





**AN APPROACH FOR THE CHEST TUBE DETECTION  
IN CHEST RADIOGRAPHS  
USING CONVOLUTIONAL NEURAL NETWORK**

**Ph.D. THESIS**

**Cem Ahmet MERCAN**

**Department of Computational Science and Engineering**

**Computational Science and Engineering Programme**

**FEBRUARY 2015**



**AN APPROACH FOR THE CHEST TUBE DETECTION  
IN CHEST RADIOGRAPHS  
USING CONVOLUTIONAL NEURAL NETWORK**

**Ph.D. THESIS**

**Cem Ahmet MERCAN  
(702002003)**

**Department of Computational Science and Engineering**

**Computational Science and Engineering Programme**

**Thesis Advisor: Prof. Dr. M. Serdar ÇELEBİ**

**FEBRUARY 2015**



**KATMANLI SİNİR AĞLARI KULLANILARAK  
GÖĞÜS RADYOGRAFİLERİNDE  
GÖĞÜS TÜPÜ TESPİTİ İÇİN BİR YAKLAŞIM**

**DOKTORA TEZİ**

**Cem Ahmet MERCAN  
(702002003)**

**Hesaplamalı Bilim ve Mühendislik Anabilim Dalı**

**Hesaplamalı Bilim ve Mühendislik Programı**

**Tez Danışmanı: Prof. Dr. M. Serdar ÇELEBİ**

**ŞUBAT 2015**









## **FOREWORD**

First of all, I would like to thank Allah who created and blessed me.

I would like to express my deep appreciation and thanks for my advisor who is Prof. Dr. M. Serdar Çelebi constantly helped me. I deeply appreciate his understanding. I am also thankful to him for his kindness and precious mentoring.

In addition, I would like to thank my coworkers. Especially, I owe them who helped me on the format of my thesis and gave me suggestions.

Finally, I appreciate all help that I have had from my family.

February 2015

Cem Ahmet MERCAN





2.4.7 Proper learning rate: .....	27
<b>3. INITIAL EXPERIMENTS .....</b>	<b>29</b>
3.1 Removing 'U' from the Chest Radiography .....	29
3.1.1 Using more dense data set .....	30
3.1.2 Results of the removing 'U' case using dense data set .....	33
<b>4. CNN FOR DETECTION OF CHEST TUBES .....</b>	<b>37</b>
4.1 Training and Testing Datasets .....	37
4.2 CNN Architecture .....	38
<b>5. NURBS CURVES .....</b>	<b>45</b>
5.1 The Noise and the Continuity .....	45
5.2 The NURBS Curve Formula .....	47
<b>6. RESULTS AND EVALUATION .....</b>	<b>49</b>
6.1 Model Performance Evaluation .....	49
6.2 Results .....	49
<b>7. DISCUSSION AND CONCLUSIONS .....</b>	<b>55</b>
7.1 Discussion .....	55
7.1.1 Limitations and Source of Errors .....	55
7.1.2 Parallelization .....	56
7.1.3 To clean the output of CNN and Curve Fitting .....	56
7.1.4 Usability in medical practice .....	57
7.2 Concluding Remarks .....	57
7.2.1 Chest Tube Inpainting .....	57
7.2.2 CNN Architecture .....	58
7.3 Conclusion .....	58
7.3.1 Other Potential Usage of Method .....	59
7.3.2 Future Work .....	59
<b>8. ACKNOWLEDGMENTS .....</b>	<b>61</b>
<b>REFERENCES .....</b>	<b>63</b>
<b>APPENDICES .....</b>	<b>67</b>

## ABBREVIATIONS

<b>AD</b>	: Automated Diagnosis
<b>ANN</b>	: Artificial Neural Network
<b>CAD</b>	: Computer Aided Diagnosis
<b>CE</b>	: Cross-Entropy
<b>CNN</b>	: Convolutional Neural Network
<b>CT</b>	: Computed Tomography
<b>D</b>	: Dimension
<b>FN</b>	: False Negative
<b>FP</b>	: False Positive
<b>JSRT</b>	: The Japanese Society of Radiological Technology
<b>MR</b>	: Magnetic Resonance
<b>NN</b>	: Neural Network
<b>NURBS</b>	: Non-Uniform Rational B-Spline
<b>PA</b>	: Postero-Anterior
<b>PACS</b>	: Picture Archiving and Communication System
<b>RIDER</b>	: The Reference Image Database to Evaluate Response
<b>RMS</b>	: Root Mean Square
<b>ROC</b>	: Receiver Operating Characteristic
<b>ROI</b>	: Region of Interest
<b>SDLM</b>	: Stochastic Diagonal Levenberg-Marquardt
<b>TN</b>	: True Negative
<b>TP</b>	: True Positive
<b>WHO</b>	: World Health Organization





## LIST OF TABLES

	<u>Page</u>
<b>Table 2.1</b> : Input Perturbing Results. ....	24
<b>Table 2.2</b> : Net Sum ( <i>net</i> ) Perturbing Results. ....	24
<b>Table 2.3</b> : Weight perturbing Results. NP: not passed.....	24
<b>Table 2.4</b> : Activation Function Perturbing Results. NP: not passed. ....	25
<b>Table 3.1</b> : Best results of removing 'U' with using sparse dataset. Each of Input and Output Layers contains one node. N.: Nodes, I.W.:Initial Weights, L.R.:Learning Rate. ....	31
<b>Table 3.2</b> : Best Results of Removing 'U' Case. Each of Input and Output Layers contains one node. N.: Nodes. ....	34
<b>Table 4.1</b> : Training and testing data sets. ....	37
<b>Table 6.1</b> : Pixel-based average RMS error values of the raw output of the Neural Network. All of the results are obtained by using CNN in 5 layers (2, 32, 32, 128, 1 nodes per layer; 32, 16, 128, 1 links per node). ....	50
<b>Table 6.2</b> : Pixel-based average ROC values per image. TP: true positive, TN: true negative, FP: false positive, FN: false negative. ....	51



## LIST OF FIGURES

	<u>Page</u>
<b>Figure 1.1</b> : A normal chest radiography. ....	2
<b>Figure 1.2</b> : An example for artificial object at the chest radiography. ....	3
<b>Figure 1.3</b> : Another example for artificial object at the chest radiography. ....	4
<b>Figure 1.4</b> : A chest radiography that contains artificial object from the JSRT database. ....	5
<b>Figure 1.5</b> : Mediastinum Region of the Chest. ....	7
<b>Figure 1.6</b> : Simplified Flowchart of the proposed system. ....	8
<b>Figure 2.1</b> : The Difference Between Traditional Neural Network Node and CNN Node. ....	12
<b>Figure 2.2</b> : Details of a CNN architecture for image processing. ....	13
<b>Figure 2.3</b> : The convolution operation. ....	15
<b>Figure 3.1</b> : Not An Easy Training Case (Object Removing). ....	29
<b>Figure 3.2</b> : The work plan for Removing 'U' Test Case. ....	30
<b>Figure 3.3</b> : Removing 'U' Case Training Results. ....	31
<b>Figure 3.4</b> : The Dense Data Set. ....	31
<b>Figure 3.5</b> : The Comparison Graph of Sparse and Dense Data Set. ....	32
<b>Figure 3.6</b> : The Dense Data Set Results. ....	32
<b>Figure 3.7</b> : The Incorrect Part of the Input Image. ....	33
<b>Figure 3.8</b> : The Best Result and Test Images. ....	33
<b>Figure 3.9</b> : The Comparison Graph of Sparse, Dense Data Set and Best Result of Removing 'U' Case. ....	35
<b>Figure 4.1</b> : Two Scales of Input Blocks. ....	38
<b>Figure 4.2</b> : The Final CNN for Detection of Chest Tubes. ....	39
<b>Figure 4.3</b> : Selected successful results from our test data set. The left side images are inputs of the CNN model. The right side images are the outputs of the CNN model. ....	40
<b>Figure 4.4</b> : Selected successful results from our test data set. The left side images are inputs of the CNN model. The right side images are the outputs of the CNN model. ....	41
<b>Figure 4.5</b> : Selected erroneous results from our test data set. The left side images are inputs of the CNN model. The right side images are the outputs of the CNN model. ....	42
<b>Figure 4.6</b> : Selected erroneous results from our test data set. The left side images are inputs of the CNN model. The right side images are the outputs of the CNN model. ....	43
<b>Figure 5.1</b> : The stages of our model. The left image is the input, the middle image is the output of the CNN, and the right image is the result of the curve fitting process. ....	45

**Figure 5.2** : The Selection of the Control Points..... 47

**Figure 6.1** : Selected successful results from our test data set. The backgrounds are input x-rays. 'X' marks show the outputs of the proposed model. .... 50

**Figure 6.2** : Selected erroneous results from our test data set. The backgrounds are input x-rays. 'X' marks show the outputs of the proposed model. .... 52

**Figure 6.3** : Selected examples from the JSRT test data set results. Because there is no chest tube present, the chest tube markers are not present in these images. The backgrounds are input x-rays. .... 53

# **AN APPROACH FOR THE CHEST TUBE DETECTION IN CHEST RADIOGRAPHS USING CONVOLUTIONAL NEURAL NETWORK**

## **SUMMARY**

The chest radiography is the most common radiological modality in practice. Especially, it is very popular for scanning and screening purposes. Unfortunately, at the same time it is one of the most difficult radiological modalities. It is known that overlapping tissues cause highly complex projections in chest radiographs. In addition, artificial objects, such as catheters, chest tubes, and pacemakers can appear on these radiographs. It is important that anomaly detection algorithms are not confused by these objects.

To achieve this goal, we propose an approach to train a Convolutional Neural Network (CNN) to detect chest tubes present on radiographs. To better detect the chest tube skeleton as the final output, non-uniform rational B-spline (NURBS) curves are used to automatically fit to the CNN output.

We have chosen CNN method, as Neural Network (NN) model, because of its shift, scale, and distortion invariance abilities and weight sharing property for reducing number of free parameters of neural network. It is inspired by a modified version of the vision system in vertebrate animals and it simulates some image processing, feature extraction, and pattern recognition stages.

This is the first study conducted to automatically detect artificial objects in the lung region of chest radiographs. Other automatic detection schemes work on the mediastinum.

According to our initial tests, we decided to use the Gradient Descent and Cross-Entropy algorithms with a sigmoid activation function with 5 layers. Our final CNN architecture contains 2, 32, 32, 128, and 1 nodes for the successive layers. Between layers, there are 32, 16, 128, and 1 links for each node in the layers. After a series of tuning tests, the learning rate was selected as 0.1.

To feed the system with a greater input region without increasing the model complexity, a multi-scale input with two scales was used. The input image blocks used by the input layer and cropped from two whole images of a chest radiograph at multi-scale sizes of 1000x1000 and 250x250 pixels, are used without any registration. The input layer contains two nodes with two input blocks of 13x13 pixels each.

During the training stage, blocks are selected from random training image sets with a random block position. At three hidden layers, the output sizes are chosen as 5x5, 1x1 and 1x1. Finally, the output layer contains a single node that gives outputs of 1x1 in size, which is used as a pixel in the resulting image at the proper location according to the input block position.

The training set is solely constructed from our data set, and contains 62 radiographs. 13 of the images contain two chest tubes, 25 contain a single chest tube, and 24 out

of 62 radiographs have no chest tube. We tested our model using two datasets. The first dataset contains our test set, and it is constructed with 21 radiographs with similar groupings. To distinguish the chest tubes from other artificial objects, our training and testing sets contain 36 and 13 radiographs, respectively, including artificial objects other than chest tubes. The second test set contains 247 images without chest tubes, and it is publicly available; it was created by the Standard Digital Image Database Project Team of the Scientific Committee of the Japanese Society of Radiological Technology (JRST)

We evaluated the performance of the model using a pixel-based ROC analysis. Each true positive, true negative, false positive and false negative pixel is counted and used for calculating average accuracy, sensitivity, and specificity percentages. The results were 99.99 % accuracy, 59 % sensitivity, and 99.99 % specificity.

The majority of errors come from our database images, which contain very bright (dense) zones that result from overlapping ribs and/or pleural effusion. The results reveal that there is no confusion between the chest tubes and other artificial objects. The JRST database results do not have a False Positive or False Negative except for only one image that had 347 False Positive pixels. These error values show that the results are well aligned with the input images and are acceptable as an output. Therefore, we obtained promising results on the detection of artificial objects.

# KATMANLI SİNİR AĞLARI KULLANILARAK GÖĞÜS RADYOGRAFİLERİNDE GÖĞÜS TÜPÜ TESPİTİ İÇİN BİR YAKLAŞIM

## ÖZET

Dünya Sağlık Örgütü (WHO), 2030 yılında dünyada 2.279.045 kişinin akciğer kanserinden öleceğini tahmin etmektedir. Kanser en yaygın ikinci ölüm nedenidir. Kanser ve başka pek çok hastalığın tespit ve takibinde radyolojik tetkikler gereklidir.

Geliştirilen onca yeni radyolojik tetkike rağmen Göğüs Radyografisi en yaygın kullanılan radyolojik tetkiktir. Özellikle tespit ve takip amaçlı olarak çok popülerdir. Çünkü ucuz, bilgi yoğun, ve hızlıdır. Her yerde bulunabilir ve düşük seviyede radyasyon içerir. Bu sebeplerle yakın gelecekte tahtını kaybetmeyecekmiş gibi görünmektedir.

Ne yazık ki aynı zamanda yorumlanması en zor radyolojik tetkiklerden biridir. Üst üste binen dokular, göğüs radyografilerinde çok karmaşık projeksiyonlar oluşturur. Üstelik, kateterler, göğüs tüpleri, kalp pili gibi insan yapımı nesnelere de bu radyografilerde görülebilir. Anormallik tespit eden algoritmaların bu nesnelere sebebiyle sorun yaşamaması önemlidir. Bu hususta çalışma eksikliği bulunduğu ve çalışma yapılmasına ihtiyaç olduğu literatürde de kabul edilmiştir.

Bu amaçla biz radyografilerde mevcut göğüs tüplerini tespit etmek için bir katmanlı sinir ağı (İng. Convolutional Neural Network) eğitmeyi öneriyoruz. Göğüs tüpünün iskeletini daha iyi tespit edebilmek için sinir ağının çıktısı üzerine otomatik olarak oturtulan nonuniform rational b-spline eğrisi kullanılmıştır.

Tüm çalışma, C++ dilinde geliştirdiğimiz yazılımlarla yapılmıştır. Bu yazılım bir kütüphane şeklinde tasarlanarak, değişik yöntemlerin denemesine müsade eden bir yapıda hazırlanmıştır. Yazılım geliştirmenin doğal bir parçası olan yazılımın doğruluğunun sınanması da yapılmış olup, bu amaçla kullanılan yöntemler de çalışma kapsamında açıklanmıştır.

Katmanlı sinir ağı, kedi gözünden ilham alınmış bir yöntemdir. En önemli üstünlüğü, örüntü tanıma (pattern recognition) yöntemlerinde çoğunlukla uygulanmak zorunda kalınan özellik çıkarımı (feature extraction) ve hassas hizalama/ölçeklendirme işlemleri gerektirmemesi, ağırlık paylaşımı yoluyla serbest parametre sayısını düşük tutabilen yapısıdır.

Özellik çıkarımı, gerçek hayat verilerinin çok boyutlu yapısı ile pek çok örüntü tanıma yönteminin baş edememesi sonucu gerekmektedir. Katmanlı sinir ağı ise bu soruna katlama (convolution) işlemi ve ağırlık paylaşımı yoluyla serbest parametreleri sınırlandırarak, yüksek boyutlu verinin oluşturduğu karmaşıklığı aşmaktadır. Katlama (convolution) işlemi aynı zamanda hassas hizalama ihtiyacını da ortadan kaldırmaktadır. Gerçek hayat verilerinin, doğal yaşam formlarının yapabildiği gibi sinir ağı algoritmaları tarafından da bir önışleme tabii tutulmadan kullanılabilmesine yönelik

çalışmalar, derin öğrenme (deep learning) başlığı altında aktif bir çalışma konusudur. Katmanlı sinir ağı modeli bu konuda başarılı ilk yöntemidir.

Bu çalışma, göğüs radyograflerinin akciğer bölgesindeki yabancı cisimleri otomatik olarak tespit etmeye yönelik ilk çalışmadır. Diğer otomatik tespit algoritmaları mediastinum üzerinde çalışmaktadır. Ayrıca tam otomatik olarak çalışması ve erişime açık bir göğüs radyografi bilgi bankası kullanılarak sınanmış olması, bu çalışmanın diğer üstünlükleridir.

Yaptığımız ön çalışmalar sonucunda Gradient Descent ve Cross-Entropy algoritmalarını sigmoid aktivasyon fonksiyonu ile kullanmanın en iyi performansı verdiğini gördük. Önerdiğimiz katmanlı sinir ağı modelimizde, girdi ve çıktı dahil 5 katman mevcuttur. Sırasıyla, her katmanda 2, 32, 32, 128, ve 1 düğüm mevcuttur. Bu katmanlar arasında, yine sırasıyla 32, 16, 128, ve 1 adet bağlantı mevcuttur. Öğrenme katsayısı olarak 0.1 değerini kullandık.

Görüleceği gibi girdi katmanında 2 düğüm mevcuttur. Bu iki düğüm, aynı radyografinin iki farklı ölçekteki görüntüsü ile beslenir. Bu ise modelin boyutunu çok büyütmeden girdi olarak alınan alanı artırmanın bir yolu olarak çok ölçekli girdi tercih edilmesi sebebiyledir. Her girdi düğümü, girdi olarak kullanılacak radyografinin 1000x1000 ve 250x250 boyutlarındaki görüntülerinden birinden aynı bölgeye ait 13x13 boyutlarında bir parçayı alır. Bu şekilde çok ölçekli girdi ile sinir ağı beslenmiş olur. Eğitim sırasında girdiler, rastgele radyograflerin rastgele kısımlarından seçilerek kullanıldı. Üç ara katmanda çıktı boyutları 5x5, 1x1 ve 1x1 şeklinde seçildi. Son katmanda 1x1 boyutlarında çıktı üreten tek bir düğüm mevcuttur. Bu şekilde modelin her çalıştırılması sonuç görüntününün bir noktasını üretir.

Sinir ağının çıktısında göğüs tüpünün opak çizgisine karşılık gelen bir eğri oluşması amaçlanmaktadır. Ancak çoğu zaman bu eğri kesintisiz ve gürültüsüz elde edilemediğinden, çıktı resmi üzerine otomatik olarak nonuniform rational b-spline eğrisi oturtularak, kesintisiz ve gürültüsüz bir eğri elde edilmesi amaçlanmaktadır. Bu eğri oturtma işlemi için gereken kontrol noktaları, mevcut çıktı üzerindeki en belirgin nokta üzerinde ilerlenirken rastlanılan küçük boşlukların eğrinin istikameti yönünde atlanarak geçilmesi ile seçilmektedir. Bu şekilde aynı zamanda eğri üzerinde olmayan gürültülerden de kurtulunmuş olunur. Kontrol noktalarının sayısını sınırlamak amacıyla, 36 noktada bir nokta seçilmektedir.

Eğitim seti tamamen bizim veri tabanımızdaki görüntülerden oluşmaktadır ve 62 radyografi içermektedir. Bunlardan 13 tanesi çift, 25 tanesi tek göğüs tüpü içerir. 24 tanesi ise hiç göğüs tüpü içermemektedir. Sonuçları sınamak içinse, iki farklı veri tabanı kullanıldı. İlk veri tabanı bizim kendi veri tabanımızdır ve 21 radyografi içerir. Modelimizin diğer yapay cisimlerle göğüs tüpünü birbirinden ayırabilmesinin sağlanması için eğitim setinde 36 tane, sınama setinde 13 tane radyografi, göğüs tüpü dışındaki diğer yapay cisimlerden içerecek şekilde seçilmiştir. İkinci sınama seti ise Standard Digital Image Database Project Team of the Scientific Committee of the Japanese Society of Radiological Technology (JRST) tarafından oluşturulmuş ve herkese açık olan 247 radyografiden oluşan JRST veri setidir. Bu set yapay cisimler içerse de göğüs tüpü içermez.

Modelin performansını piksel tabanlı ROC analizi ile test ettik. Herbir gerçek pozitif, gerçek negatif, hatalı pozitif, hatalı negatif sayı ve ortalama accuracy, sensitivity, specificity, yüzdelere hesaplamak için kullanıldı. Sonuç olarak % 99.99 doğruluk



(accuracy), % 59 hassasiyet(sensitivity), % 99.99 özgünlük(specificity) değerleri elde edildi.

Sonuçlarımızdaki hataların çoğu üst üste binmiş kaburgalar yada plevral efüzyon sebebiyle oluşan son derece yoğun (radyografide açık renkte olan) bölgelerde oluşmaktadır. Testlerde yapay cisimler, göğüs tüpü ile hiç karıştırılmamıştır. JRST veri setinde oluşan tüm hatalar tek bir görüntüde oluşan 347 hatalı pozitif noktadan ibarettir. Hiç hatalı negatif nokta oluşmamıştır. Testler, tüm sonuç görüntülerinin girdi radyografileri ile aynı hizalanmıştır. Elde edilen bu sonuçlara göre, yabancı cisim tespitinde önerilen modelin kullanımı umut vadetmektedir.



## **1. INTRODUCTION**

### **1.1 Problem Definition**

#### **1.1.1 The Lung Cancer and Lung Related Deaths**

Worldwide, 1.549.121 new lung cancer cases and 1.351.034 lung cancer deaths are estimated in 2007 [1]. The cancer is the second leading cause of death. Lower respiratory infections, chronic obstructive pulmonary disease, and tuberculosis are other lung related causes in top 10 death causes [1]. Also, World Health Organization (WHO) estimates 2.279.045 lung cancer deaths in 2030 (Baseline scenario, [2]).

