

---

Doctoral Dissertations

Student Theses and Dissertations


---

Fall 2019

## Applications of machine learning in nuclear imaging and radiation detection

Shaikat Mahmood Galib

Follow this and additional works at: [https://scholarsmine.mst.edu/doctoral\\_dissertations](https://scholarsmine.mst.edu/doctoral_dissertations)

 Part of the [Artificial Intelligence and Robotics Commons](#), [Bioimaging and Biomedical Optics Commons](#), and the [Nuclear Commons](#)

**Department: Mining and Nuclear Engineering**

---

### Recommended Citation

Galib, Shaikat Mahmood, "Applications of machine learning in nuclear imaging and radiation detection" (2019). *Doctoral Dissertations*. 2829.

[https://scholarsmine.mst.edu/doctoral\\_dissertations/2829](https://scholarsmine.mst.edu/doctoral_dissertations/2829)

This thesis is brought to you by Scholars' Mine, a service of the Missouri S&T Library and Learning Resources. This work is protected by U. S. Copyright Law. Unauthorized use including reproduction for redistribution requires the permission of the copyright holder. For more information, please contact [scholarsmine@mst.edu](mailto:scholarsmine@mst.edu).

APPLICATIONS OF MACHINE LEARNING IN NUCLEAR IMAGING AND  
RADIATION DETECTION

by

SHAIKAT MAHMOOD GALIB

A DISSERTATION

Presented to the Graduate Faculty of the

MISSOURI UNIVERSITY OF SCIENCE AND TECHNOLOGY

In Partial Fulfillment of the Requirements for the Degree

DOCTOR OF PHILOSOPHY

in

NUCLEAR ENGINEERING

2019

Approved by

Hyoung Koo Lee, Advisor

Ayodeji B. Alajo

Shoaib Usman

Xin Liu

Donald Wunsch

Copyright 2019

SHAIKAT MAHMOOD GALIB

All Rights Reserved

## ABSTRACT

The main focus of this work is to use machine learning and data mining techniques to address some challenging problems that arise from nuclear data. Specifically, two problem areas are discussed: nuclear imaging and radiation detection. The techniques to approach these problems are primarily based on a variant of Artificial Neural Network (ANN) called Convolutional Neural Network (CNN), which is one of the most popular forms of 'deep learning' technique.

The first problem is about interpreting and analyzing 3D medical radiation images automatically. A method is developed to identify and quantify deformable image registration (DIR) errors from lung CT scans for quality assurance (QA) purposes. The method includes the process of preparing a CT scan dataset for machine learning model training and testing, design of a 3D convolutional neural network architecture that classifies registrations into good or poor classes and a metric called Registration Error Index (REI), which gives us a quantitative measure of registration error. The method achieves 0.882 AUC-ROC on the test dataset. Furthermore, the combined standard uncertainty of the estimated REI by our model lies within 11%, with a confidence level of approximately 68%.

The second problem is about automatic radioisotope identification in real-time. A comparison of five different machine learning models is presented for gamma spectroscopy at a wide variety of testing conditions. Moreover, Hybrid Neural Network (HNN) was developed specifically for gamma-ray spectra data. Three different methods for feature extraction were tested as well. Experiments on MCNP simulated spectra suggest that HNN can achieve 2-12% higher F1 score at difficult testing conditions compared to best performing traditional ML models and obtains 93.33% F1 score during evaluation.

## ACKNOWLEDGMENTS

First and foremost, I would like to thank Almighty for giving me this opportunity to reach this stage of my life.

I am and forever will remain grateful to my advisor, Professor Dr. Hyoung Koo Lee, for guiding me through my doctoral journey. I sincerely thank him for his insights, constant encouragement, advice, feedback, and overall his patience with me despite of numerous disagreements, all along.

I also express my heartfelt gratitude to Dr. Usman, Dr. Alajo, Dr. Liu, Dr. Wunsch, for their continuous support and advice throughout my research period. I thank Dr. Geoffrey Hugo for his invaluable support and advice regarding my project.

