabeled data to leverage its utility and realize the most effective model possible with a limited amount of labeled data. Therefore, we proposed MaligNet, a ladder network extension of Mask R-CNN for lesion instance segmentation in bone scintigraphy that uses semi-supervised learning for training.

MaligNet is a single network that is simple, effective, flexible, and lightweight. Normally, semi-supervised models must be trained in multiple steps. However, MaligNet is an end-to-end solution that can be trained in one step with both labeled and unlabeled data simultaneously, which reduces the training time. Our data are bone scintigraphy images, which have a similar pattern, characteristics, and composition among the images, unlike general images. For this reason, the LFPN can take advantage of the specificity of the data that enables the model to learn the representation of the bone scan image from unlabeled data. Furthermore, applying global features helps to classify the lesion types based on the overall composition of the image, which mimics the diagnostic approach of physicians.

We evaluated the model using the mean precision, mean sensitivity, and mean f1-score in the lesion instance segmentation task and the accuracy, precision, sensitivity, specificity, and f1-score in bone cancer metastasis prediction. MaligNet significantly outperforms the baseline model by up to 2.33% without global features and by 3.92% with global features.

We plan to compare our results with those of a nuclear medicine physician as a gold standard to determine the difference in decision making between a machine and physician for performance improvements. In further analyses, we plan to visualize the model to determine what the model sees and the reasons for categorizations by the model. We also plan to apply our model to other domains, e.g., MRI and CT. Finally, we believe that our method provides an alternative approach for handling unlabeled data and will be useful for applications in other works.

9. Appendix

9.1 APPENDIX A: Details of patient gender and age in the dataset

The details of the patients' gender data and age statistics, which are divided into supervised training, validation, testing, and unsupervised datasets, are shown in Table 9. We also display the age range of the patients in a histogram in Figure 41 - 44.

Table 9: Details of the patients' gender and age statistics for each dataset type.

Dataset type	Male images	Female images	Min age	Mean age	Max age
Supervised training data	274	467	2	59.16	96
Supervised validation data	86	145	5	58.94	96
Supervised testing data	38	78	2	59.02	90
Unsupervised data	7,624	10,936	2	57.40	97

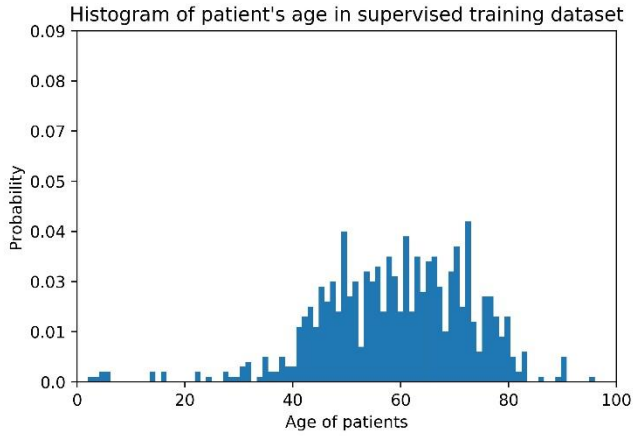


Figure 41: A histogram of patient age at bone scintigraphy in the supervised training dataset.

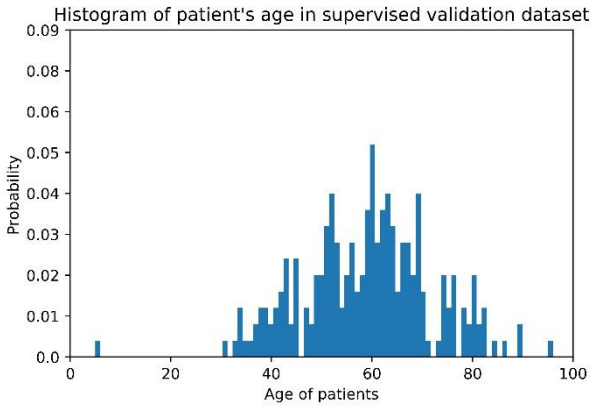


Figure 42: A histogram of patient age at bone scintigraphy in the supervised validation dataset.