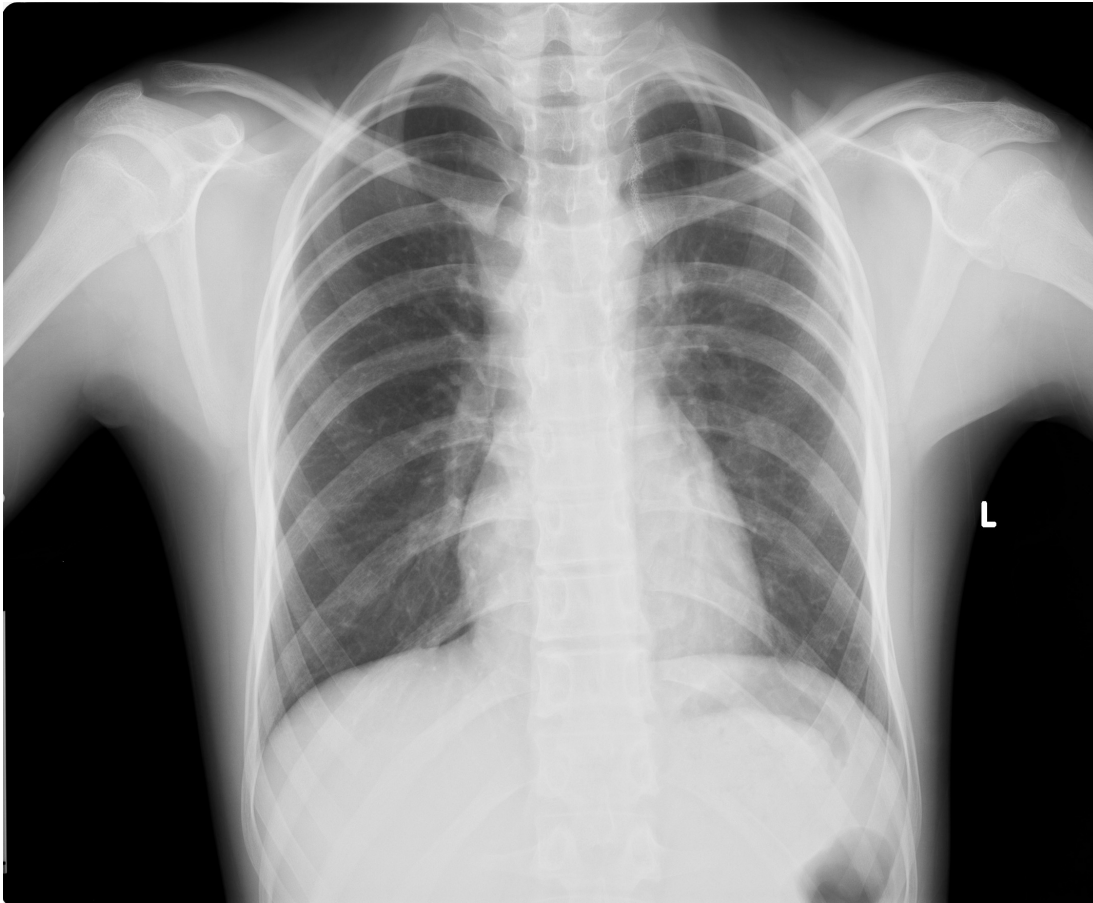
Beside, the high mortality rate of lung cancer, there is another problem: the high miss rate for detection of lung cancer nodules. This is one of most frequent reasons for malpractice lawsuits against radiologist [3].

#### **1.1.2 The Chest Radiography**

For scanning of lung cancer and other lung related diseases, the chest radiography has been widely used. Even now, the chest radiography (Figure 1.1) is the most common radiological modality in the practice [4,5]. In short term, it seems that modern imaging techniques are not going to replace the chest radiography [5]. Because, it is fast, dense, cheap, and accessible than most of the modern imaging techniques such as Magnetic Resonance (MR) or Computed Tomography (CT) [6]. Also, the chest radiography is using less radiation than some of the modern imaging techniques like CT [6].

However, most important problem of these modern techniques is huge amount of data obtained [7]. CT and MR are volumetric methods. This means radiologists must investigate up to 1500 slices per patient [7] and up to 300 per thorax [8]. However, the chest radiography produces just a single image. Because of these reasons, the chest radiography is very popular for scanning and screening purposes. Even patients of the

intensive care unit are monitored by doctors who daily use the chest radiography [9]. On the average 236 chest radiographs are performed to 1000 patients per year [4].

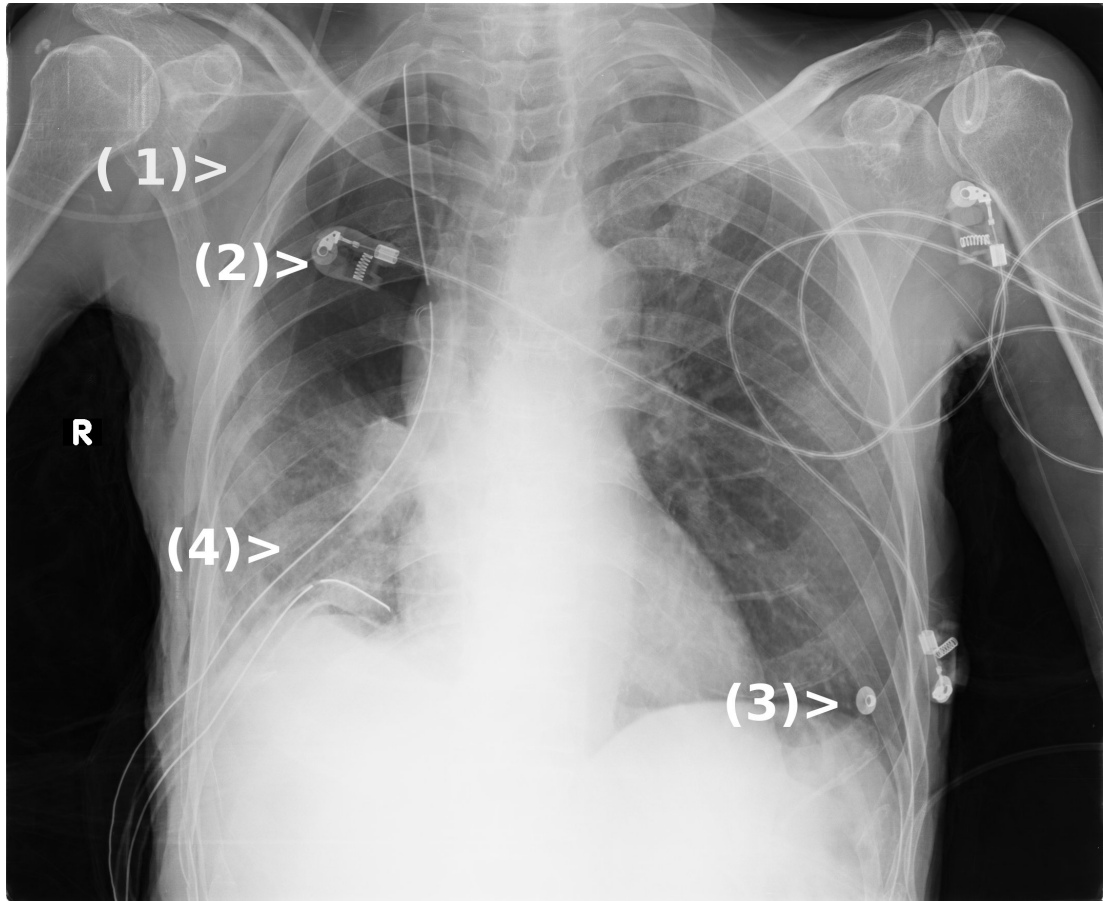


**Figure 1.1:** A normal chest radiography.

### **1.1.3 The Computerized Analysis of Radiographs**

The researchers are motivated by the high mortality and high miss rates to find an easy and dependable way of the detection of lung cancer. A possible solution might be computerized analysis of radiographs. The computerized analysis of radiographs for anomaly detection became an active research area with two main branches. First of all, Computer Aided Diagnosis (CAD) model, gives suggestions to the doctors, is an important and active research area. Moreover, the importance of CAD is increased with the widespread use of a picture archiving and communication system (PACS) [10]. Automated Diagnosis (AD), is the other branch, and more difficult than CAD [10]. However, both of them need robust anomaly detection schemes.

A major part of this research effort is focusing on the chest radiography, because it is the most common radiological modality in the practice. Especially, it is very popular

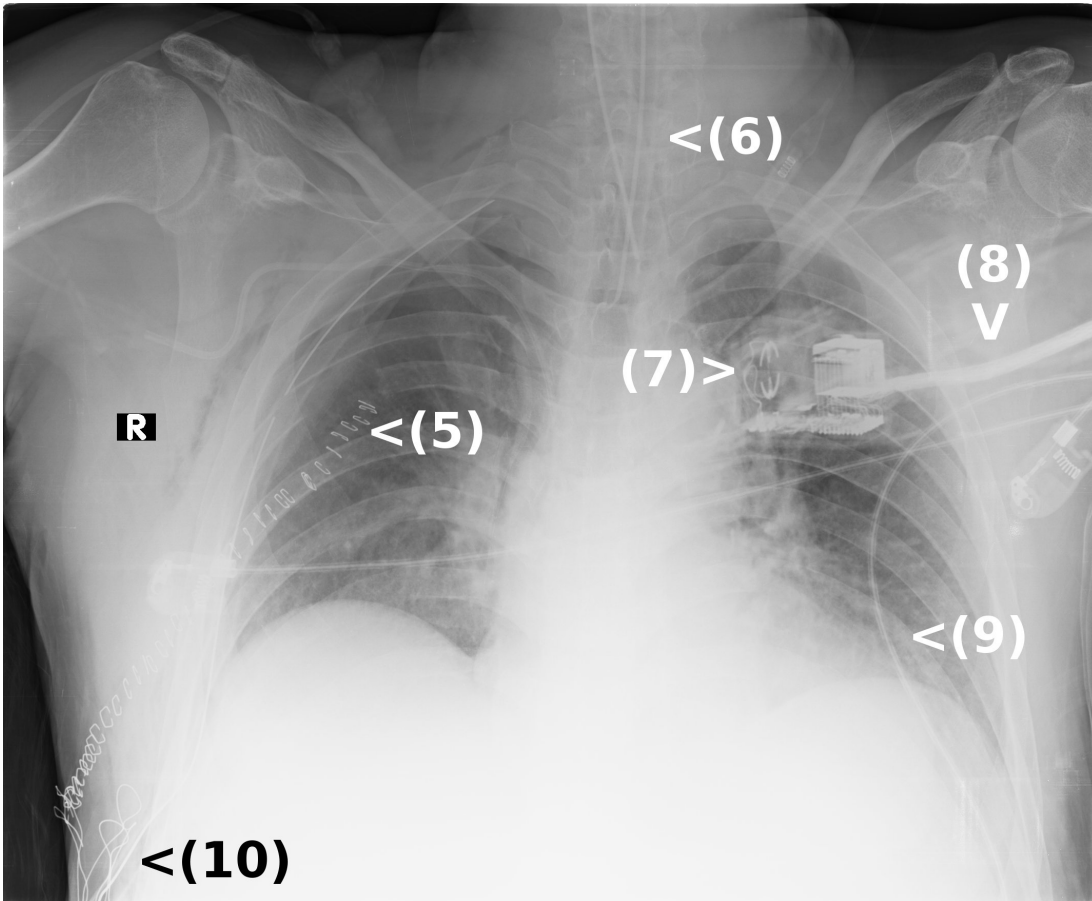


**Figure 1.2:** An example for artificial object at the chest radiography. 1) Oxygen cannula. 2) Connected EKG electrode. 3) Non-connected EKG electrode. 4) Chest tube.

for scanning purposes. Unfortunately, at the same time it is one of the most difficult radiological modalities [11, 12]. The overlapping tissues cause a highly complex projection. In addition, artificial objects such as catheters, chest tube, pacemaker, and/or even cloths might be presented at this projection image (Figures 1.2 and 1.3).

#### **1.1.4 The Artificial Object Problem**

Clinical practice shows that the presence of an artificial object in radiography is common and creates further complexities. The abnormal findings in chest radiography are studied by MacMahon et al. [13]. They report that the second common abnormal finding is catheter (33% of chest radiographs contains a catheter) [13]. In our investigations, we found that chest radiography databases also contain artificial objects. For example, The Japanese Society of Radiological Technology (JSRT) chest radiography database with and without chest lung nodules [14], contains 247 images. Four of them (JPCLN140, JPCLN147, JPCNN044, JPCNN083) have at least

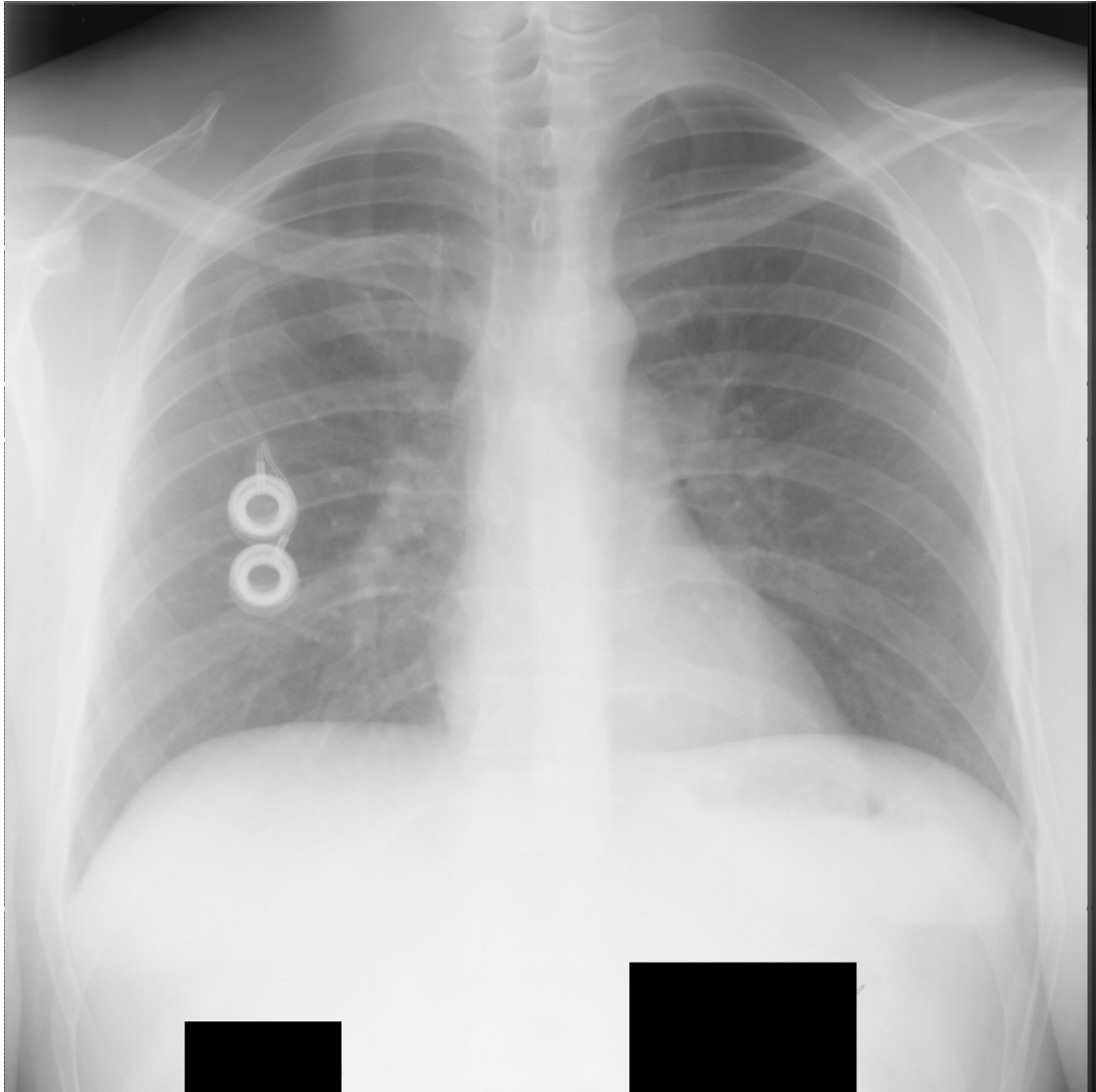


**Figure 1.3:** Another example for artificial object at the chest radiography. 5) Stapler. 6) Endotracheal tube. 7) Some electronic device. 8, 9, 10) Metal wires.

an artificial object (Figure 1.4). Also, The Reference Image Database to Evaluate Response (RIDER) Pilot Project database, contains 17 chest radiographs [15]. Except only one image, all images contain at least one artificial object. These databases are consisting of selected images, natural frequency might be different.

When a researcher attempts to develop a robust CAD approach that works for chest radiography, he or she must be certain that each and every algorithm works properly for all chest radiographs containing foreign objects. Therefore, the anomaly detection algorithm should not be confused by artificial objects.

There are only few papers known discussing the presence of artificial objects. Kuhlman et al. mention about artificial objects such as pacemaker might cause artifacts in the dual-energy subtraction chest radiography [16]. Another study reports that 20% percent (23 of 120) of the test data of the study is selected from chest radiographs which contains artificial object such as electrocardiographic wire and central venous catheter [17]. However, it has been not discussed about the correlation between



**Figure 1.4:** A chest radiography that contains artificial object from the JSRT database. artificial objects and detection errors. Arzhaeva et al. [18] report that their data set selected by discarding images, which contain any artificial object.

Many techniques used in CAD papers are possibly susceptible to artificial objects. For example; both temporal and contralateral subtraction techniques used for suppressing unchanged part of radiography. These techniques assume image differences show abnormal progress of a disease [5]. However, if only one of radiographs, which was subtracted, contains an artificial object, this assumption will be violated. Also, contour fitting algorithms that used finding segmentation borders such as rib segmentation, might be failed. Because the literature does not mention a possible artificial object weakness, it is not possible to list the definitely sensitive algorithms. However, any CAD solution is a collection of different algorithms, and if it is wanted to develop a CAD solution that robustly works for chest radiography, it should be found a way

to be sure that every algorithm is safe about overlapping foreign objects. Hence, the detecting and even better than detecting, the removing foreign object as a preprocessing module is very meaningful for CAD research.

### **1.1.5 Chest Tube**

Fortunately, the majority of medical objects such as catheters and chest tubes (Artificial Object nr. 4, Figure 1.2) are long, thin, mostly constant in shape, and not totally opaque. These features make them easily detectable by the radiologist.

Despite these helpful features, automatic detection of the chest tube figure is very difficult. The reason is the different parts of chest tube show different characteristics and also having discontinuities (Figure 1.2). Firstly, the head of the chest tube is not identical to the body of the tube. After slightly rounded head, there is also a discontinuity at opaque line. Even the body, itself, might be a source of the problem because of that the crosscut of tube contains opaque part and non-opaque parts. Unfortunately, according to the crosscut of tube, the position and width of opaque line is different. Finally, it must be mentioned the width of chest tube is a variable in the practice.

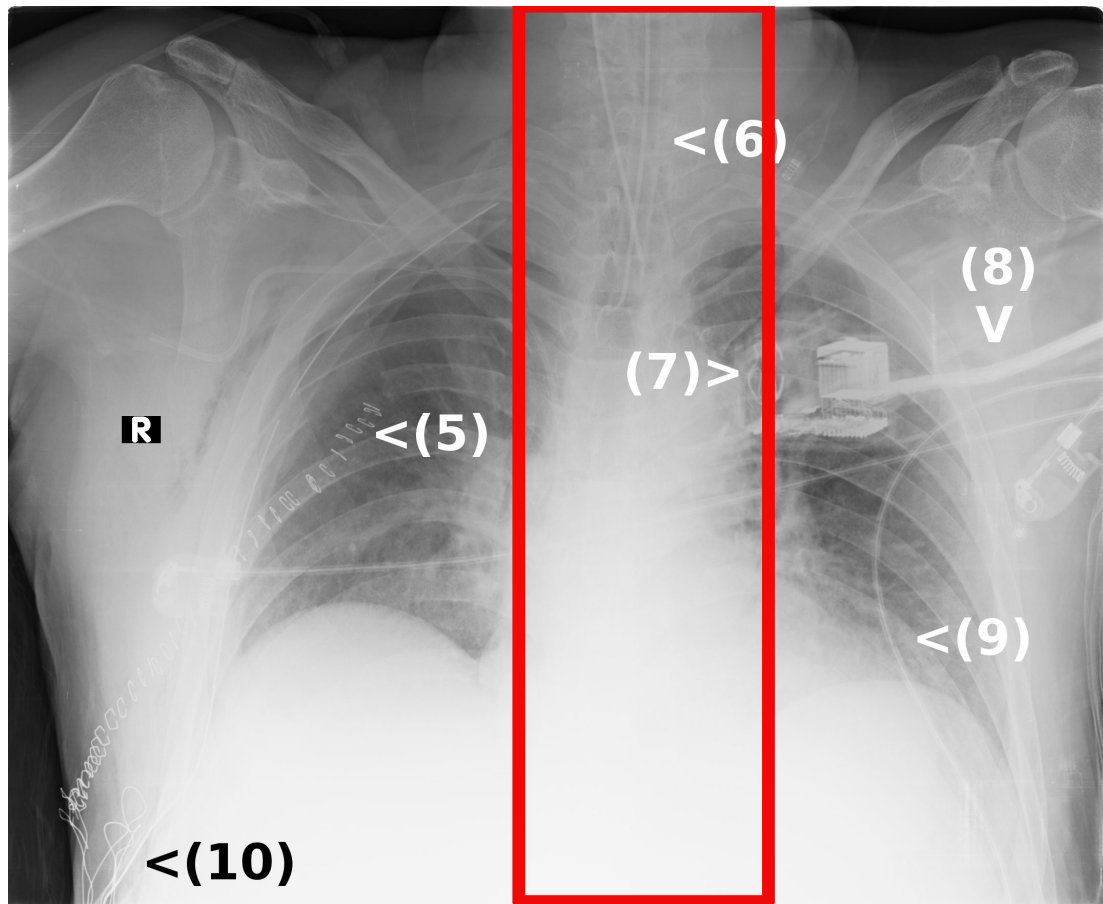
## **1.2 Literature Review**

Detecting foreign objects is a critical issue for CAD research. However, a survey by Van Ginneken et al. [5] reported that the detection of artificial objects is one of the unsolved problems of CAD. Only a few studies in the CAD literature have focused on the detection of artificial objects in chest radiography.

Four prominent studies can be highlighted. First, a semi-automated method for tracking the location of naso-gastric tubes, endo-tracheal tubes, chest tubes, PICC and central venous catheters using five chest radiographs was proposed by Keller et al. [19]. This method requires two seed points specified by user. Second and third studies have investigated the automatic detection of tubes that are located only in the mediastinum. A method for automatic detection and positioning of endotracheal, feeding and nasogastric tubes using 107 chest radiographs was studied by Sheng et al. [20]. Additionally, Ramakrishna et al. [21, 22], worked on the automatic detection



of endotracheal and nasogastric tubes. All of these automatic methods work for objects located in the mediastinum only (Figure 1.5).



**Figure 1.5:** Mediastinum Region of the Chest.

Fourth, the detection and removal of simulated chest tubes from radiographs were also studied in our previous work [23].

### **1.3 Aim of the dissertation**

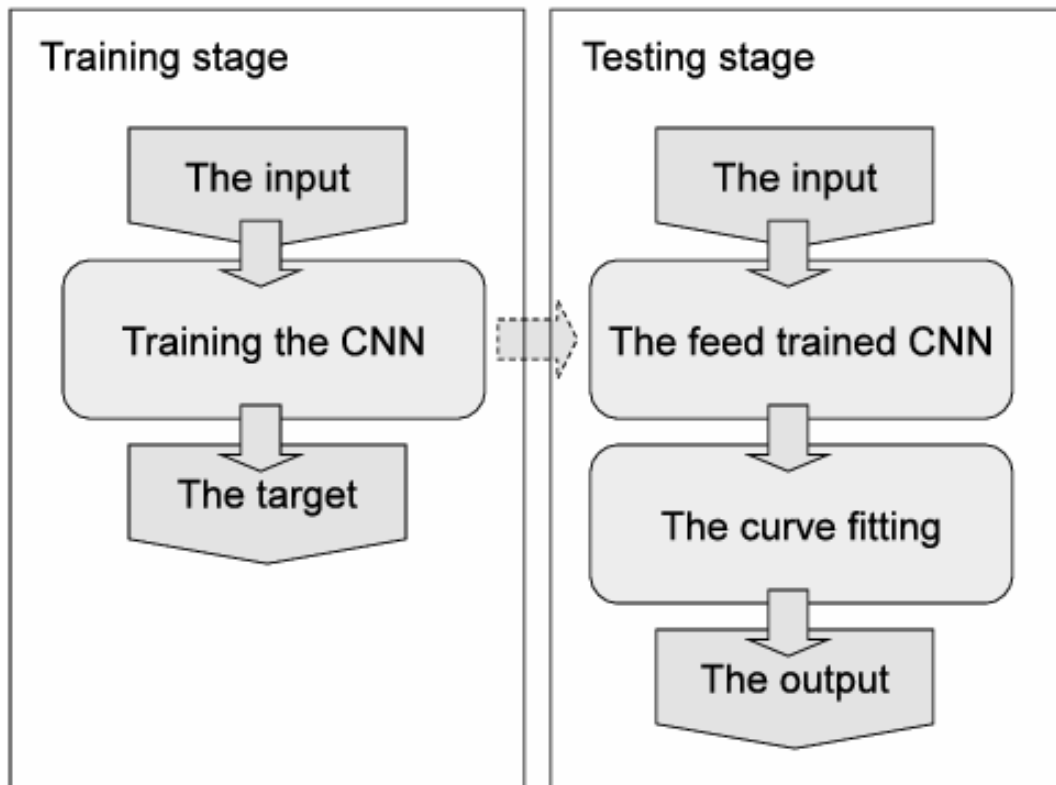
In this dissertation, we proposed a scheme basically focusing on detecting a chest tube figure (as an example of an artificial object might be presented at the radiography) from Postero-Anterior (PA) Chest Radiography.

A Convolutional Neural Network (CNN) that takes the chest tube containing x-ray image as an input and gives an artificial image with a chest tube skeleton as an output is studied. The trained CNN is used together with non-uniform rational B-splines (NURBS) to automatically detect the presence of chest tubes in chest radiographs (Figure 1.6).

Although there are some medical image processing and CAD that work on chest radiographs and use a CNN such as segmentation of bones [24], and detection of cancer on chest radiographs [25, 26], there are any artificial object detection schemes which use a CNN, in the literature.

Moreover, our proposed model is the first study conducted to automatically detect artificial objects in the lung region of chest radiographs. Other automatic detection schemes work on the mediastinum.

In our research, we noted that in some cases, the CNN output of the form of the detected chest tube skeleton was not clear and had short and long discontinuities. To overcome this problem, we introduced a NURBS-based curve fitting algorithm that was applied to the detected chest tube image to obtain a better chest tube skeleton as a final output.



**Figure 1.6:** Simplified Flowchart of the proposed system.

It has chosen the CNN method as a Neural Network (NN) model, due to the shifting, scaling, and distortion invariance abilities, and weight sharing property for reducing number of free parameters of neural network [27]. It is inspired by a modified version of the vision system in vertebrate animals and it simulates all image processing, feature extraction, and pattern recognition stages for working on images. CNN takes pictures

as input, without a need for a feature extraction stage. Therefore, in order to detect the chest tube figure, we have decided to use the CNN method trained with chest images containing chest tubes as an input and chest tube skeleton as a target (Figure 1.6).



## **2. CONVOLUTIONAL NEURAL NETWORK (CNN)**

The CNN is a neural network method inspired by biological visual systems [28] and it can handle some of image processing problems internally such as shift, scale, and distortion variations and it reduces number of free parameters of neural network using shared weights. It uses matrices, instead of scalars that are basically used by traditional NN method (Figure 2.1). The main advantage of using matrices is that they work as local receptive fields and protect local spatial neighborhoods. It allows to us simulate some image processing, feature extraction and pattern recognition stages of the classic scheme of the pattern recognition case [27, 28].

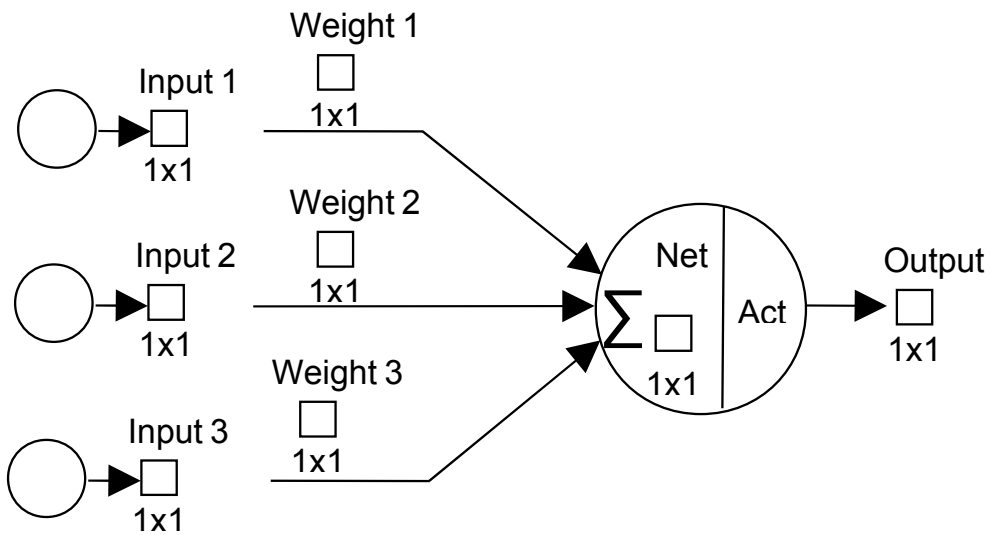
Although the CNN architecture is often used for pattern recognition purpose [11, 27, 29, 30], the work of Browne and Ghidary [28] is first and the only paper that CNN is used for image processing, except for a few variants of CNN. Especially, for pattern recognition purpose, CNN papers show very successful results.

However, CNN is not a popular method. Simard et al. [29] argued that the main reason for that was the complexity of CNN. In addition, our experience also showed us that another potential reason was imperfect documentation of CNN.

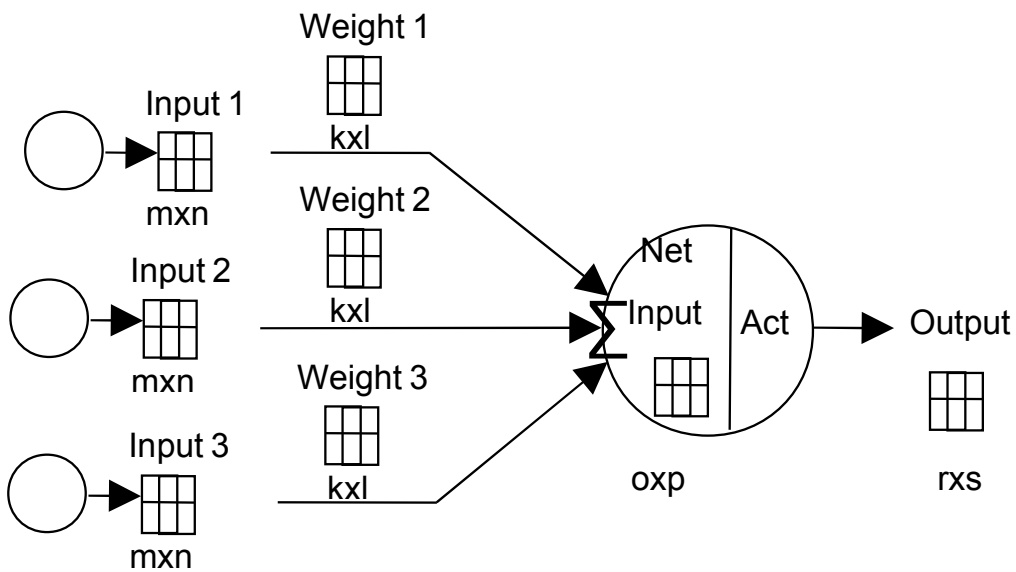
There is a group of research papers that contains small parts of the method with multiple variations [27, 29]. A nice alternative for these papers is a manuscript [28] that explains the simplified and unusable version of CNN formulation assuming to work with vectors (1D) but not with matrices (2D). After that, we focused obtaining formula by ourselves.

### **2.1 Obtaining Formula**

The training of a CNN works at two main steps: feedforward and backward steps. At the feedforward step, NN calculates outputs from inputs. Backward step is the optimization of free parameters of the model.

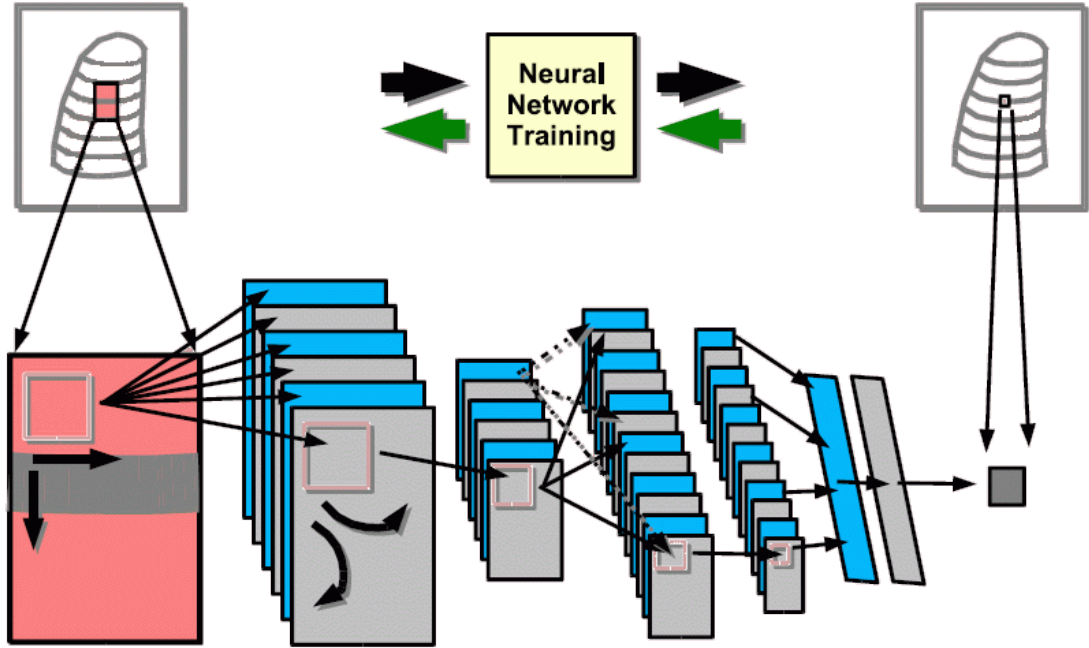


(a) The Traditional Node: Scalar Inputs are multiplied with scalar weights. Whole parameters are scalar numbers.



(b) The CNN Node: The Convolution operation works on Matrix Inputs with the Matrix Weights. Whole parameters are matrices.

**Figure 2.1:** The Difference Between Traditional Neural Network Node and CNN Node.



**Figure 2.2:** Details of a CNN architecture for image processing. The input frame has been taken from a picture. Then, the result is going to be copied to the output picture.

A CNN requires training with many image pairs. An image pair is composed of a chest image, which contains a chest tube as an input, and a map image, which contains a line representing the opaque line of a chest tube as the target. After the training stage for a given x-ray image that contains a chest tube, the CNN output will be a map image containing the chest tube.

### 2.1.1 Feedforward Step

Let define a CNN architecture that contains  $L$  number of layers. There are three types of layers according to layer order: input layer ( $l = 1$ ), output layer ( $l = L$ ), and hidden layers ( $1 < l < L$ ). The input layer is an abstract layer that only contains the input data without any calculations. Hidden and Output layers differing in calculations will be explained at following sections.

#### 2.1.1.1 Feedforward step of the output layer

Let  $net_k$  denote the  $k$ th output image of the output layer of our neural network that contains  $L$  layer (in other words, output of the  $k$ th node of output layer).  $O_j$  denote the  $j$ th image group which is the output of the  $j$ th node at layer  $L - 1$ , and let  $J$  be the

number of such groups.

$$net_k = b_k + \sum_j^J O_j ** W_{kj} \quad (2.1)$$

where  $**$  denotes the two-dimensional (2D) convolution (Figure 2.3) and  $W_{kj}$  denotes the kernel of weights connecting the  $j$ th group in the  $(L-1)$ th layer to the  $k$ th group in the  $(L)$ th layer. Also  $b_k$  shows the bias value of the  $k$ th node.

The width  $S$  of the weight kernel  $W_{kj}$ , the step size  $m$  of the kernel at horizontal direction, and the width  $H^{(jw)}$  of the output image  $O_j$  of layer  $L-1$  defines the width  $H^{(kw)}$  of the output image  $O_k$  of layer  $L$ :

$$H^{(kw)} = \frac{H^{(jw)} - S + m}{m} \quad (2.2)$$

also, the same equation is valid for the height:

$$H^{(kh)} = \frac{H^{(jh)} - T + n}{n} \quad (2.3)$$

We try to separate each entry of the matrix  $net_k$ :  $net_k^{kw,kh}$  means the element at column  $kw$  th and row  $kh$  th of the image  $net_k$ .

$$net_k^{kw,kh} = b_k + \sum_j^J \sum_s^S \sum_t^T W_{kj}^{s,t} O_j^{\overbrace{((kw*m)+s)}^{jw}, \overbrace{((kh*n)+t)}^{jh}} \quad (2.4)$$

Then,  $net_k$  used for calculating  $O_k$ :

$$O_k^{kw,kh} = f(net_k^{kw,kh}), \quad k = 1, \dots, K \quad (2.5)$$

The  $f()$  is the transfer function that we select.

### 2.1.1.2 Feedforward step of hidden layers

Nearly, same equations are valid for hidden layers with little differences. Let  $net_j$  denote the  $j$ th output image of the  $l$ th layer of our neural network that contains  $L$  layers (in other words, output of the  $j$ th node of  $l$ th layer).  $O_i$  denote the  $i$ th image group which is the output of the  $i$ th node at layer  $l-1$ , and let  $I$  be the number of such groups.

$$net_j = b_j + \sum_i^I O_i ** W_{ji} \quad (2.6)$$

$$net_j^{jw,jh} = b_j + \sum_i^I \sum_u^U \sum_v^V W_{ji}^{u,v} O_i^{\overbrace{((jw*m')+u)}^{iw}, \overbrace{((jh*n')+v)}^{ih}} \quad (2.7)$$



## Convolution Operation

Input Matrix

i	i	i	i	i
i	i	i	i	i
i	i	i	i	i
i	i	i	i	i
i	i	i	i	i

Kernel Matrix

w	w	w
w	w	w
w	w	w

\*\*

=

Output Matrix

out	out	out
out	out	out
out	out	out

i×w	i×w	i×w	i	i
i×w	i×w	i×w	i	i
i×w	i×w	i×w	i	i
i	i	i	i	i
i	i	i	i	i

Start with first element of Output

w	w	w
w	w	w
w	w	w

\*\*

=

Σ(i×w)		

i	i×w	i×w	i×w	i
i	i×w	i×w	i×w	i
i	i×w	i×w	i×w	i
i	i	i	i	i
i	i	i	i	i

Continue with moving a step size

w	w	w
w	w	w
w	w	w

\*\*

=

Σ(i×w)	Σ(i×w)	

**Figure 2.3:** Details of the convolution operation. A block of input is multiplied element wise with the kernel matrix. Then, the result is going to be stored as an element of the output matrix.

$$O_j^{jw,jh} = f(\text{net}_j^{jw,jh}), \quad j = 1, \dots, J \quad (2.8)$$

The width  $U$  of the weight kernel  $W_{ji}$ , the step size  $m'$  of the kernel at horizontal direction and the width  $H^{(iw)}$  of the output image  $O_i$  of layer  $l - 1$  defines the width  $H^{(jw)}$  of the output image  $O_j$  of layer  $l$ :

$$H^{(jw)} = \frac{H^{(iw)} - U + m'}{m'} \quad (2.9)$$

also, same equation is valid for the height:

$$H^{(jh)} = \frac{H^{(ih)} - V + n'}{n'} \quad (2.10)$$

### 2.1.2 Backpropagation

The training of a neural network is an optimization process of weights of the neurons according to an error formula of the neural network. Our error function is:

$$E = \frac{1}{2} \sum_k^K \sum_{kw}^{H^{(kw)}} \sum_{kh}^{H^{(kh)}} (t_k^{kw,kh} - O_k^{kw,kh})^2 \quad (2.11)$$

and total node count  $K$  in output layer, width  $H^{(kw)}$  and height  $H^{(kh)}$  sizes of output nodes define single output count, and each output  $O_k^{kw,kh}$  produces a local error, according to itself target value  $t_k^{kw,kh}$ .

Instead of the specific solution per neural network, the backpropagation method obtains a general solution to update weights per layer [31]. From output layer to first hidden layer, calculations of each layer depend on previous calculations of next layers.

#### 2.1.2.1 Backpropagation step of the output layer

First step of the backpropagation is calculating weight update for nodes of the output layer:

$$\Delta W_{kj}^{s,t} = -\eta \frac{\partial E}{\partial W_{kj}^{s,t}} \quad (2.12)$$

that will be used for new weights. Delta rule for gradient decent method can be given as an example:

$$W_{kj}^{s,t(\text{new})} = W_{kj}^{s,t(\text{old})} - \eta \frac{\partial E}{\partial W_{kj}^{s,t}} \quad (2.13)$$

Then, we will convert formula from matrix derivative form to entry derivative version [32]:

$$\frac{\partial E}{\partial W_{kj}^{s,t}} = \frac{\partial E}{\partial net_k} \frac{\partial net_k}{\partial W_{kj}^{s,t}} \quad (2.14)$$

$$= \sum_{kw}^{H(kw)} \sum_{kh}^{H(kh)} \underbrace{\frac{\partial E}{\partial net_k} \frac{net_k^{kw,kh}}{W_{kj}^{s,t}}}_{-\delta_k^{kw,kh}} \quad (2.15)$$

$$\frac{\partial net_k^{kw,kh}}{\partial W_{kj}^{s,t}} = \frac{\partial}{\partial W_{kj}^{s,t}} (b_k + \sum_{j'}^J \sum_{s'}^S \sum_{t'}^T W_{kj}^{s,t} O_j^{((kw*m)+s'),((kh*n)+t')}) \quad (2.16)$$

$$= O_j^{((kw*m)+s),((kh*n)+t)}, \quad (j = j', s = s', t = t'). \quad (2.17)$$

$$\delta_k^{kw,kh} = -\frac{\partial E}{\partial net_k^{kw,kh}} \quad (2.18)$$

$$= -\frac{\partial E}{\partial O_k^{kw,kh}} \frac{\partial O_k^{kw,kh}}{\partial net_k^{kw,kh}} \quad (2.19)$$

$$= - (t_k^{kw,kh} - O_k^{kw,kh}) f'(net_k^{kw,kh}) \quad (2.20)$$

$$\delta_k^{kw,kh} = (t_k^{kw,kh} - O_k^{kw,kh}) f'(net_k^{kw,kh}) \quad (2.21)$$

$$\Delta W_{kj}^{s,t} = \eta \sum_{kw}^{H(kw)} \sum_{kh}^{H(kh)} \delta_k^{kw,kh} O_j^{((kw*m)+s),((kh*n)+t)} \quad (2.22)$$

### 2.1.2.2 Backpropagation step of hidden layers

Most imported part of backpropagation method is obtained at this section:

$$\Delta W_{ji}^{u,v} = -\eta \frac{\partial E}{\partial W_{ji}^{u,v}} \quad (2.23)$$

$$\frac{\partial E}{\partial W_{ji}^{u,v}} = \frac{\partial E}{\partial net_j} \frac{\partial net_j}{\partial W_{ji}^{u,v}} \quad (2.24)$$

$$= \sum_{jw}^{H(jw)} \sum_{jh}^{H(jh)} \underbrace{\frac{\partial E}{\partial net_j} \frac{net_j^{jw,jh}}{W_{ji}^{u,v}}}_{-\delta_j^{jw,jh}} \quad (2.25)$$

Equation 2.25 is the most important part of backpropagation method. The  $\frac{\partial E}{\partial net_j^{jw,jh}} = -\delta_j^{jw,jh}$  part of formula links the calculation of derivative of hidden layers to the calculation of derivative of next layer. The good think is that the  $\delta_j^{jw,jh}$  term makes that backpropagation formulation is general. With using this term, we can calculate any derivatives of NN, layer by layer, without the need of re-obtaining formulas.

$$\frac{\partial net_j^{jw,jh}}{\partial W_{ji}^{u,v}} = \frac{\partial}{\partial W_{ji}^{u,v}} \left( b_k + \sum_{i'}^I \sum_{u'}^U \sum_{v'}^V W_{ji}^{u,v} O_i^{((jw*m') + u'), ((jh*n') + v')} \right) \quad (2.26)$$

$$= O_i^{((jw*m') + u), ((jh*n') + v)}, \quad (i = i', u = u', v = v'). \quad (2.27)$$

$$\delta_j^{jw,jh} = -\frac{\partial E}{\partial net_j^{jw,jh}} \quad (2.28)$$

$$= -\frac{\partial E}{\partial O_j^{jw,jh}} \frac{\partial O_j^{jw,jh}}{\partial net_j^{jw,jh}} \quad (2.29)$$

$$-\frac{\partial E}{\partial O_j^{jw,jh}} = \sum_k^K -\frac{\partial E}{\partial net_k} \frac{\partial net_k}{\partial O_j^{jw,jh}} \quad (2.30)$$

$$= \sum_k^K \sum_{kw}^{H(kw)} \sum_{kh}^{H(kh)} \underbrace{\frac{-\partial E}{net_k^{kw,kh}}}_{\delta_k^{kw,kh}} \frac{\partial net_k^{kw,kh}}{\partial O_j^{jw,jh}} \quad (2.31)$$

$$\frac{\partial net_k^{kw,kh}}{\partial O_j^{jw,jh}} = \frac{\partial}{\partial O_j^{jw,jh}} \left( \sum_{j'}^J \sum_s^S \sum_t^T W_{kj}^{s,t} O_{j'}^{\overbrace{((kw*m') + s)}^{jw'}, \overbrace{((kh*n') + t)}^{jh'}} + b_k \right) \quad (2.32)$$

$$= W_{kj}^{(jw - (kw*m'), (jh - (kh*n'))}, (j = j', jw = jw', jh = jh'). \quad (2.33)$$

$$\frac{O_j^{jw,jh}}{net_j^{jw,jh}} = f'(net_j^{jw,jh}) \quad (2.34)$$

$$\delta_j^{jw,jh} = \left( \sum_k^K \sum_{kw}^{H(kw)} \sum_{kh}^{H(kh)} \delta_k^{kw,kh} W_{kj}^{(jw - (kw*m'), (jh - (kh*n'))} \right) f'(net_j^{jw,jh}) \quad (2.35)$$

$$\Delta W_{ji}^{u,v} = \eta \sum_{jw}^{H(jw)} \sum_{jh}^{H(jh)} \delta_j^{jw,jh} O_i^{((jw*m')+u),((jh*n')+v)} \quad (2.36)$$

### 2.1.3 Subsampling

The subsampling is another important method of the CNN. It is mainly an averaging operation with an only single weight parameter. The subsampling ensures that the rotating and shifting invariance to NN, because the output of the node is not dependent on the input location. Also, it reduces the parameter count dramatically, because of the using single weight. The formulation of the subsampling is given as:

$$net_k^{kw,kh} = b_k + w_k \sum_j^J \sum_s^S \sum_t^T O_j^{\overbrace{((kw*m)+s)}^{jw}, \overbrace{((kh*n)+t)}^{jh}} \quad (2.37)$$

If we remember the convolution formula:

$$net_k^{kw,kh} = b_k + \sum_j^J \sum_s^S \sum_t^T W_{kj}^{s,t} O_j^{\overbrace{((kw*m)+s)}^{jw}, \overbrace{((kh*n)+t)}^{jh}} \quad (2.38)$$

we can observe that the convolution step intrinsically contains subsampling operation.