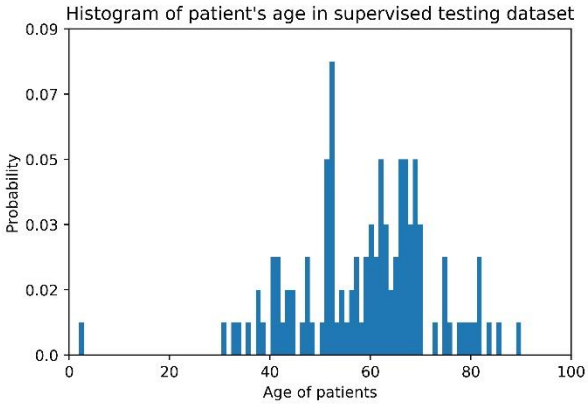


Figure 43: A histogram of patient age at bone scintigraphy in the supervised testing dataset.

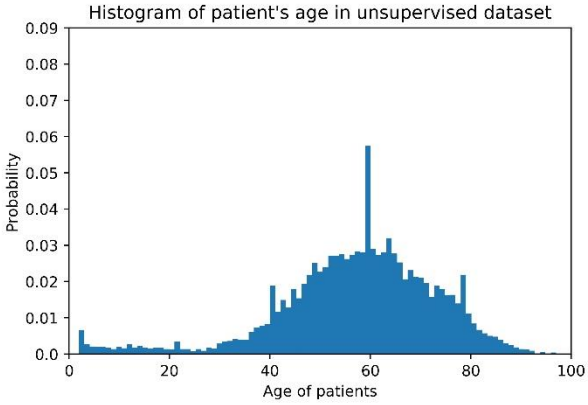


Figure 44: A histogram of patient age at bone scintigraphy in the unsupervised dataset.

9.2 APPENDIX B: Hyperparameters of Single Shot MultiBox Detector (SSD) in the experiments of chest detection

In Section 6.1, our SSD has pretrained weights from ImageNet and is retrained with our data using the hyperparameters, as shown in Table 10.

Table 10: Final values of the hyperparameters used in the chest detection experiment.

Parameters	Parameter used
Image size (width,height)	(512,512)
Core network	VGG-16
Batch size	16
Optimizer	Adam
Learning rate	0.001
Weight decay	0.0005
L2 regularization	0.0005
IoU threshold	0.45
Anchor box scaling factors	[0.07, 0.15, 0.3, 0.45, 0.6, 0.75, 0.9, 1.05]
Anchor box steps	[8, 16, 32, 64, 128, 256, 512]
Anchor box offsets	[0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5]

9.3 APPENDIX C: Hyperparameters of MaligNet n the experiments of instance segmentation

All experiments in Section 6.2 use the same hyperparameters. We trained the network from scratch without pre-trained weights. We attempted to optimize the performance of the model by searching for the optimal hyperparameters to the greatest extent possible. The final hyperparameter values are shown in Table 11. Because all cost functions are self-normalized and the costs do not largely vary, we use λ equal to one for all experiments. For the batch size hyperparameter, we found that larger batches led to more accurate results. Due to GPU resource limitations, 16 is the maximum batch size that can be used.

Table 11: Final values of hyperparameters used in the lesion instance segmentation experiment from the parameter search.

Parameters	Parameter search	Final parameters
Image size (width,height)	(320,320), (512,512)	(320,320)
$\lambda_{rc}, \lambda_{rb}, \lambda_{cc}, \lambda_{cb}, \lambda_{cm}$	1.0	1.0
Gaussian noise ratio	0.03, 0.05, 0.3	0.03
Batch size	2, 8, 16 (maximum batch size)	16
Optimizer	Adam	Adam
Learning rate	0.0001, 0.005, 0.002, 0.001, 0.01, 0.02	0.001
Weight decay	0.0001, 0.001, 0.01, 0.01, 0.02	0.0001
RPN NMS threshold	0.6, 0.7, 0.9, 0.99	0.7
Train ROI per image	80, 100, 200, 300	200
RPN anchor scales	(32, 64, 128, 256, 512)	(32,64,128,256,512)
RPN anchor ratio	[0.5, 1, 2]	[0.5, 1, 2]
RPN anchor stride	1	1
RPN anchor per image	256	256
Max ground truth instances	50, 100, 200, 300	100
Detection minimum confidence	0.7	0.7

REFERENCES

Ambellan, F., et al. (2019). "Automated segmentation of knee bone and cartilage combining statistical shape knowledge and convolutional neural networks: Data from the Osteoarthritis Initiative." Medical image analysis **52**: 109-118.