#### 2.1.3.1 Backpropagation step of the subsampling method at the output layer

First step of the backpropagation is to calculate the weight update for nodes of the output layer:

$$\Delta w_k = -\eta \frac{\partial E}{\partial w_k} \quad (2.39)$$

that will be used for new weights. For example; delta rule for the gradient decent method is:

$$w_k^{(new)} = w_k^{(old)} - \eta \frac{\partial E}{\partial w_k} \quad (2.40)$$

Then, we will convert it's from matrix derivative to entry derivative version [32]:

$$\frac{\partial E}{\partial w_k} = \frac{\partial E}{\partial net_k} \frac{\partial net_k}{\partial w_k} \quad (2.41)$$

$$= \sum_{kw}^{H(kw)} \sum_{kh}^{H(kh)} \underbrace{\frac{\partial E}{\partial net_k^{kw,kh}}}_{-\delta_k^{kw,kh}} \frac{net_k^{kw,kh}}{w_k} \quad (2.42)$$

$$\frac{\partial net_k^{kw,kh}}{\partial w_k} = \frac{\partial}{\partial w_k} (b_k + w_k \sum_{j'}^J \sum_{s'}^S \sum_{t'}^T O_j^{((kw*m)+s'),((kh*n)+t')}) \quad (2.43)$$

$$= \sum_{j'}^J \sum_{s'}^S \sum_{t'}^T O_j^{((kw*m)+s'),((kh*n)+t')}, \quad (j = j', s = s', t = t') \quad (2.44)$$

$$\delta_k^{kw,kh} = -\frac{\partial E}{\partial net_k^{kw,kh}} \quad (2.45)$$

$$= -\frac{\partial E}{\partial O_k^{kw,kh}} \frac{\partial O_k^{kw,kh}}{\partial net_k^{kw,kh}} \quad (2.46)$$

$$= -\left(-\left(t_k^{kw,kh} - O_k^{kw,kh}\right)\right) f'(net_k^{kw,kh}) \quad (2.47)$$

$$\delta_k^{kw,kh} = \left(t_k^{kw,kh} - O_k^{kw,kh}\right) f'(net_k^{kw,kh}) \quad (2.48)$$

$$\Delta w_k = \eta \sum_{kw}^{H(kw)} \sum_{kh}^{H(kh)} \delta_k^{kw,kh} \sum_{j'}^J \sum_{s'}^S \sum_{t'}^T O_j^{((kw*m)+s'),((kh*n)+t')} \quad (2.49)$$

### 2.1.3.2 Backpropagation step of the subsampling method at hidden layers

$$\Delta w_j = -\eta \frac{\partial E}{\partial w_j} \quad (2.50)$$

$$\frac{\partial E}{\partial w_j} = \frac{\partial E}{\partial net_j} \frac{\partial net_j}{\partial w_j} \quad (2.51)$$

$$= \sum_{jw}^{H(jw)} \sum_{jh}^{H(jh)} \underbrace{\frac{\partial E}{\partial net_j^{jw,jh}}}_{-\delta_j^{jw,jh}} \frac{net_j^{jw,jh}}{w_j} \quad (2.52)$$

$$\frac{\partial net_j^{jw,jh}}{\partial w_j} = \frac{\partial}{\partial w_j} \left( b_k + w_j \sum_{i'}^I \sum_{u'}^U \sum_{v'}^V O_i^{((jw*m')+u'),((jh*n')+v')} \right) \quad (2.53)$$

$$= \sum_{i'}^I \sum_{u'}^U \sum_{v'}^V O_i^{((jw*m')+u'),((jh*n')+v')}, \quad (i = i', u = u', v = v') \quad (2.54)$$

$$\delta_j^{jw,jh} = -\frac{\partial E}{\partial net_j^{jw,jh}} \quad (2.55)$$

$$= -\frac{\partial E}{\partial O_j^{jw,jh}} \frac{\partial O_j^{jw,jh}}{\partial net_j^{jw,jh}} \quad (2.56)$$

$$-\frac{\partial E}{\partial O_j^{jw,jh}} = \sum_k^K -\frac{\partial E}{\partial net_k} \frac{\partial net_k}{\partial O_j^{jw,jh}} \quad (2.57)$$

$$= \sum_k^K \sum_{kw}^{H^{(kw)}} \sum_{kh}^{H^{(kh)}} \underbrace{\frac{-\partial E}{net_k^{kw,kh}}}_{\delta_k^{kw,kh}} \frac{\partial net_k^{kw,kh}}{\partial O_j^{jw,jh}} \quad (2.58)$$

$$\frac{\partial net_k^{kw,kh}}{\partial O_j^{jw,jh}} = \frac{\partial}{\partial O_j^{jw,jh}} \left( b_k + w_k \sum_{j'}^J \sum_s^S \sum_t^T O_{j'}^{\overbrace{((kw*m)+s)}^{jw'}, \overbrace{((kh*n)+t)}^{jh'}} \right) \quad (2.59)$$

$$= w_k. \quad (2.60)$$

$$\frac{O_j^{jw,jh}}{net_j^{jw,jh}} = f'(net_j^{jw,jh}) \quad (2.61)$$

$$\delta_j^{jw,jh} = \left( \sum_k^K \sum_{kw}^{H^{(kw)}} \sum_{kh}^{H^{(kh)}} \delta_k^{kw,kh} w_k \right) f'(net_j^{jw,jh}) \quad (2.62)$$

## 2.2 Custom Code for CNN

We developed a custom CNN library using C++ language to easily modify the architecture. As a result, deploying a CNN algorithm with custom input/output sizes and links is has been more practical.

```
int layerCount = 5;
// Defining layer count
base_ann *v1 = new base_ann(layerCount);

// Defining Architecture
v1->archFeedForwardSigmoidCE();

// Defining node count and node output sizes
v1->feedForwardLayer(0, lnode[0], 0, lnode[1],
                    Insize[0][0], Insize[0][1]);

v1->feedForwardLayer(1, lnode[1], lnode[0], lnode[2],
                    Insize[1][0], Insize[1][1], 2, 2);

v1->feedForwardLayer(2, lnode[2], lnode[1], lnode[3],
                    Insize[2][0], Insize[2][1], 2, 2);

v1->feedForwardLayer(3, lnode[3], lnode[2], lnode[4],
                    Insize[3][0], Insize[3][1]);

v1->feedForwardLayer(4, lnode[4], lnode[3], 0,
```

```

        lsize[4][0], lsize[4][1]);

// Setting initial weight values
vl->setKernelsAutoInit();

// Setting Learning Rate
vl->setLearningRate(eta, maxepoch);
vl->init();

// Link nodes
vl->layer(0).setFullLinkTo();
for (int z = 1; z < (layerCount - 2); z++) {
    vl->layer(z).setPerLinkTo(linkcount[z]);
}
vl->layer(layerCount - 2).setFullLinkTo();

```

Loading an input/target images set:

```

// Random Chest Radiography selection
// Getting first input image for first input node
inbtrain.getPic(0).init(inputlist[rasgele], irmin, irmax,
    lsize[0][0], lsize[0][1], 1, 1,
    (lsize[0][0]/2) * (olc - 1),
    (lsize[0][1] / 2) * (olc - 1), 1
    + ((lsize[0][0] / 2) * (olc - 1)), 1
    + ((lsize[0][1] / 2) * (olc - 1)), true);

// Getting second input image for second input node
inbtrain.getPic(1).init(inputlist2[rasgele], irmin, irmax,
    lsize[0][0], lsize[0][1], 1, 1, 0, 0, 0, 0,
    true, olc, olc);

// Getting target image for output node
tarbtrain.getPic(0).init(targetlist[rasgele], rmin, rmax, 1, 1, 1,
    1, (lsize[0][0] / 2) * olc, (lsize[0][1] / 2) * olc,
    (lsize[0][0] / 2) * olc, (lsize[0][1] / 2) * olc, true);

```

Training CNN for an input/target images set:

```

// Training using this random input
// scout = How many samples will be taken
// inbtrain = input images
// tarbtrain = target image

train_err =
    vl->trainLoopMergedRandErrorAround(scout,
    inbtrain, tarbtrain);

```

Testing CNN for an input/target images set:

```

// Testing with a radiograph
// "output.pgm" = output image name
// rmin = treshhold for 0 value at output image
// rmax = treshhold for 255 value at output image
// inbtest = input images
// tarbtest = target image
// outb = output image

```



```

test_err =
    vl->bigPictureRun("output.pgm", rmin, rmax,
        inbtest[0], tarbtest[0], outb);

```

Major functions used to construct a CNN are defined at base\_ann class (Appendix A).

### 2.3 Validation of the Formula and the Code

The NN is known as a very robust method, even though using buggy codes and a formula that contains discrepancies. It can be considered that this is the major advantage of NN method. However, from the software implementation perspective, this is not an advantage. It causes some serious difficulties at the debugging stage of the software. Observing calculations manually for big models such as CNN is very difficult.

In order to check the accuracy of the results, an alternative approach would be to use the another numerical differentiation techniques such as finite (central) differences that is very convenient and simple technique [33].

The derivatives of the perturbation weights, net sums (*net*), and inputs are evaluated with the finite difference formulation as follows:

$$\frac{\partial E}{\partial W_{kj}^{s,t}} = \frac{E(W_{kj}^{s,t} + \varepsilon) - E(W_{kj}^{s,t} - \varepsilon)}{2\varepsilon} + \mathcal{O}(\varepsilon^2) \quad (2.63)$$

$$\frac{\partial E}{\partial net_k^{kw,kh}} = \frac{E(net_k^{kw,kh} + \varepsilon) - E(net_k^{kw,kh} - \varepsilon)}{2\varepsilon} + \mathcal{O}(\varepsilon^2) \quad (2.64)$$

$$\frac{\partial E}{\partial O_i^{jw,jh}} = \frac{E(O_i^{jw,jh} + \varepsilon) - E(O_i^{jw,jh} - \varepsilon)}{2\varepsilon} + \mathcal{O}(\varepsilon^2) \quad (2.65)$$

Perturbing tests of weights, net sums (*net*), and inputs, gave us correct result for a lower bound of the values within the  $\varepsilon = 10^{-7}$ . Sample outputs of these tests are presented at Tables 2.1, 2.2, and 2.3. At our software environment, the double data type has 16 digits precision. Test results showed that difference of all output have a maximum 14 digits accuracy.

Therefore, we investigated the potential the reason of loosing two digits precision on these calculations. We tried same evaluation technique for activation function  $f()$ .

**Table 2.1:** Input Perturbing Results.

Epsilon	([E+]-[E-])/2 Epsilon	input	Difference	The result
0.000000001	-0.000079486417	-0.000079493935	0.00000000751785	not passed
0.000000010	-0.000716165460	-0.000716165400	0.00000000005995	not passed
0.000000100	0.000639053310	0.000639053257	0.00000000005238	not passed
Test Results are not correct for the values out of the lower bound of $\epsilon = 10^{-7}$				
0.000001000	-0.000447143133	-0.000447143136	0.000000000002700	passed
0.000010000	-0.000163595652	-0.000163595653	0.000000000000525	passed
0.000100000	0.000538057734	0.000538057734	0.000000000000093	passed
0.001000000	-0.000622696962	-0.000622696985	0.000000000022999	passed
0.010000000	-0.000053653760	-0.000053653524	0.000000000236161	passed
0.100000000	0.000168174450	0.000167966391	0.000000208058993	passed
0.000100000	0.000364155993	0.000384254499	0.000020098505333	passed
0.001000000	0.000020931019	0.000693037434	0.000672106414911	passed

**Table 2.2:** Net Sum (*net*) Perturbing Results.

Epsilon	([E+]-[E-])/2 Epsilon	delta	Difference	The result
0.0000000010	-0.177685366420	-0.177685355259	0.000000011161	not passed
0.0000000100	0.147506491743	0.147506489943	0.000000001799	not passed
0.0000001000	-0.003295744428	-0.003295744424	0.000000000003	not passed
Test Results are not correct for the values out of the lower bound of $\epsilon = 10^{-7}$				
0.0000010000	-0.270490333048	-0.270490333072	0.0000000000023	passed
0.0000100000	0.212703600508	0.212703600512	0.0000000000004	passed
0.0001000000	0.162735418196	0.162735418367	0.000000000170	passed
0.0010000000	-0.258074055308	-0.258074100766	0.000000045457	passed

**Table 2.3:** Weight perturbing Results. NP: not passed.

Epsilon	([E+]-[E-])/2 Eps.	Weight	Difference	The result
0.000000001	0.26026451827032	0.26026454051157	0.00000002224124	NP
0.000000010	-0.00490656665687	-0.00490656664767	0.00000000000920	NP
0.000000100	-0.03905583908364	-0.03905583910458	0.00000000002094	NP
Test Results are not correct for the values out of the lower bound of $\epsilon = 10^{-7}$				
0.000001000	0.03816106281754	0.03816106281635	0.00000000000119	passed
0.000010000	0.01230801236543	0.01230801236545	0.00000000000002	passed
0.000100000	-0.00751168860717	-0.00751168861326	0.00000000000609	passed
0.001000000	0.04911920939557	0.04911920973274	0.00000000033717	passed

**Table 2.4:** Activation Function Perturbing Results. NP: not passed.

Epsilon	([Act+]-[Act-])/2 Eps.	Derivative	Difference	The result
0.000000001	1.02568442628125	1.02568444088404	-0.00000001460278	NP
0.000000010	1.02568444293460	1.02568444088404	0.00000000205055	NP
0.000000100	1.02568444071415	1.02568444088404	-0.00000000016989	NP
Test Results are not correct for the values out of the lower bound of $\varepsilon = 10^{-7}$				
0.000001000	1.02568444088069	1.02568444088404	-0.00000000000335	passed
0.000010000	1.02568444086403	1.02568444088404	-0.00000000002001	passed
0.000100000	1.02568443983541	1.02568444088404	-0.00000000104863	passed
0.001000000	1.02568433605326	1.02568444088404	-0.00000010483078	passed

```
// Activation function of hidden nodes
double hidddenode::activation(const double x)
{
    return (1.7159*tanh(2.0*x)/3.0);
}
```

Perturbing activation function:

$$f'(x) = \frac{f(x + \varepsilon) - f(x - \varepsilon)}{2\varepsilon} + \vartheta(\varepsilon^2) \quad (2.66)$$

The results reveals that the precision loss may caused from the activation function specifically  $\tanh()$  function (see Table 2.4).

## 2.4 Fine Tuning

Training a NN is commonly named as a black art, instead of a science. The source of this opinion is the performance of the NN is very dependent on the training parameters and the data set. There are a lot of small tricks that help to success of NN [29, 30, 34, 35]. To obtain a better training, some of these tricks are implemented in our model, described as below.

### 2.4.1 Intrinsic subsampling:

By selecting the step size equal to two at layer-2 and layer-3, the subsampling is implemented implicitly [29].

### 2.4.2 Stochastic learning and shuffling the examples:

We train the NN with a single sample at per training epoch (i.e. stochastic training) and shuffle samples for better training performance.

### 2.4.3 Normalizing the inputs:

When all inputs are positive, then all weights will be updated to the same direction. Thus, the NN requires a greater bias, which is proportional to the signs and values of inputs to shifting system. Reaching to a greater bias value takes time (i.e. iteration), as a result of this, the performance of training is reduced. To avoid this problem, the average of input samples are shifted to zero and the covariance scaled to one.

Covariance of inputs calculated by [34]:

$$C_i = \frac{1}{P} \sum_{p=1}^P (O_i^p)^2. \quad (2.67)$$

### 2.4.4 Narrowing initial weights:

Toward to borders of the activation function, activation function gives a flat output. The activation function takes the sum of the all input that is multiplied by their own weights as a parameter. We should also remember that the training is the process of finding the weight values. As a result of this, selecting good initial weights is an important process that directly effects the resulting converting rates. To initialize the weights, we use the following Equation 2.68 that is refined formula of Lecun et al. [27]:

$$\frac{0.3}{M} < |W_{\text{init}}| < \frac{2.4}{M}. \quad (2.68)$$

where  $M$  is the number of inputs that feed the node.  $\pm 0.3/M$  part of the equation is added for avoiding of initial weights too close to zero.

### 2.4.5 Selecting the activation function:

Same as all positive inputs described above, all positive outputs also reduce the training performance. The input of the next layer is the output of the previous layer. Then, selecting an activation function, that the middle of its ranges is zero, helps to normalize the inputs of the next layer. Therefore,  $\tanh()$  function is selected as an activation function primarily in which its output ranges from -1 to +1.

### 2.4.6 Defining target ranges:

As mentioned above item, the activation function gives a flat output at borders. Therefore, the target values are scaled between -0.9 and 0.9.

### 2.4.7 Proper learning rate:

In our study, we use two different learning rate techniques:

First, *Setting different learning rate for every node*: Derivative of an error function used by weight update, is smaller at lower layers and bigger at higher layers. To eliminate this factor, different learning rates are selected for every node ( $\eta_n^l$ ) using following Equation:

$$\eta_n^l = \frac{m^{\frac{1}{2}} \eta^G}{20 l}, \quad (2.69)$$

where  $m$  is the number of inputs feeding the node,  $l$  is the layer number, and  $\eta^G$  is the global learning rate.

The second learning rate technique is *Automatically decreasing learning rate*: To avoid the local minima, we used a learning rate decreasing approach given by following equation [30]:

$$\eta_{n,now}^l = \frac{\eta_{n,begin}^l}{\frac{r}{R/2} + \frac{50}{\max(1, (50 - \frac{\max(0, 50(r-0.65R))}{(1-0.65)R}))}}. \quad (2.70)$$

where  $\eta_{n,now}^l$  is the current learning rate and  $\eta_{n,begin}^l$  is the initial learning rate for this node,  $r$  is the current training epoch, and  $R$  is total number of training epochs.

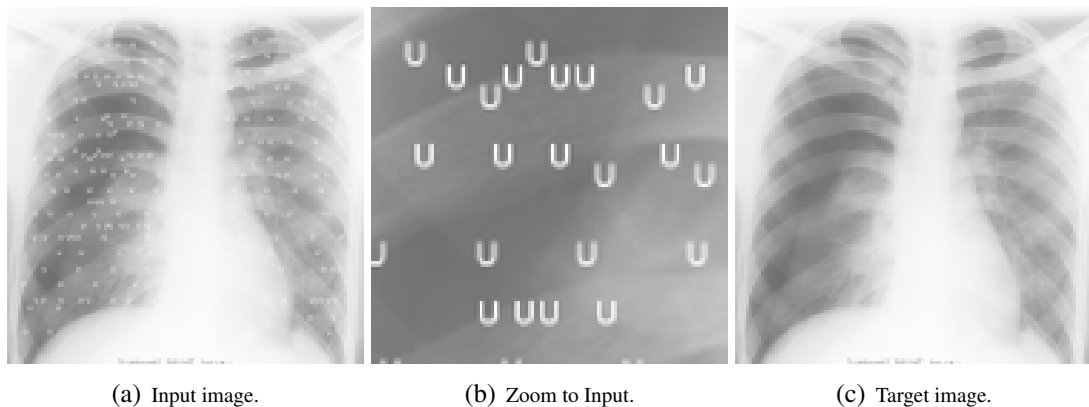


### 3. INITIAL EXPERIMENTS

In this dissertation, it is preferred to present our numerical experiments in a time order and naturally, from simple one to more complex ones.

#### 3.1 Removing 'U' from the Chest Radiography

We have tried to run a test case to find and test the suitable model size and architecture of the chest tube detecting case. A Postero-Anterior (PA) chest radiography image is selected, and small size U letters are embedded to the images which are seen in Figure 3.1(b).



**Figure 3.1:** Not An Easy Training Case (Object Removing).

In our model, this artificially contaminated image is used as an input image (Figure 3.1(a)) and uncontaminated (original) chest radiography image is used as a target image (Figure 3.1(c)). The work plan of the model is simple: First feed the NN with contaminated chest image, and then train to get uncontaminated (original) image (Figure 3.2 ).

We started to solve this test case as an example of artificial object removing case (in fact, this type of image reconstruction is named as the inpainting). The letter 'U' embedded to the chest x-ray image is tried to remove. This test case is intentionally selected in a way that it represents nearly as hard as our target case in terms of

architecture, but also as easy as possible in terms of data requirement (Figure 3.3 ).

We tried from 1 to 3 hidden layers, and 1 to 150 nodes and lots of parameter variations. It is known that the NN training success depends on the data set and algorithm as much as architecture. After certain number of experiment on the data set and architecture, we were ready to test some algorithm modifications such as different training techniques.

### 3.1.1 Using more dense data set

Initial studies reveal that results given in Figures 3.1 and 3.3 are not correct. However, errors mainly coming from the removing letter "U" part of the process. Inputs that do not contain letter "U", were seem to be accurately produced in our results.

As a result of this observation, we thought that our data set does not contain enough number of letter "U" for the learning process to remove. After using the new data set that contains more densely letter "U" (Figure 3.4), our results were very promisingly corrected and displayed in numbers (Figure 3.5), but it should be noted that there are some striping marks which may be observed by eye (Figure 3.6).

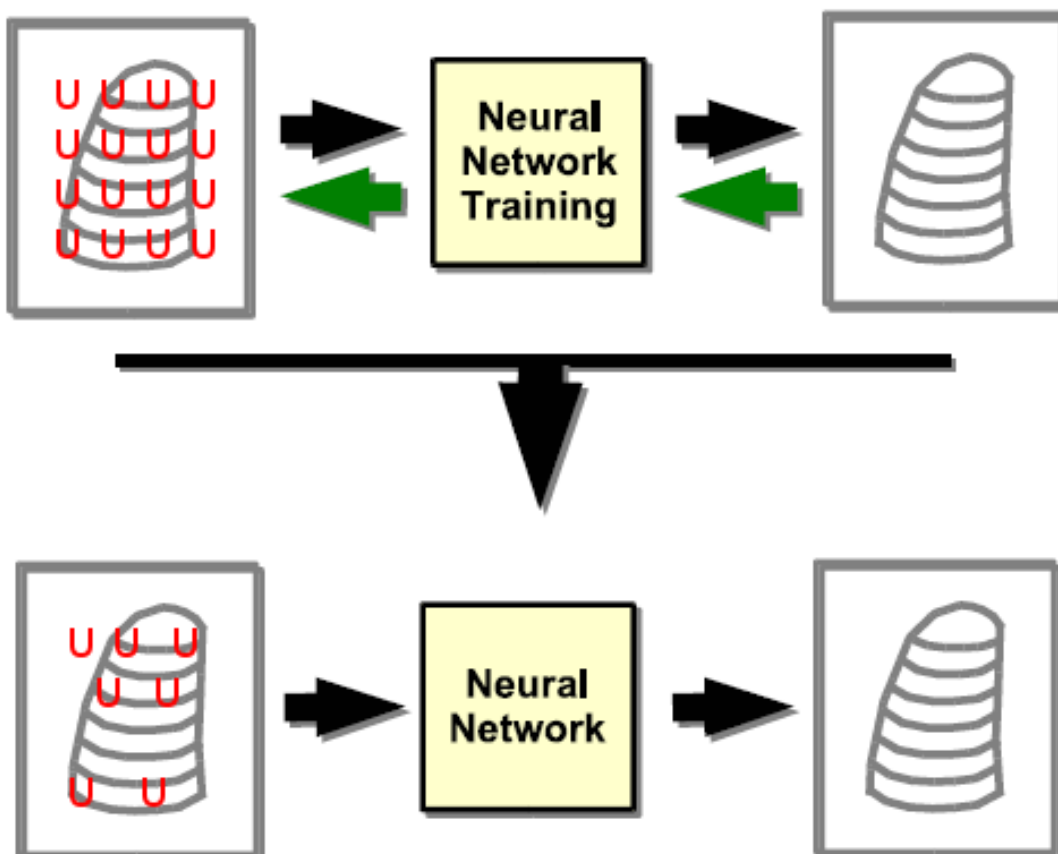
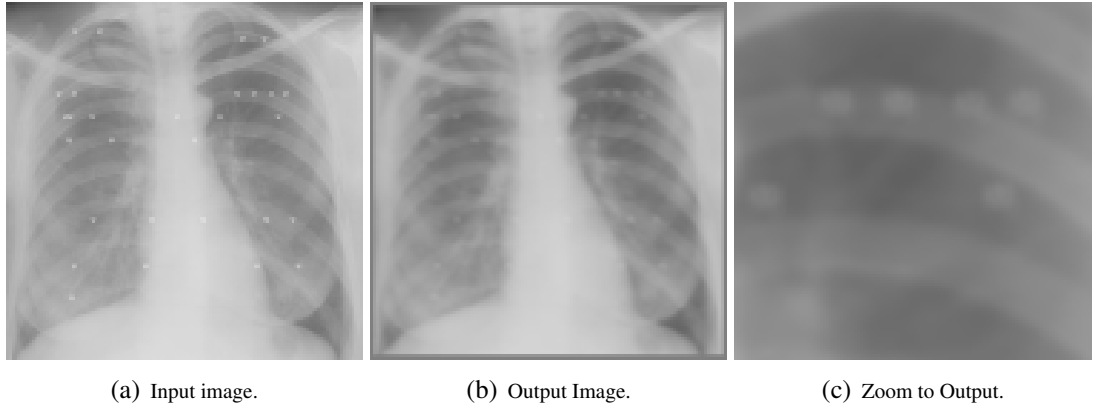


Figure 3.2: The work plan for Removing 'U' Test Case.

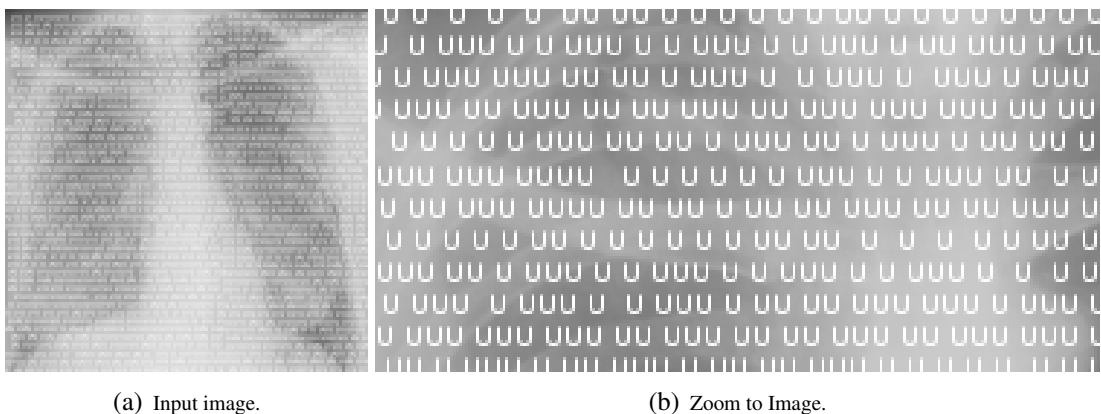




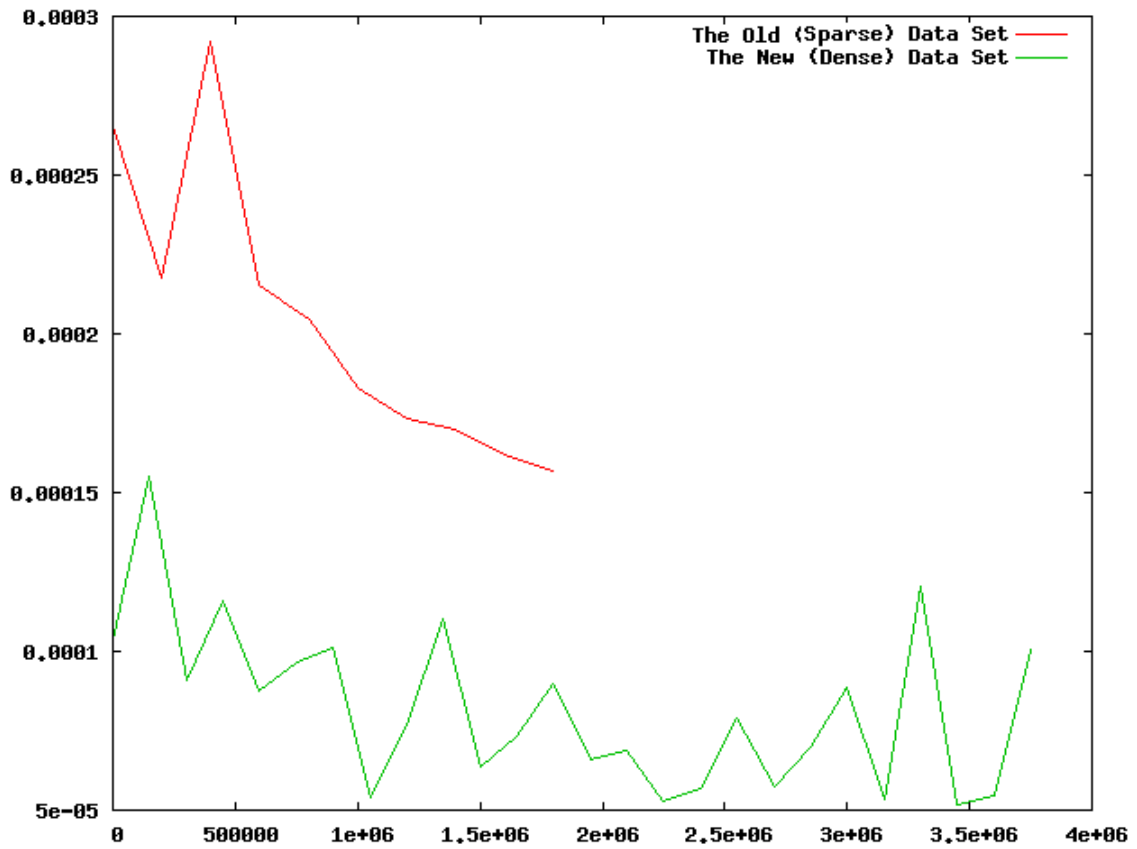
**Figure 3.3:** Removing 'U' Case Training Results.

**Table 3.1:** Best results of removing 'U' with using sparse dataset. Each of Input and Output Layers contains one node. N.: Nodes, I.W.:Initial Weights, L.R.:Learning Rate.

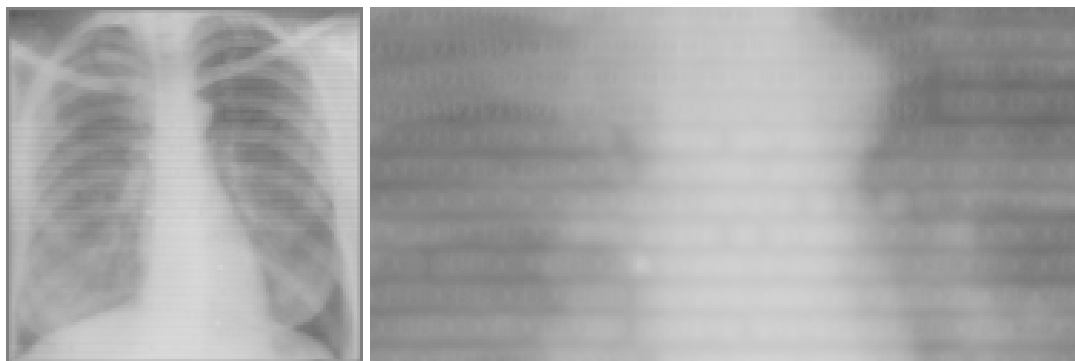
Input (1 N.)	Hidden 1 (4 N.)	Hidden 2	Hidden 3	I.W.	L.R.	Best Results
11x11	7x7	5x5 (12 N.)	3x3 (20 N.)	0.01	0.02	0.000148918
11x11	9x9	5x5 (12 N.)	3x3 (20 N.)	0.01	0.02	0.000155617
11x11	7x7	3x3 (12 N.)	1x1 (20 N.)	0.01	0.02	0.000161590
17x17	5x5	5x5 (12 N.)	1x1 (50 N.)	0.01	0.02	0.000168466
15x15	11x11	5x5 (12 N.)	3x3 (16 N.)	0.01	0.02	0.000171094
17x17	9x9	5x5 (12 N.)	3x3 (20 N.)	0.01	0.02	0.000171929
15x15	11x11	5x5 (4 N.)	3x3 (20 N.)	0.01	0.02	0.000172323
15x15	11x11	5x5 (12 N.)	3x3 (20 N.)	0.005	0.02	0.000172638
15x15	11x11	5x5 (12 N.)	3x3 (8 N.)	0.01	0.02	0.000173310
15x15	15x15	5x5 (12 N.)	3x3 (12 N.)	0.01	0.02	0.000173821
17x17	9x9	5x5 (12 N.)	3x3 (20 N.)	0.01	0.02	0.000174136
15x15	11x11	5x5 (4 N.)	3x3 (20 N.)	0.01	0.02	0.000174679



**Figure 3.4:** The Dense Data Set.



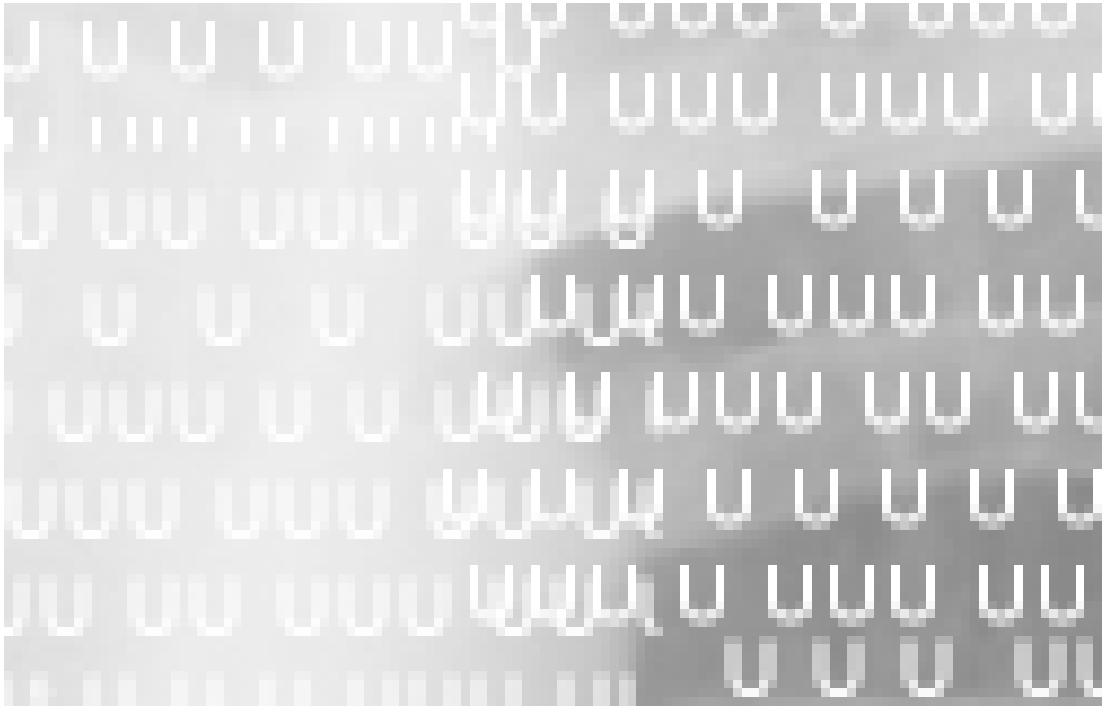
**Figure 3.5:** The Comparison Graph of Sparse and Dense Data Set. Same architecture, different data sets: 11x11 (1 node), 7x7 (4 nodes), 5x5 (12 nodes), 3x3 (20 nodes), 1x1 (1 node), 0.01, and 0.02.



(a) Result image.

(b) Zoom to Image.

**Figure 3.6:** The Dense Data Set Results.



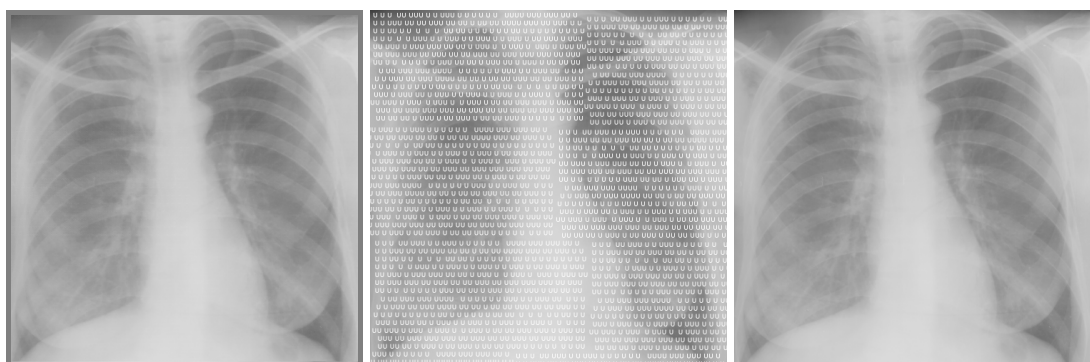
**Figure 3.7:** The Incorrect Part of the Input Image.

We found that some incorrectly generated input files may cause these stripes (Figure 3.7).

After the correction of these input files, the problem of stripes related to the training dataset were removed from our final results. Results show that the performance of NN is very dependent on the training dataset [29].

### 3.1.2 Results of the removing 'U' case using dense data set

We have tried 1000 run with 100 different architectures and techniques. The performance of these NNs is very different, and generally all tuning tips gave small gain. The best result is obtained from a composition of these whole small tricks. Due



(a) The Best Result.

(b) Input Image for Testing.

(c) Target Image for Testing.

**Figure 3.8:** The Best Result and Test Images.

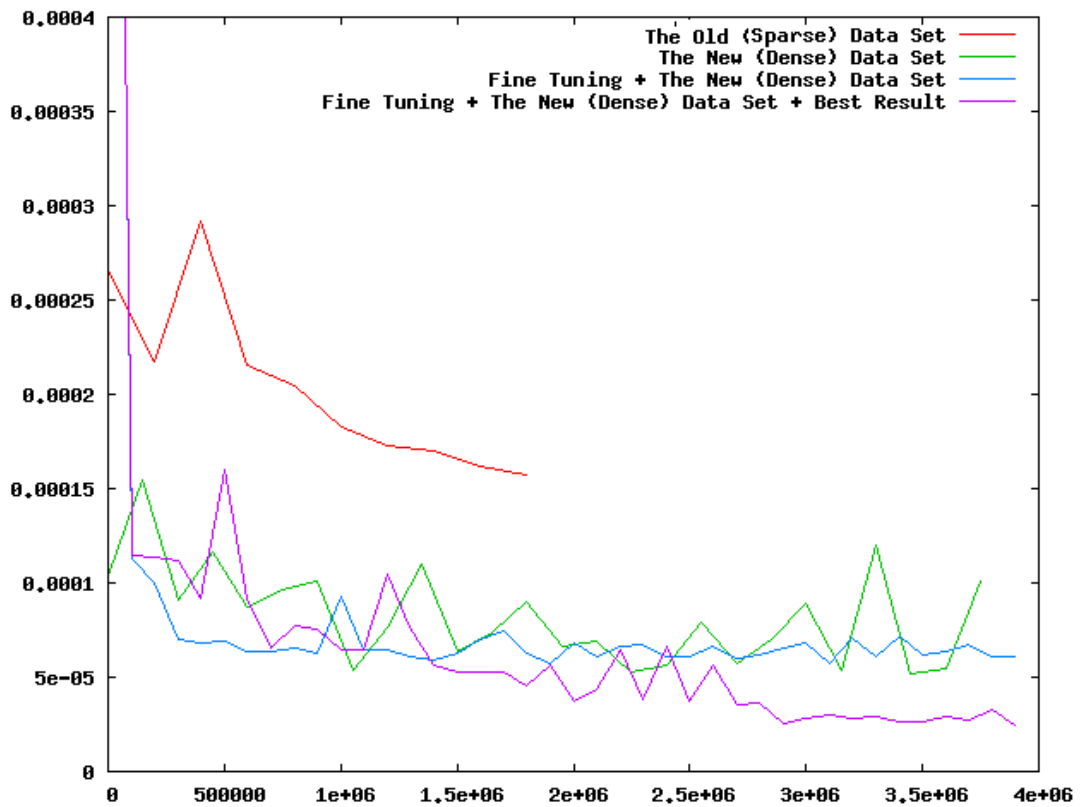
**Table 3.2:** Best Results of Removing 'U' Case. Each of Input and Output Layers contains one node. N.: Nodes.

Input	Hidden 1	Hidden 2	Hidden 3	Learn.Rate	Best Result
15x15	6x6 (23 N.)	3x3 (23 N.)	1x1 (23 N.)	0.7	0.00001714
15x15	6x6 (23 N.)	3x3 (23 N.)	1x1 (23 N.)	1.0	0.00001725
15x15	6x6 (23 N.)	3x3 (37 N.)	1x1 (23 N.)	0.9	0.00001780
15x15	6x6 (37 N.)	3x3 (37 N.)	1x1 (37 N.)	0.5	0.00001824
15x15	6x6 (23 N.)	3x3 (23 N.)	1x1 (23 N.)	0.9	0.00001825
15x15	6x6 (23 N.)	3x3 (37 N.)	1x1 (23 N.)	1.0	0.00001831
15x15	6x6 (23 N.)	3x3 (23 N.)	1x1 (23 N.)	0.8	0.00001849
15x15	6x6 (23 N.)	3x3 (23 N.)	1x1 (23 N.)	0.6	0.00001858
15x15	6x6 (50 N.)	3x3 (50 N.)	1x1 (50 N.)	0.9	0.00001861
15x15	6x6 (17 N.)	3x3 (23 N.)	1x1 (17 N.)	0.8	0.00001930
15x15	6x6 (17 N.)	3x3 (23 N.)	1x1 (17 N.)	1.0	0.00001958
13x13	5x5 (50 N.)	2x2 (50 N.)	1x1 (50 N.)	0.8	0.00002007
13x13	5x5 (25 N.)	2x2 (50 N.)	1x1 (25 N.)	0.6	0.00002069
3x13	6x6 (4 N.)	3x3Sub (4 N.)	1x1 (8 N.)	0.08	0.00004769
11x11	4x4 (4 N.)	2x2 (12 N.)	1x1 (20 N.)	0.8	0.00004993
11x11	4x4 (4 N.)	2x2 (12 N.)	1x1 (20 N.)	0.7	0.000 05058

to the difficulty of showing the whole study at this dissertation, the best results and some characteristic examples are given in the Table 3.2.

After all small fine tunings are applied, our synthetic case results gave us very good performance given in Table 3.2, Figure 3.8, and Figure 3.9. In order to compare our results with the literature, we found only limited papers on image inpainting. However, the inpainting literature generally uses a gap map to select which part of the image will be inpainting. However, in our method, CNN recognize letter "U" (object recognition), then, automatically remove and fill the gap (image reconstruction/image inpainting).

Similar to some of the previous literatures [28, 29], we found that the separate subsampling layer does not useful for better performance. We concluded that the selecting step size greater than one at convolution layer, intrinsically does the subsampling with a better performance (Table 3.2).



**Figure 3.9:** The Comparison Graph of Sparse, Dense Data Set and Best Result of Removing 'U' Case. Same architecture, different data sets: 11x11 (1 node), 7x7 (4 nodes), 5x5 (12 nodes), 3x3 (20 nodes), 1x1 (1 node), 0.01, and 0.02. The Architecture of the Best Result is different: 15x15 (1 node), 6x6 (23 nodes), 3x3 (23 nodes), 1x1 (23 nodes), 1x1 (1 node), Auto Weights, and 0.7.



## 4. CNN FOR DETECTION OF CHEST TUBES

The CNN architecture for detection of chest tubes, has some differences from the architecture of our initial experiments. The changes required because of the complexity of chest tube detection process.