Azmi, R., et al. (2011). "IMPST: a new interactive self-training approach to segmentation suspicious lesions in breast MRI." Journal of medical signals and sensors **1**(2): 138.

Belcher, L. (2017). "Convolutional Neural Networks for Classification of Prostate Cancer Metastases Using Bone Scan Images."

Bustamante, M., et al. (2018). "Automated multi-atlas segmentation of cardiac 4D flow MRI." Medical image analysis **49**: 128-140.

Cheplygina, V., et al. (2019). "Not-so-supervised: a survey of semi-supervised, multi-instance, and transfer learning in medical image analysis." Medical image analysis.

Confavreux, C. B., et al. (2019). "Bone metastases from lung cancer: A paradigm for multidisciplinary onco-rheumatology management." Joint Bone Spine **86**(2): 185-194.

Courbariaux, M., et al. (2016). "Binarized neural networks: Training deep neural networks with weights and activations constrained to+ 1 or-1." arXiv preprint arXiv:1602.02830.

Dang, J. (2016). "Classification in bone scintigraphy images using convolutional neural networks." Master's Theses in Mathematical Sciences.

de Vos, B. D., et al. (2016). 2D image classification for 3D anatomy localization: employing deep convolutional neural networks. Medical Imaging 2016: Image Processing, International Society for Optics and Photonics.

Deng, J., et al. (2009). Imagenet: A large-scale hierarchical image database. 2009 IEEE conference on computer vision and pattern recognition, Ieee.

Everingham, M., et al. (2010). "The pascal visual object classes (voc) challenge." International journal of computer vision **88**(2): 303-338.

Geng, S., et al. (2015). Combining CNN and MIL to assist hotspot segmentation in bone

scintigraphy. International Conference on Neural Information Processing, Springer.

Geng, S., et al. (2016). A mil-based interactive approach for hotspot segmentation from bone scintigraphy. 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE.

Girshick, R. (2015). Fast r-cnn. Proceedings of the IEEE international conference on computer vision.

Girshick, R., et al. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. Proceedings of the IEEE conference on computer vision and pattern recognition.

Goodfellow, I., et al. (2016). Deep learning, MIT press.

Graham, S., et al. (2019). "MILD-Net: Minimal information loss dilated network for gland instance segmentation in colon histology images." Medical image analysis **52**: 199-211.

Han, S., et al. (2015). "Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding." arXiv preprint arXiv:1510.00149.

Han, S., et al. (2015). Learning both weights and connections for efficient neural network. Advances in neural information processing systems.

Han, Z., et al. (2018). "Spine-GAN: Semantic segmentation of multiple spinal structures." Medical image analysis **50**: 23-35.

He, K., et al. (2017). Mask r-cnn. Proceedings of the IEEE international conference on computer vision.

He, K., et al. (2016). Deep residual learning for image recognition. Proceedings of the IEEE conference on computer vision and pattern recognition.

Heinrich, M. P., et al. (2019). "OBELISK-Net: Fewer layers to solve 3D multi-organ segmentation with sparse deformable convolutions." Medical image analysis **54**: 1-9.

Huang, G., et al. (2017). Densely connected convolutional networks. Proceedings of the IEEE conference on computer vision and pattern recognition.

Hubara, I., et al. (2017). "Quantized neural networks: Training neural networks with low precision weights and activations." The Journal of Machine Learning Research **18**(1): 6869-6898.

Ibrahim, T., et al. (2013). "Bone and cancer: the osteoncology." Clinical Cases in Mineral and Bone Metabolism **10**(2): 121.

Institute, N. C. (2018, 2018-20-11). "Primary Bone Cancer." Retrieved 03, 2019, from https://www.cancer.gov/types/bone/bone-fact-sheet.

Kamnitsas, K., et al. (2017). "Efficient multi-scale 3D CNN with fully connected CRF for accurate brain lesion segmentation." Medical image analysis **36**: 61-78.