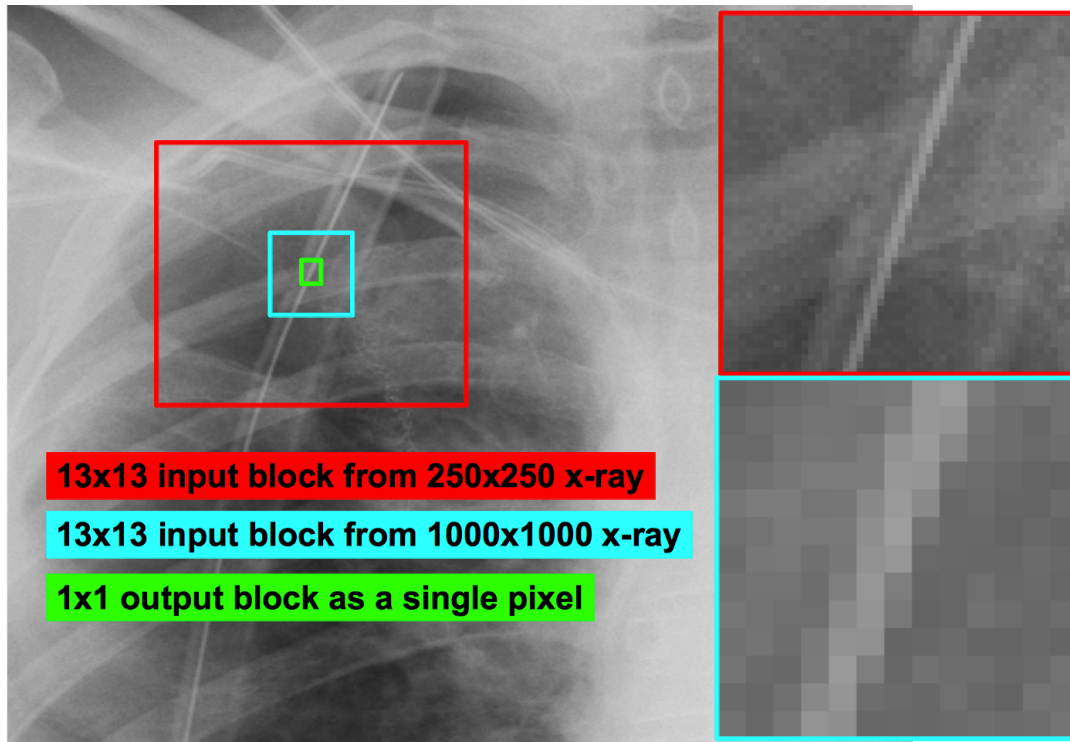
### 4.1 Training and Testing Datasets

Two different sources of the chest radiography used to obtain images as used for training and testing datasets (Table 4.1). All images converted to 8 bits gray scale color depth to obtain a standard intensity range, because of different x-ray machines have different intensity range such as 10, 12, 14 bits.

The training set is solely constructed from our data set, and contains 62 radiographs. As seen in Table 4.1, 13 of the images contain two chest tubes, 25 contain a single chest tube, and 24 out of 62 radiographs have no chest tube. We tested our model using two datasets. The first dataset contains our test set, which is constructed with 21 radiographs with similar groupings. To distinguish the chest tubes from other artificial objects, our training and testing sets contain 36 and 13 radiographs, respectively, including artificial objects other than chest tubes. The second test set contains 247 images without chest tubes, which is publicly available; it was created by the Standard Digital Image Database Project Team of the Scientific Committee of the Japanese Society of Radiological Technology (JRST) 6.3.

**Table 4.1:** Training and testing data sets.

Radiographs Content(s)	Our Training Set	Testing Images		
		Our Test Set	JRST Database	Testing Total
No chest tube	24	8	247	255
Single chest tube	25	11	0	11
Two chest tubes	13	2	0	2
Other artificial object(s)	36	13	4	17
Total	62	21	247	268



**Figure 4.1:** Two Scales of Input Blocks.

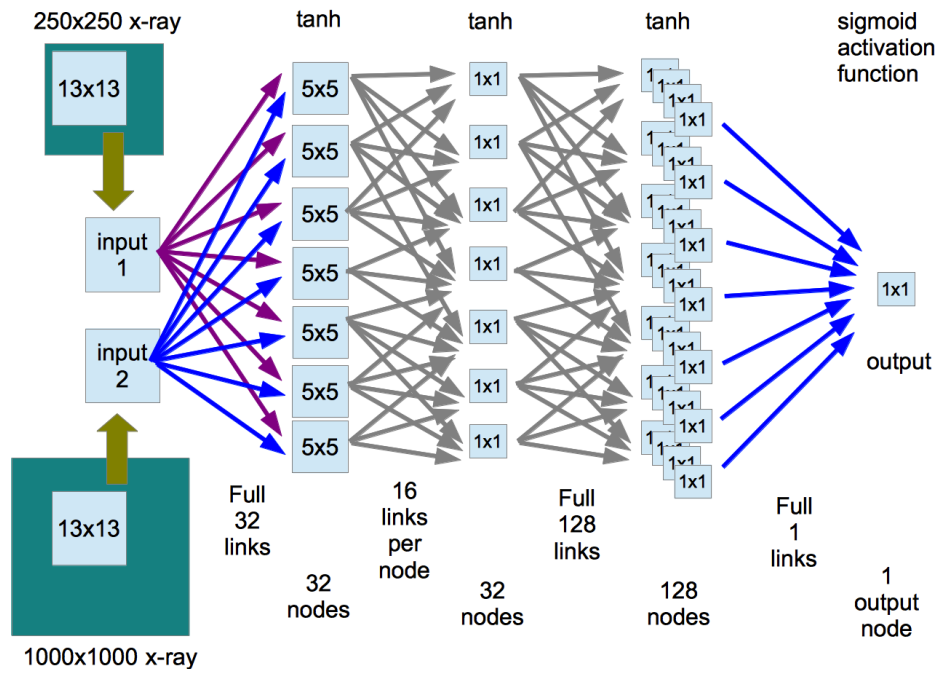
## 4.2 CNN Architecture

To detect the chest tubes in chest radiographs, a CNN architecture containing  $L = 5$  layers is used (Figure 4.2). There are three types of layers; these layers are the input layer ( $l = 1$ ), the output layer ( $l = L$ ), and the hidden layers ( $1 < l < L$ ). The input layer is an abstract layer that contains only the input data without any calculations. Different numbers of node combinations of hidden layers, including 8, 16, 32, 64, 128 and 256, were tested using different learning rates, such as 0.1, 0.01, 0.001, 0.0001, and a gradient descent with a *tanh* activation function in the hidden layers.

We also tested the Stochastic Diagonal Levenberg-Marquardt (SDLM) update rule and the Cross-Entropy (CE) method. On our tests and the results of Simard et al. [29] show that CE gives best performance. According to these findings, we decided to use the Gradient Descent and Cross-Entropy algorithms with a sigmoid activation function. Our final CNN architecture contains 2, 32, 32, 128, and 1 nodes for the successive layers. Between layers, there are 32, 16, 128, and 1 links for each node in the layers. After a series of tuning tests, the learning rate was selected as 0.1.

To feed the system with a greater input region without increasing the model complexity, a multi-scale input with two scales was used. The input image blocks used by the input layer, cropped from two whole images of a chest radiograph at multi-scale sizes

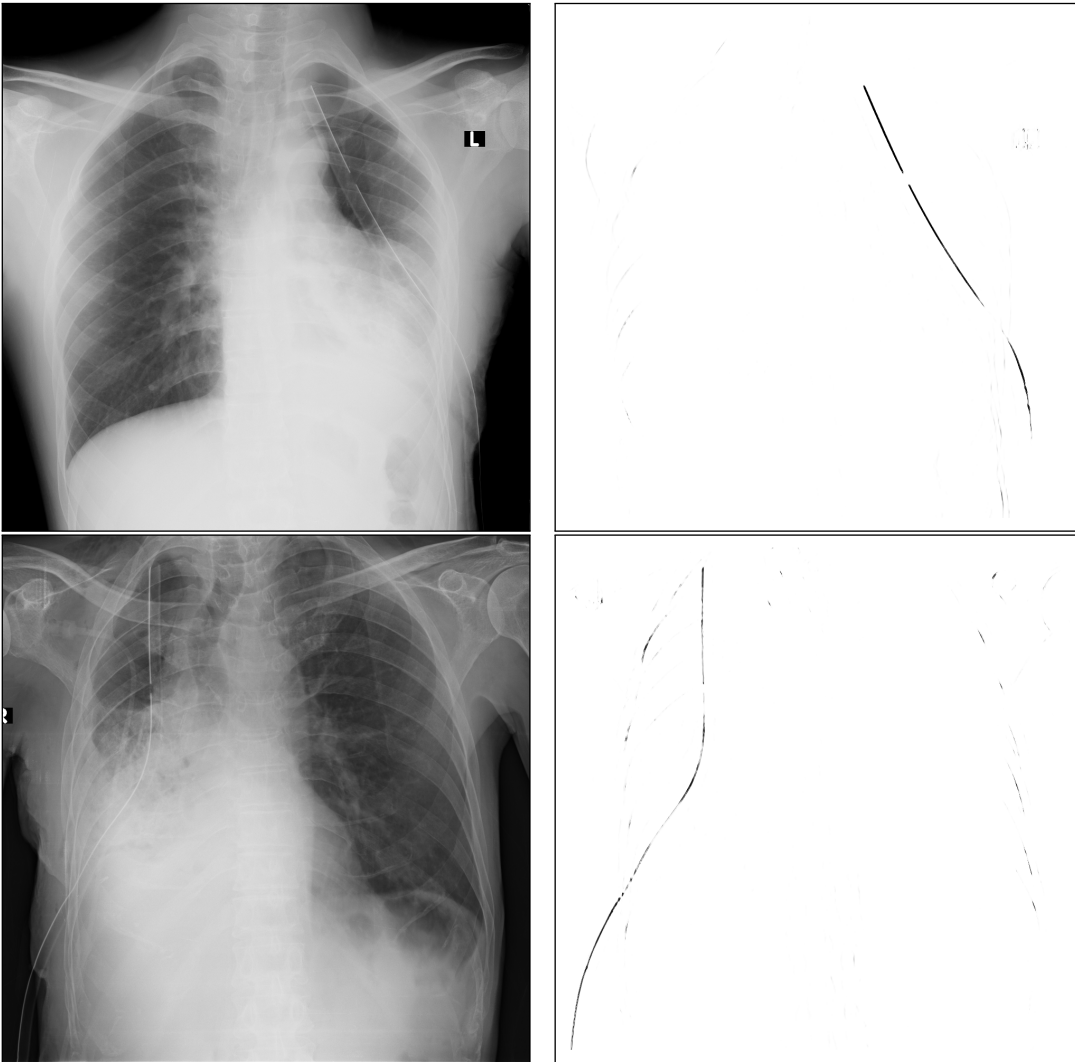




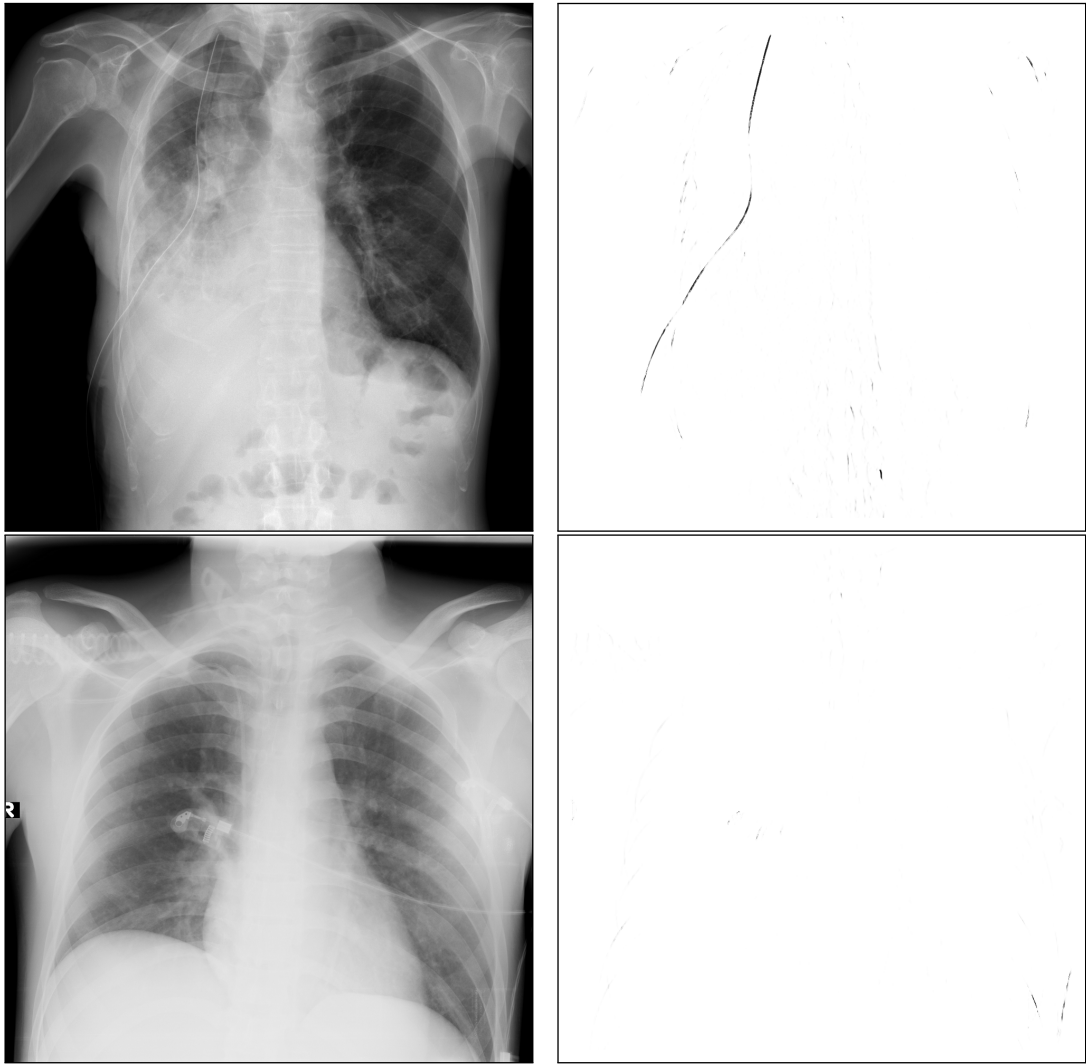
**Figure 4.2:** The Final CNN for Detection of Chest Tubes.

of 1000x1000 and 250x250 pixels, are used without any registration. The input layer contains two nodes with two input blocks of 13x13 pixels each (Figure 4.1). During the training stage, blocks are selected from random training image sets with a random block position. At three hidden layers, the output sizes are chosen as 5x5, 1x1 and 1x1. Finally, the output layer contains a single node that gives outputs of 1x1 in size, which is used as a pixel in the resulting image at the proper location according to the input block position.

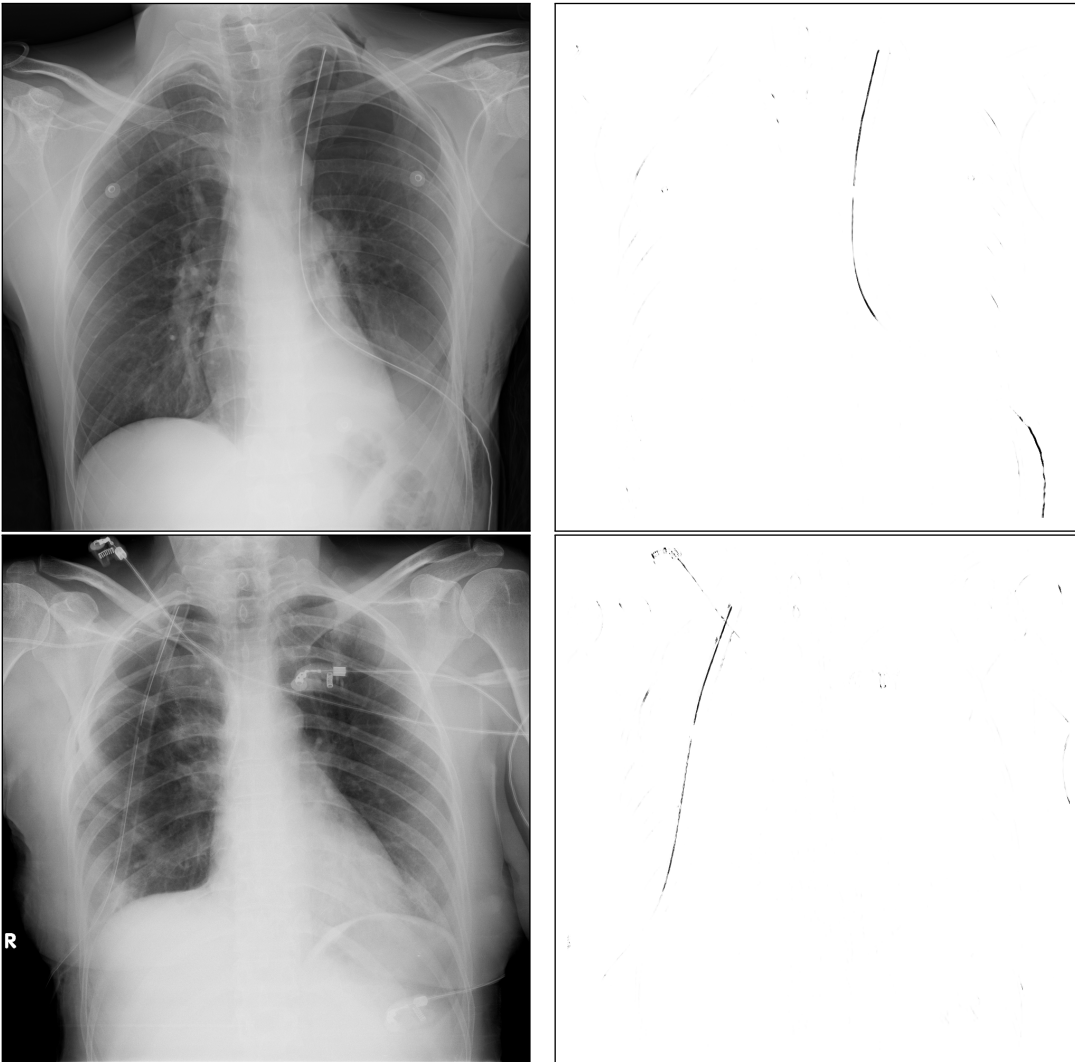
The results of the stage of the CNN for chest tube detection, are presented in Figures 4.3, 4.4, 4.5, and 4.6.



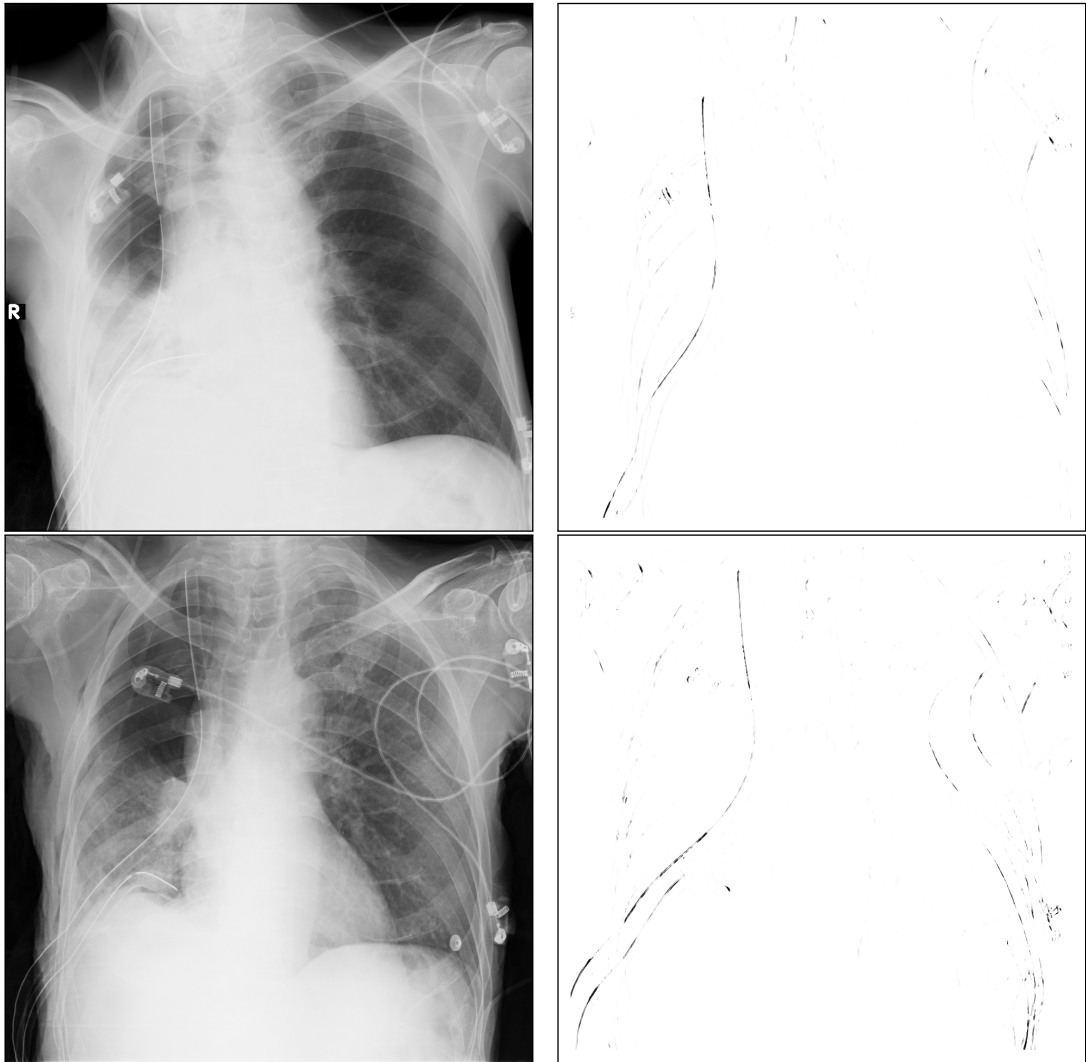
**Figure 4.3:** Selected successful results from our test data set. The left side images are inputs of the CNN model. The right side images are the outputs of the CNN model.



**Figure 4.4:** Selected successful results from our test data set. The left side images are inputs of the CNN model. The right side images are the outputs of the CNN model.



**Figure 4.5:** Selected erroneous results from our test data set. The left side images are inputs of the CNN model. The right side images are the outputs of the CNN model.



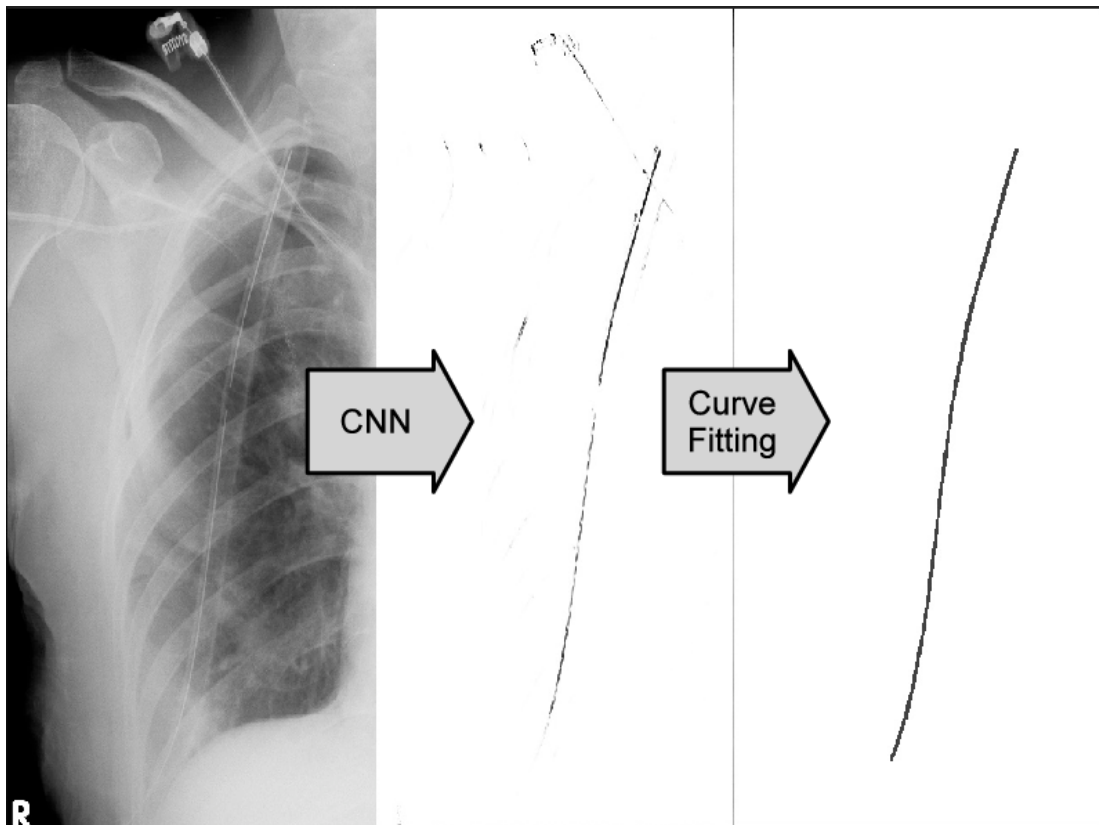
**Figure 4.6:** Selected erroneous results from our test data set. The left side images are inputs of the CNN model. The right side images are the outputs of the CNN model.



## 5. NURBS CURVES

### 5.1 The Noise and the Continuity

In some cases, the output of the CNN algorithm provides an unclear and interrupted chest tube skeleton that is not practical for use. Therefore, we introduced an adaptive curve-fitting approach to the output images of our CNN model to obtain a final continuous chest tube skeleton (Figure 5.1).



**Figure 5.1:** The stages of our model. The left image is the input, the middle image is the output of the CNN, and the right image is the result of the curve fitting process.

The output of the CNN contains sparse single positive pixels and some small group of positive pixels. These are easily cleanable by scanning output. We clean these small residues before curve fitting. But all noises cannot be removed by scanning.

To eradicate the all noise and enhance the output, we apply the curve fitting algorithm (Algorithm 1). The algorithm starts with the strongest (darkest in the image) and longest continuous line segment, and follows the line in both directions. While following the strongest output, the algorithm searches longest continuous line segment from beginning point in the range of 60 degrees. The longest continuous line segment is followed until 36 pixels long. When length of line segment reaches the 36 pixels, or it reaches a discontinuity (a space), the last pixel is selected as a control points to reduce the number of control points that are used for curve fitting. Then again the algorithm searches a new longest continuous line segment from the last control point in the range of 60 degrees.

---

**Algorithm 1** Pseudo code for selecting control points of the NURBS curve.

---

```

repeat
  repeat
    repeat
      Save current position as a control point,
      for all do  $-30, -25, -20, \dots, 0, \dots, 20, 25, 30$  degrees
        Search the end of the connected pixels in this angular direction,
      end for
      Select the angle of the longest continuous line,
    until Walk 36 continuous pixels ahead or end of the connected pixels,
    Try to find new connected pixel group in the same direction,
  until No new connected pixels,
  for all  $-30, -25, -20, \dots, 0, \dots, 20, 25, 30$  degrees do
    Search a gap followed by a continuous line in this angular direction,
  end for
  Select the angle of the shortest gap followed by a continuous line,
  Save the end of the gap as a control point,
until No new gap followed by a continuous line,
Use control points for curve fitting.

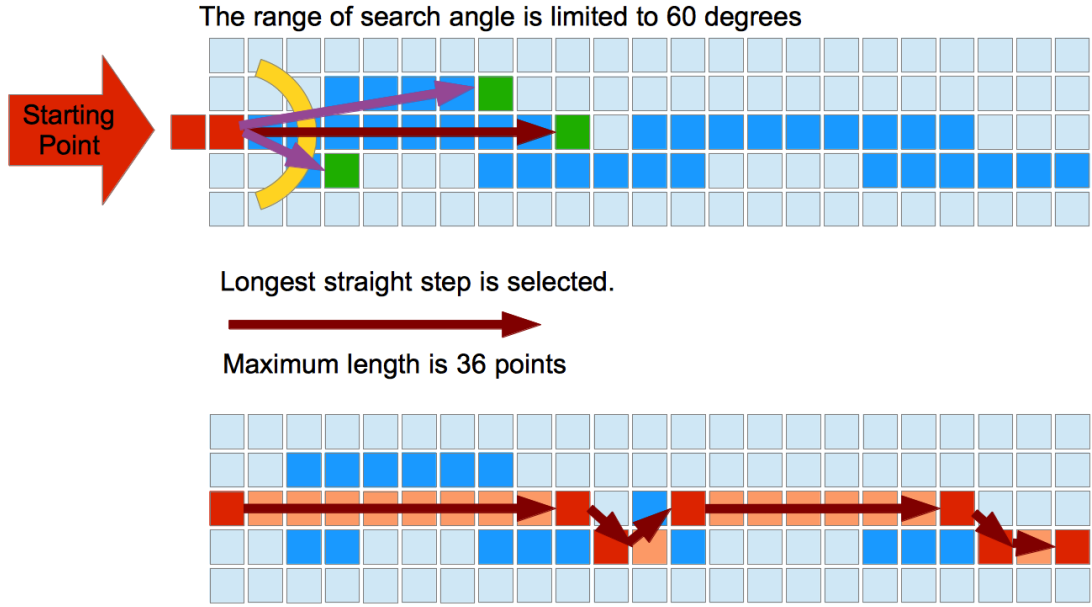
```

---

The control point selection process is conducted by walking over the entire output chest tube curve and it selects control points with an interval of 36 pixels. During this walking process, when the process meets to any discontinuity on a curve, it is forced to jump to the point where the next continuous line segment starts (Figure 5.2). When a new line segment is continued at same direction of previous line segment, the continuous line segment search begins again.

For example, in Figure 4.3, NURBS-based piece wise curves are fitted over segmented lines on the output images of our proposed CNN model and are shown at right side





**Figure 5.2:** The Selection of the Control Points.

images. The pseudo code for fitting NURBS over the output image is given in Algorithm 1. The main idea behind the curve-fitting process is to capture the whole chest tube figure on the x-ray image regardless of any mis-interpretation or confusion.

## 5.2 The NURBS Curve Formula

Nonuniform Rational B-Spline (NURBS) Curves,  $P(t)$ , are used for curve fitting.

$$P(t) = \sum_{i=1}^{n+1} B_i R_{i,k}(t) \quad (5.1)$$

where the  $B_i$ 's are the control polygon vertices, and the  $R_{i,k}(t)$ 's are the rational basis functions.

$$R_{i,k}(t) = \frac{h_i N_{i,k}(t)}{\sum_{i=1}^{n+1} h_i N_{i,k}(t)} \quad (5.2)$$

where the  $N_{i,k}(t)$ 's are the basis functions, and the  $h_i$ 's are homogeneous weighting factors [36].

The control point selection process filters noise and provides clean and discontinuous line segments. This line segments will be transformed to the continuous curve after the NURBS curve fitting.

This approach eliminates the possibility of small artifacts in the output images. Sample input-output merged images of our proposed model are presented in Figures 6.1, 6.2, and 6.3.



## 6. RESULTS AND EVALUATION

### 6.1 Model Performance Evaluation

We evaluated the performance of the model using a pixel-based Receiver Operating Characteristic (ROC) analysis, which is defined very well by Fawcett [37]. Each true positive, true negative, false positive and false negative pixel was counted and average true positive ( $N_{tp}$ ), true negative ( $N_{tn}$ ), false positive ( $N_{fp}$ ), and false negative ( $N_{fn}$ ) values per image were determined. These values were used to calculate the accuracy ( $\psi$ ), sensitivity ( $S_n$ ), and specificity ( $S_p$ ) using the following formulas [37]:

$$\psi = \frac{N_{tp} + N_{tn}}{N_{tp} + N_{tn} + N_{fp} + N_{fn}}, \quad (6.1)$$

$$S_n = \frac{N_{tp}}{N_{tp} + N_{fn}} \quad (6.2)$$

and

$$S_p = \frac{N_{tn}}{N_{tn} + N_{fp}}. \quad (6.3)$$

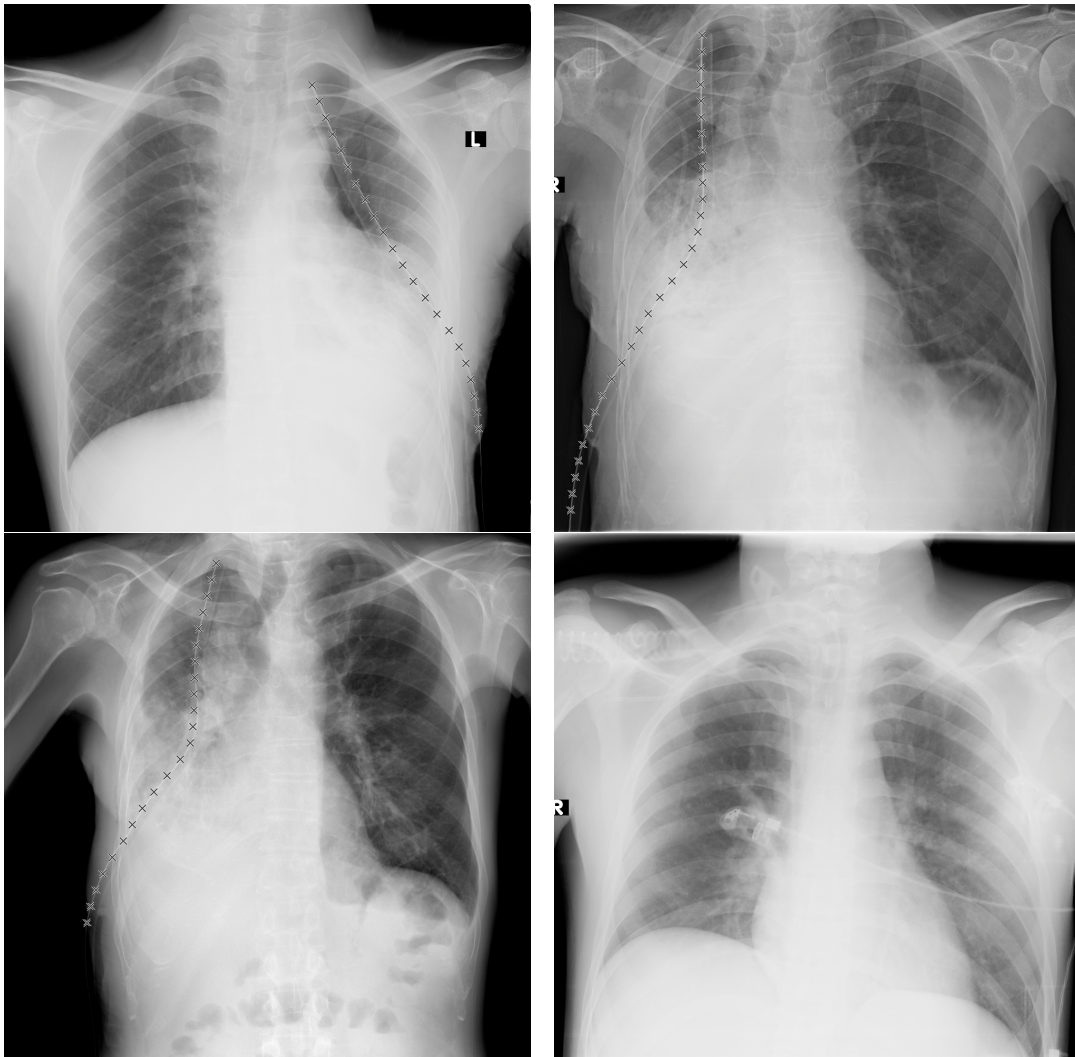
### 6.2 Results

We tested our model using two datasets (Table 4.1). The first dataset contains our test set, including artificial objects and chest tubes, as previously mentioned (Figures 6.1 and 6.2). The second test set contains 247 images without chest tubes, and it is publicly available; it was created by the Standard Digital Image Database Project Team of the Scientific Committee of the Japanese Society of Radiological Technology (JSRT) [14].

The results of our test set show the average Root Mean Square (RMS) errors of the raw output of the Neural Network are summarized in Table 6.1.

**Table 6.1:** Pixel-based average RMS error values of the raw output of the Neural Network. All of the results are obtained by using CNN in 5 layers (2, 32, 32, 128, 1 nodes per layer; 32, 16, 128, 1 links per node).

Radiographs Content	Our Test Set		JSRT Test Set	
	Rads.	RMS Err.	Rads.	RMS Err.
No chest tube	8	0.01857	247	0.006238
Chest tube(s)	13	0.03716	0	NA
Other artificial obj(s).	13	0.03142	4	0,005366
Total	21	0.03008	247	0.006238



**Figure 6.1:** Selected successful results from our test data set. The backgrounds are input x-rays. 'X' marks show the outputs of the proposed model.

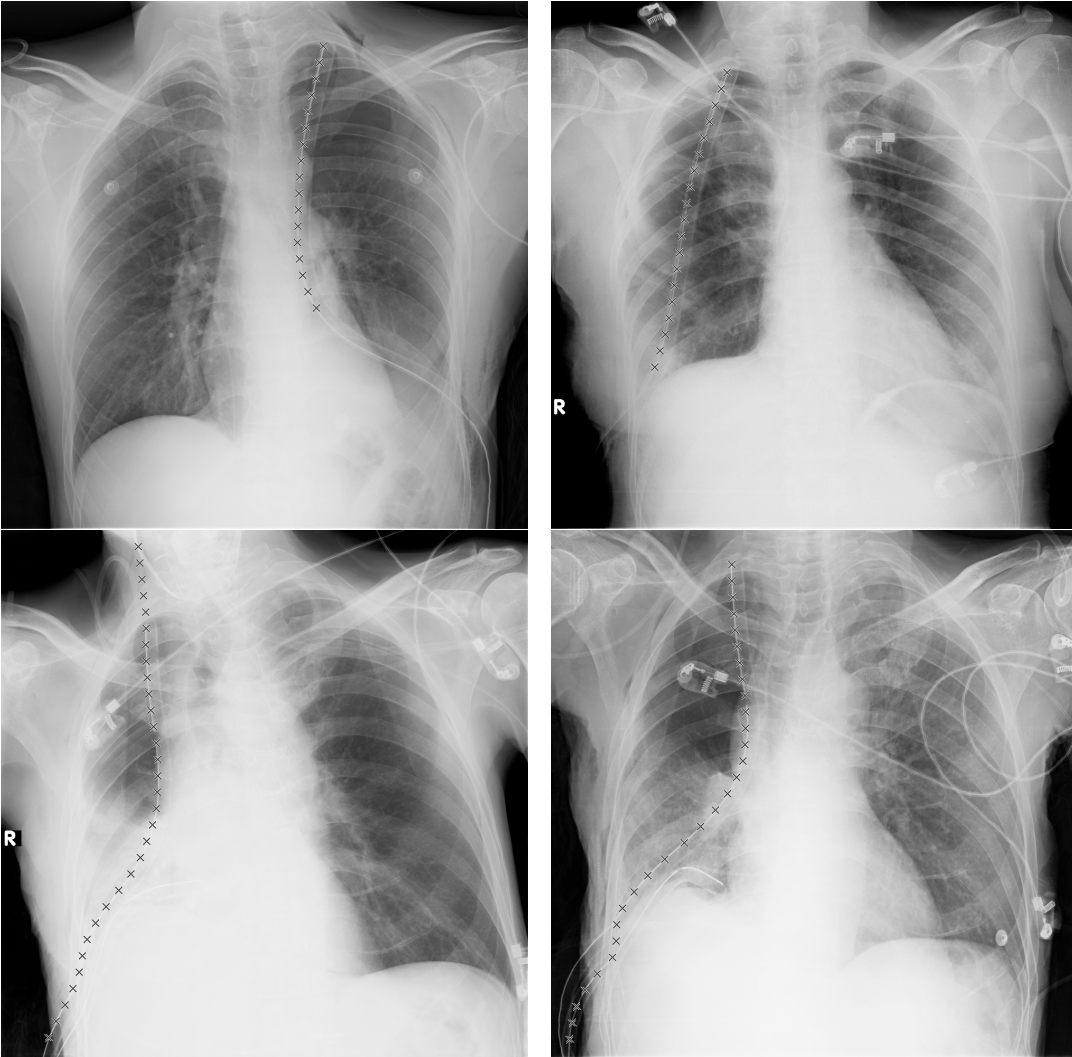
**Table 6.2:** Pixel-based average ROC values per image. TP: true positive, TN: true negative, FP: false positive, FN: false negative.

	Our test set	JSRT Database	All test sets
Image Count	21	247	268
Pixels/image	1000000	1000000	1000000
TP/image	1120	0	88
TN/image	997341	999999	999790
FP/image	765	1	61
FN/image	774	0	60
Accuracy (%)	99.85	99.99	99.99
Sensitivity (%)	59.13	NAN	59.46
Specificity (%)	99.92	99.99	99.99

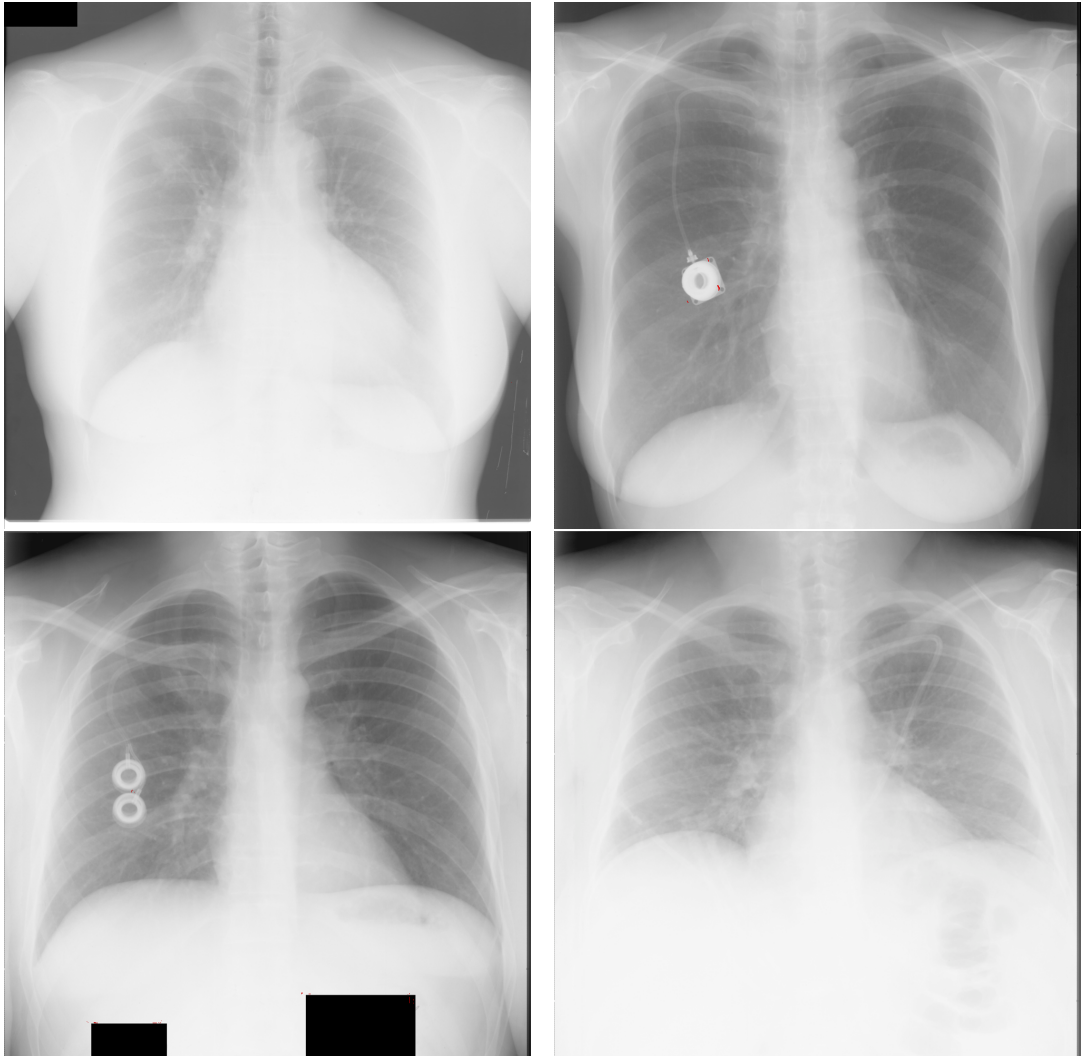
The raw outputs of the neural network refined by our proposed curve-fitting process. The result of our tests using all of the 268 test images was a 99.99% Accuracy, a 59% Sensitivity, and a 99.99% Specificity with 61 False Positive pixels/image and 60 False Negative pixels/image (Table 6.2).

The JSRT database results do not have a False Positive or False Negative except for only one image that had 347 False Positive pixels. These error values show that the results are well aligned with the input images and are acceptable as an output. Some examples of the JSRT database results are shown in Figure 6.3.

Our study is published with title "An approach for chest tube detection in chest radiographs" at Image Processing, IET journal [38].



**Figure 6.2:** Selected erroneous results from our test data set. The backgrounds are input x-rays. 'X' marks show the outputs of the proposed model.