Kang, S. K., et al. (2019). "Unsupervised lesion detection in bone scintigraphy using deep learning-based image inpainting technology." Journal of Nuclear Medicine **60**(supplement 1): 403-403.

Kervadec, H., et al. (2019). "Constrained-CNN losses for weakly supervised segmentation." Medical image analysis **54**: 88-99.

Küstner, T., et al. (2018). Semantic organ segmentation in 3d whole-body mr images. 2018 25th IEEE International Conference on Image Processing (ICIP), IEEE.

Lin, T.-Y., et al. (2017). Feature pyramid networks for object detection. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.

Lin, T.-Y., et al. (2017). Focal loss for dense object detection. Proceedings of the IEEE international conference on computer vision.

Lin, T.-Y., et al. (2014). Microsoft coco: Common objects in context. European conference on computer vision, Springer.

Liu, W., et al. (2016). Ssd: Single shot multibox detector. European conference on computer vision, Springer.

Magee, D. J., et al. (2015). Pathology and intervention in musculoskeletal rehabilitation, Elsevier Health Sciences.

Moon, K. R., et al. (2017). "PHATE: a dimensionality reduction method for visualizing trajectory structures in high-dimensional biological data." bioRxiv: 120378.

Payer, C., et al. (2016). Regressing heatmaps for multiple landmark localization using CNNs. International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer.

Rajpurkar, P., et al. (2017). "Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning." arXiv preprint arXiv:1711.05225.

Rasmus, A., et al. (2015). Semi-supervised learning with ladder networks. Advances in neural information processing systems.

Raza, S. E. A., et al. (2019). "Micro-Net: A unified model for segmentation of various objects in microscopy images." Medical image analysis **52**: 160-173.

Redmon, J., et al. (2016). You only look once: Unified, real-time object detection. Proceedings of the IEEE conference on computer vision and pattern recognition.

Ren, S., et al. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. Advances in neural information processing systems.

Selvaraju, R. R., et al. (2017). Grad-cam: Visual explanations from deep networks via gradient-based localization. Proceedings of the IEEE International Conference on Computer Vision.

Siegel, R. L., et al. (2019). "Cancer statistics, 2019." CA: a cancer journal for clinicians **69**(1): 7-34.

Simonyan, K. and A. Zisserman (2014). "Very deep convolutional networks for large-scale image recognition." arXiv preprint arXiv:1409.1556.

Tong, Q., et al. (2019). "RIANet: Recurrent interleaved attention network for cardiac MRI segmentation." Computers in biology and medicine **109**: 290-302.

Tsui, B. M., et al. (1981). "Analysis of recorded image noise in nuclear medicine." Physics in Medicine & Biology **26**(5): 883.

Vincent, P., et al. (2008). Extracting and composing robust features with denoising autoencoders. Proceedings of the 25th international conference on Machine learning, ACM.

Wang, S., et al. (2019). "CT Male Pelvic Organ Segmentation Using Fully Convolutional Networks with Boundary Sensitive Representation." Medical image analysis.

Winkels, M. and T. S. Cohen (2019). "Pulmonary Nodule Detection in CT Scans with Equivariant CNNs." Medical image analysis.

Xu, Y., et al. (2017). "Gland instance segmentation using deep multichannel neural networks." IEEE Transactions on Biomedical Engineering **64**(12): 2901-2912.

Yang, D., et al. (2015). Automated anatomical landmark detection ondistal femur surface using convolutional neural network. 2015 IEEE 12th international symposium on biomedical imaging (ISBI), IEEE.

Zeng, M., et al. (2017). Semi-supervised convolutional neural networks for human activity recognition. 2017 IEEE International Conference on Big Data (Big Data), IEEE.

VITA

NAME	Terapap Apiparakoon
DATE OF BIRTH	8 August 1995
PLACE OF BIRTH	Bangkok
INSTITUTIONS ATTENDED	Chulalongkorn University
HOME ADDRESS	2004, Prachasongkroh Road, Din Daeng Zone Din Daeng District, Bangkok 10400
AWARD RECEIVED	First place in Open Topis : Artificial Intelligence and Robotics innovation contest (AIROBIC) 2018