**Figure 6.3:** Selected examples from the JSRT test data set results. Because there is no chest tube present, the chest tube markers are not present in these images. The backgrounds are input x-rays.





## **7. DISCUSSION AND CONCLUSIONS**

### **7.1 Discussion**

In our study, the chest tube is chosen as an artificial object that spans almost the whole chest. In this way, the area of interest for searching and the differentiation of this area are increased. Specifically, ribs can create a pattern that is easily mixed with tubes. Moreover, while the density on the radiograph images increases with the overlapping ribs at the sides, at the same time, there could be zones in which the image density fluctuates with an increase in the rib clearance toward the center of the lung. In spite of this difficulty, we obtained promising results in our study.

While a 99.99% Accuracy, and a 99.99% Specificity were calculated from our results, a 59% Sensitivity is looks like inconsistent. Accuracy, and Specificity values calculated by using TN value, and TN value is very high and because of this reason, it is very dominant (Table 6.2). But Sensitivity value calculated by not using TN value, and as a result FN value reduces the Sensitivity ratio.

#### **7.1.1 Limitations and Source of Errors**

The results reveal that there is no confusion between the chest tubes and other artificial objects (Figures 6.1 and 6.2 ). In our work, we found that the largest source of error is the high-density zones, where the possible location of a chest tube is very difficult to track. These high density zones are formed by overlapping ribs and/or pleural effusion.

Also an other major error source is the presence of radiographs, which contains multiple chest tubes. Our curve fitting method is depending on the longest and darkest line segment is a part of chest tube assumption. But this assumption brings implicitly a deficiency which is our method only can detect one chest tube per radiograph. Because of the "longest and darkest line segment" statement defines a single object. As a result, all radiographs, which contains more than one chest tubes, generates errors. Both false negative and false positive errors are produced. When one of chest tubes is

detected, others will be undetected, and their pixels are marked as false negatives. Sometimes, two of these chest tubes overlaps, and at overlapping area, our algorithm confused to follow the detected one, it follows other chest tube, and these pixels are marked as false positives.

### **7.1.2 Parallelization**

Processing a radiograph using our testing process, runs 3 minutes on the server, which has dual socket, single core 3.4GHz CPU with 16KB L1 cache, 1MB L2 cache and AMD64 architecture with 4GB RAM. Training stage for all training dataset, requires approximately one week. We tried parallelization to reduce one-week wall clock time, but stochastic learning makes parallelization very difficult due to recalculation of the weights for every iteration. For overcoming these difficulties, we tried to parallelized our algorithm by decoupling the training and testing processes. But this approach was not useful, because of unbalanced wall-clock times of training and testing processes. The testing process is quickly finished, and wastefully occupies CPU without calculation.

Also there is an other issue against this type of parallelization: Single training run is not enough to decide about performance of any configuration, because of the random initial parameters. We run five different training instances with randomly selected initial weights. The best result is selected. Because of that, instead of idling CPU's, we preferred to run the algorithm based on Single Program Multiple Data (SPMD) model which is a simple but effective parallel model.

### **7.1.3 To clean the output of CNN and Curve Fitting**

To cleaning output of the CNN, we firstly scan the output for sparse positive (black) pixels. These are false outputs of CNN. But sometimes, some of them are true outputs but not connected to main chest tube skeleton. Because of this reason excessive cleaning is not help to better results.

Also, curve-fitting process requires some tricky settings. Firstly selecting control points as more than 36 pixels long, cause unfairly located control points. Straight parts of chest tube, will be represented with few control points, and curvy parts will be presented with many of control points. The fitted curve, using unevenly

selected control points, is not aligned onto original chest tube skeleton. Therefore, we empirically obtained 36 pixels as a threshold for maximum length of single line segment. With using this threshold, all line segments will be short, and fitted curve exactly aligned.

The fitted curve was calculated as a fourth order polynomial. The flexibility of the chest tubes requires higher order polynomials, but higher order polynomials tend to wiggle excessively. Our test showed that representing the NURBES curve as a fourth order polynomial was an optimal selection for better alignment.

#### **7.1.4 Usability in medical practice**

It is important to note that the detection of artificial objects in medical images has critical importance in the medical image analysis field. Detecting the existence of an artificial object and its location on radiographs has three critically important aspects. First, this capability helps practitioners to detect the locations of these objects. Second, artificial objects can be identified in the images for a PAC system. Finally, taking the required measurements for other CAD methods will not be affected by the existence of an artificial object.

Even if our algorithm has a theoretical capability to detect the multiple chest tubes, currently it is not automatically detecting more than one chest tube and this can be considered as a weakness of our algorithm. But this situation does not makes the algorithm useless. As future work this capability will be activated.

## **7.2 Concluding Remarks**

### **7.2.1 Chest Tube Inpainting**

Inpainting is to remove some image part, estimating possible replacement of the hole remaining from removing operation. Chest tube inpainting is removing the chest tube figure and estimating replacement part from removing image parts and their neighbors.

Chest tube detection is useful, but removing (i.e. inpainting) chest tube figure from the chest tube radiograph image will be more useful. We tried to remove chest tubes using CNN, and we obtain some promising results too.

But, major problem about medical inpainting is ensuring accuracy of the replaced parts. When inpainting ordinary image, replacement parts must only be visually acceptable. But, in medical area, estimated parts must be physiologically and pathologically correct. Certifying the correctness of the estimated parts nearly impossible, due to highly variable nature of pathology.

As a result, we shifted our research focus from inpainting to detection of chest tubes. By this way, our algorithm did not modified the originality of radiographs at hand.

### **7.2.2 CNN Architecture**

CNN architecture is consist of multiple variations, and possible settings of parameters. Layer counts, link directions and counts, learning rate changes, initial values of weights are some of them. Each of them somehow affects success of the output.

When initial values of weights are close to zero, achieving correct weights takes more iterations, and training process requires more cpu time.

Increasing count of layers, makes difficult to training CNN, because of backpropagation process transfers differentials of output error from next layer to previous layers. Each transfer reduces affect of error. Increasing layer count requires more iteration for training.

Also, increasing node counts of last layers, is act as high order polynomial curve fitting. It fits very well to training dataset, but very poor for testing dataset. Because of this reason, excessive node counts of last layers is a trap for success.

An other issue about CNN, is the symmetry. When all nodes and links are symmetric, two half of CNN is tend to produce symmetric output. Thus, the performance of CNN decreases by half, but computational complexity remains the same.

### **7.3 Conclusion**

Specifically, the chest radiography of a human contains vital organs and could have many different artificial components that target these organs. The researches that are focused on the detection of artificial objects in chest radiography, are mostly limited to the tubes in the mediastinum. It is critical to know that an artificial object can be found

anywhere on the whole chest radiograph, and our proposed model is the first study for automatically detecting artificial objects in the lung region.

### **7.3.1 Other Potential Usage of Method**

It is critical to mention that the output of the our method will be automatically obtained without specifying any region of interest on the x-ray image. It is very important that the proposed model doesn't require any human intervention in the detection process, because of the human intervention could be potential source of error. Because of this feature, our method will be useful for other artificial objects such as other catheter types and stapler wires. Also some pathological patterns may be detectable by using our method.

It should be noted that it is important not only to successfully detect an artificial object but also correctly analyses and evaluate the radiographs that have no chest tube. Adding the JSRT data set, we obtained a larger test set that includes different pathological patterns and chest tube configurations. However, for more robust tests, there is a need for openly accessible, larger datasets that include every type of artificial object and diverse pathological patterns.

### **7.3.2 Future Work**

Although, results are very promising, there are weaknesses of our proposed method that need to be improved further. Some new methods for CNN, offers better performance such as Deep Neural Networks (DNN) architecture [39], which is very popular today.

Also, the multiple chest tube detection problem, requires some modification. It seems to be a simple modification can handle multiple chest tubes, but like every modifications, this modification requires additional tests and improvement cycles.

We do not filter regions of input images. The selection of a Region of Interest (ROI) may increase the performance of the neural network and filter some high-density zones, which could cause some errors.

Testing to detecting other types of artificial objects and / or other radiographic imaging techniques such as Magnetic Resonance (MR) or Computed Tomography (CT) , will

also be a good candidates for future study, including orthopedic implant detection. But, it should be noted that, all of these new implementations require some modifications and new datasets for specific artificial objects.

## **8. ACKNOWLEDGMENTS**

Authors would like to acknowledge the computer resources granted by the High Performance Computing Laboratory of the Informatics Institute at Istanbul Technical University.





## REFERENCES

- [1] **Garcia, M., Jemal, A., Ward, E., Center, M., Hao, Y., Siegel, R. & Thun, M.** (2007). Global Cancer Facts & Figures 2007, Technical Report, American Cancer Society, Atlanta, GA, American Cancer Society.
- [2] **Health Statistics Informatics Department, W.H.O.** (2008). The global burden of disease: 2004 update, Technical Report, Health Statistics and Informatics Department, World Health Organization, Geneva, Switzerland.
- [3] **de Boo, D., Prokop, M., Uffmann, M., van Ginneken, B. & Schaefer-Prokop, C.** (2009). Computed-aided detection (CAD) of lung nodules and small tumours on chest radiographs, *European Journal of Radiology*.
- [4] **Speets, Anouk, M., van der Graaf, Yolanda, Hoes, Arno, W., Kalmijn, Sandra, Sachs, Alfred, P., Rutten, Matthieu, J., Gratama, Jan, van Swijndregt, M., Alexander, D., Mali & Willem** (2006). Chest radiography in general practice: indications, diagnostic yield and consequences for patient management, *British Journal of General Practice*, 56(529), 574–578.
- [5] **van Ginneken, B., ter Haar Romeny, B. & Viergever, M.** (2001). Computer-aided diagnosis in chest radiography: a survey, *IEEE Transactions on Medical Imaging*, 20(12), 1228–1241.
- [6] **Hardie, R.C., Rogers, S.K., Wilson, T. & Rogers, A.** (2008). Performance analysis of a new computer aided detection system for identifying lung nodules on chest radiographs, *Med Image Anal*, 12(3), 240–258.
- [7] **Partain, C.L., Chan, H.P., Gelovani, J.G., Giger, M.L., Izatt, J.A., Jolesz, F.A., Kandarpa, K., Li, K.C.P., McNitt-Gray, M., Napel, S., Summers, R.M. & Gazelle, G.S.** (2005). Biomedical Imaging Research Opportunities Workshop II: Report and Recommendations<sup>1</sup>, *Radiology*, 236(2), 389–403.
- [8] **Armato III, S.G., McLennan, G., McNitt-Gray, M.F., Meyer, C.R., Yankelevitz, D., Aberle, D.R., Henschke, C.I., Hoffman, E.A., Kazerooni, E.A., MacMahon, H., Reeves, A.P., Croft, B.Y. & Clarke, L.P.** (2004). Lung Image Database Consortium: Developing a Resource for the Medical Imaging Research Community<sup>1</sup>, *Radiology*, 232(3), 739–748.
- [9] **Speets, A., Kalmijn, S., Hoes, A., Graaf, Y., Smeets, H. & Mali, W.** (2005). Frequency of Chest Radiography and Abdominal Ultrasound in The Netherlands: 1999–2003, *European Journal of Epidemiology*, 20(12), 1031–1036.

- [10] **Doi, K.** (2007). Computer-aided diagnosis in medical imaging: historical review, current status and future potential., *Comput Med Imaging Graph*, 31(4-5), 198–211.
- [11] **Lo, S.C.B., Lin, J.S.J., Freedman, M.T. & Mun, S.K.** (1998). Application of Artificial Neural Networks to Medical Image Pattern Recognition: Detection of Clustered Microcalcifications on Mammograms and Lung Cancer on Chest Radiographs, *J. VLSI Signal Process. Syst.*, 18(3), 263–274.
- [12] **Schilham, A., van Ginneken, B. & Loog, M.** (2006). A computer-aided diagnosis system for detection of lung nodules in chest radiographs with an evaluation on a public database, *Medical Image Analysis*, 10(2), 247–258.
- [13] **van Ginneken, B., Hogeweg, L. & Prokop, M.** (2009). Computer-aided diagnosis in chest radiography: Beyond nodules, *Eur J Radiol*, 72(2), 226–30.
- [14] **Shiraishi, J., Katsuragawa, S., Ikezoe, J., Matsumoto, T., Kobayashi, T., Komatsu, K.i., Matsui, M., Fujita, H., Kodera, Y. & Doi, K.** (2000). Development of a Digital Image Database for Chest Radiographs With and Without a Lung Nodule: Receiver Operating Characteristic Analysis of Radiologists' Detection of Pulmonary Nodules, *Am. J. Roentgenol.*, 174(1), 71–74.
- [15] **Armato, S.G., Meyer, C.R., Mcnitt-Gray, M.F., McLennan, G., Reeves, A.P., Croft, B.Y., Clarke, L.P. & RIDER Research Group** (2008). The Reference Image Database to Evaluate Response to therapy in lung cancer (RIDER) project: a resource for the development of change-analysis software, *Clin Pharmacol Ther*, 84(4), 448–456.
- [16] **Kuhlman, J.E., Collins, J., Brooks, G.N., Yandow, D.R. & Broderick, L.S.** (2006). Dual-Energy Subtraction Chest Radiography: What to Look for beyond Calcified Nodules, *Radiographics*, 26(1), 79–92.
- [17] **Kobayashi, T., Xu, X.W., MacMahon, H., Metz, C.E. & Doi, K.** (1996). Effect of a computer-aided diagnosis scheme on radiologists' performance in detection of lung nodules on radiographs., *Radiology*, 199(3), 843–848.
- [18] **Arzhaeva, Y., Tax, D. & van Ginneken, B.** (2009). Dissimilarity-based classification in the absence of local ground truth: application to the diagnostic interpretation of chest radiographs, *Pattern Recognition*, 42, 1768–1776.
- [19] **Keller, B., Reeves, A., Cham, M., Henschke, C. & Yankelevitz, D.** (2007). Semi-automated location identification of catheters in digital chest radiographs, *Medical imaging 2007: computer-aided diagnosis*, volume 6514-10, SPIE International Society for Optical Engineering, pp.651410-1–9.
- [20] **Sheng, C., Li, L. & Pei, W.** (2009). Automatic detection of supporting device positioning in intensive care unit radiography, *The International Journal of Medical Robotics and Computer Assisted Surgery*, 5(3), 332–340, <http://dx.doi.org/10.1002/rcs.265>.

- [21] **Ramakrishna, B., Brown, M., Goldin, J., Cagnon, C. & Enzmann, D.** (2011). Catheter detection and classification on chest radiographs: an automated prototype computer-aided detection (CAD) system for radiologists, volume 7963, SPIE, p.796333.
- [22] **Ramakrishna, B., Brown, M., Goldin, J., Cagnon, C. & Enzmann, D.** (2012). An improved automatic computer aided tube detection and labeling system on chest radiographs, *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, volume 8315 of *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*.
- [23] **Mercan, C.A. & Celebi, M.S.** (2010). Fully Automatic Chest Tube Figure Removing from the Postero-Anterior Chest Radiography, *Proceedings of the 11th IASTED International Conference on Computer Graphics and Imaging*, 649-041, IASTED, Innsbruck, Austria.
- [24] **Cernazanu-Glavan, C. & Holban, S.** (2013). Segmentation of Bone Structure in X-ray Images using Convolutional Neural Network, *ADVANCES IN ELECTRICAL AND COMPUTER ENGINEERING*, 13(1), 87–94.
- [25] **Lo, S.C., Lou, S.L., Lin, J.S., Freedman, M., Chien, M. & Mun, S.** (1995). Artificial convolution neural network techniques and applications for lung nodule detection, *Medical Imaging, IEEE Transactions on*, 14(4), 711–718.
- [26] **Lo, S.C.B., Lin, J.S., Freedman, M.T. & Mun, S.K.** (1993). Computer-assisted diagnosis of lung nodule detection using artificial convolution neural network, *Proc. SPIE 1898, Medical Imaging 1993: Image Processing*, 859–869, <http://dx.doi.org/10.1117/12.154572>.
- [27] **LeCun, Y., Bottou, L., Bengio, Y. & Haffner, P.** (1998). Gradient-Based Learning Applied to Document Recognition, *Proceedings of the IEEE*, 86(11), 2278–2324.
- [28] **Browne, M. & Ghidary, S.S.** (2003). Convolutional Neural Networks for Image Processing: An Application in Robot Vision., *Australian Conference on Artificial Intelligence*, volume 2903 of *Lecture Notes in Computer Science*, Springer, pp.641–652.
- [29] **Simard, P., Steinkraus, D. & Platt, J.C.** (2003). Best Practices for Convolutional Neural Networks Applied to Visual Document Analysis, *ICDAR*, IEEE Computer Society, pp.958–962.
- [30] **Lawrence, S., Giles, C.L., Tsoi, A.C. & Back, A.D.** (1998). Face Recognition: A Convolutional Neural Network Approach., *IEEE Transactions on Neural Networks*, 8(1), 98–113.
- [31] **Gill, P., Murray, W. & Wright, M.** (1981). *Practical Optimization*, Elsevier.
- [32] **Petersen, K.B. & Pedersen, M.S.**, (2007), *The Matrix Cookbook*, <http://www2.imm.dtu.dk/pubdb/p.php?3274>, version 20070905.

- [33] **Bishop, C.M.** (1995). *Neural Networks for Pattern Recognition*, Oxford University Press, Inc., New York, NY, USA.
- [34] **LeCun, Y., Bottou, L., Orr, G. & Muller, K.** (1998). Efficient BackProp, *Neural Networks: Tricks of the trade*, volume 1524, Springer, pp.5–50.
- [35] **LeCun, Y.** (1989). Generalization and Network Design Strategies, *Connectionism in Perspective*, Elsevier, Zurich, Switzerland, an extended version was published as a technical report of the University of Toronto.
- [36] **Rogers, D.** (2001). *An introduction to NURBS: with historical perspective*, The Morgan Kaufmann Series in Computer Graphics and Geometric Modeling Series, Morgan Kaufmann Publishers.
- [37] **Fawcett, T.** (2006). An Introduction to ROC Analysis, *Pattern Recogn. Lett.*, 27(8), 861–874, <http://dx.doi.org/10.1016/j.patrec.2005.10.010>.
- [38] **Mercan, C. & Celebi, M.** (2014). An approach for chest tube detection in chest radiographs, *Image Processing, IET*, 8(2), 122–129.
- [39] **Ciresan, D.C., Meier, U. & Schmidhuber, J.** (2012). Multi-column Deep Neural Networks for Image Classification, *Computer Vision and Pattern Recognition*, pp.3642–3649.

## **APPENDICES**

### **APPENDIX A : The base\_ann Class**

## APPENDIX A : THE BASE\_ANN CLASS

Major functions used to construct a CNN, defined at base\_ann class:

```
/// \file mycnn_ann.h
/// \author Cem Ahmet MERCAN
/// \brief Bu dosya base_ann sinifini bildiriyor
#ifndef MYCNN_BASE_ANN
#define MYCNN_BASE_ANN

#include "mycnn_baselayer.h"
#include "mycnn_inputlayer.h"
#include "mycnn_hiddenlayer.h"
#include "mycnn_outputlayer.h"
#include "mycnn_sigmoid_outputlayer.h"
#include "mycnn_ce_sigmoid_outputlayer.h"
#include "mycnn_noact_outputlayer.h"
#include "mycnn_subsampling_hiddenlayer.h"
#include "mycnn_subsampling_outputlayer.h"
#include "mycnn_basenode.h"
#include "mycnn_matrix_array.h"
#include "mycnn_matrix_grid.h"
#include "mycnn_matrix.h"
#include "mycnn_mvector.h"
#include "mycnn_sdlm_hiddenlayer.h"
#include "mycnn_sdlm_outputlayer.h"
#include "mycnn_sdlm_ce_sigmoid_outputlayer.h"

using namespace std;

namespace mycnn {

class base_ann {
public:
    virtual ~base_ann();
    base_ann(int layerCount);
    base_ann(const char *filename);

    void feedForwardLayer(const int layerNo, const int nodeCount,
        const int backLink, const int forwardLink,
        const int outputHeight, const int outputWidth,
        const int stepSizeH = 1, const int stepSizeW = 1,
        const int weightHeight = -1,
        const int weightWidth = -1,
        const int weightBlock = -1);

    void setKernelsAutoInit();
    void setKernelsInit(double randMin, double randMax,
        double randAbsMin);

    void weightRescale();

    void setLearningRate(double eta,
        long int totalEpochCount = 100000);

    void archFeedForward();
    void archFeedForwardSigmoid();
    void archFeedForwardSigmoidCE();
    void archFeedForwardSigmoidCE_SDLM();
};
};
```

```

void archFeedForward_SDLM();
void archFeedForwardNoact();
void archCNN(bool subsampling1);
void archSubsampling();
void archSubsamplingReverse();
void archReverse();
void archFromSets();
void init();

void setFullLink();
void setCNNLink();
bool checkSettings();
virtual void preUpdateLearningRate(bigpic_array & inBP,
    bigpic_array & tarBP, int loop, int * mem);

double updateLearningRate(); ///

```

```

double bigPictureRun(const string & filename , double pgmmin,
                    double pgmmax, bigpic_array & bigIn ,
                    bigpic_array & bigTar , bigpic_array & bigOut ,
                    bool difit = false );

double bigPictureRunFull(const string & filename ,
                        double pgmmin, double pgmmax, bigpic_array & bigIn ,
                        bigpic_array & bigTar , bigpic_array & bigOut ,
                        bool difit = false );

double bigPictureRunO256(const string & filename ,
                        double pgmmin, double pgmmax, int bigInNo ,
                        int bigTargetNo , int bigOutNo, bool difit = false );

void calcD2Net(); ///

```

protected :



```
        baselayer **layers;  
        int LayerCount;  
};  
//end of base_ann class  
  
} // end of name space mycnn  
#endif
```



## CURRICULUM VITAE

**Name Surname:** Cem Ahmet MERCAN

**Place and date of birth:** Sivas, 19.06.1971

**Universities and Colleges attended:** Istanbul University,  
Faculty of Medicine (Capa) 1988-1994

### **Publications:**

- Turna, A., **Mercan, C.A.**, Bedirhan, M.A., 2005 Dec., Prediction of morbidity after lung resection in patients with lung cancer using fuzzy logic. *The Thoracic And Cardiovascular Surgeon*, 53 (6);368-74.
- **Mercan, C.A.** and Celebi, M.S., 2010, Fully Automatic Chest Tube Figure Removing from the Postero-Anterior Chest Radiography, A. Sappa, editor, *Proceedings of the 11th IASTED International Conference on Computer Graphics and Imaging*, 649-041, IASTED, Innsbruck, Austria.
- **Mercan, C.A.**, Çelebi, M.S., 2014 Feb., An approach for chest tube detection in chest radiographs. *Image Processing, IET*, 8 (2); 122-129.