I am grateful to my faculty, staff, colleagues, and friends. Special thanks to a few of my friends: Abir, Brendan, Aashiesh, Manish, Yousaf, Palash, Huseyin, Rami, Camila, Mubarak, Saud, and Abdul Alim from the nuclear engineering department, who made my life easier and enjoyable here. I truly appreciate the support and well wishes from my fellow Bangladeshis here, who made this place feel like a home away from home.

I am eternally indebted to my parents for their unconditional love and continuous encouragement. They are true blessings in my life. I am also grateful to my only sister, who took care of my parents back home. My son Ibraheem also deserves a thanks for melting away my stress every day with his innocent smile. Last but not least, I thank my wife Fatema, who was always beside me when I needed her the most. She inspired me to move forward and supported me emotionally and intellectually. She is my true companion.

I dedicate this dissertation to my first love, my mother. Her strength, patience, guidance, wisdom, prayers, and endless love has made me who I am.

## TABLE OF CONTENTS

	Page
ABSTRACT .....	iii
ACKNOWLEDGMENTS .....	iv
LIST OF ILLUSTRATIONS .....	ix
LIST OF TABLES .....	xi
NOMENCLATURE .....	xii
 SECTION	
1. INTRODUCTION .....	1
1.1. MACHINE LEARNING .....	1
1.2. DEEP LEARNING .....	2
1.3. DISSERTATION OVERVIEW .....	3
2. RELATED WORKS ON MACHINE LEARNING .....	6
2.1. ARTIFICIAL NEURAL NETWORK .....	6
2.1.1. Fully Connected Neural Network .....	7
2.1.1.1. Training algorithm - backpropagation .....	8
2.1.1.2. Activation functions .....	9
2.1.2. Recurrent Neural Network .....	9
2.1.3. Convolutional Neural Network .....	10
2.1.3.1. Convolutional layer .....	11
2.1.3.2. Activation layer .....	11

2.1.3.3.	Pooling layer .....	12
2.1.3.4.	Fully connected layer.....	13
2.1.3.5.	Loss layer .....	13
2.1.3.6.	Hyperparameters .....	14
2.1.3.7.	Regularization .....	15
2.2.	GRADIENT BOOSTED DECISION TREES .....	15
3.	QUALITY ASSURANCE OF MEDICAL IMAGE REGISTRATION WITH CONVOLUTIONAL NEURAL NETWORK.....	17
3.1.	BACKGROUND AND SECTION OUTLINE .....	17
3.2.	RELATED WORKS .....	19
3.3.	MATERIALS AND METHODS.....	22
3.3.1.	Training Dataset: DIRLAB 4DCT Dataset .....	22
3.3.2.	Testing Dataset: 4D Lung CT Dataset.....	22
3.3.3.	Quality Assurance Algorithm.....	23
3.3.3.1.	Image pre-processing.....	25
3.3.3.2.	Training dataset construction .....	25
3.3.3.3.	Neural network architecture.....	26
3.3.3.4.	Training method .....	27
3.3.4.	Experiments .....	28
3.3.4.1.	Registration .....	28
3.3.4.2.	Landmarks .....	29
3.3.4.3.	Train and test set generation .....	29
3.3.4.4.	Implementation .....	31
3.4.	RESULTS .....	32
3.5.	DISCUSSION .....	34
3.6.	REMARKS .....	38

4. AUTOMATED RADIOISOTOPE DETECTION USING HYBRID NEURAL NETWORKS .....	40
4.1. BACKGROUND AND SECTION OUTLINE .....	40
4.2. RELATED WORKS .....	42
4.3. MATERIALS.....	44
4.3.1. Dataset Description.....	44
4.3.2. Radioisotope Types .....	45
4.4. METHODS .....	46
4.4.1. Training Samples Generation .....	46
4.4.2. Feature Selection .....	47
4.4.2.1. Spectrum counts (SC) .....	48
4.4.2.2. Discrete cosine transform (DCT) .....	48
4.4.2.3. Peak ratios (PR) .....	49
4.4.3. Classifiers .....	50
4.4.3.1. Fully-connected neural network (FCNN) .....	50
4.4.3.2. Recurrent neural network (RNN) .....	51
4.4.3.3. Hybrid neural network 1 (HNN1) .....	51
4.4.3.4. Hybrid neural network 2 (HNN2) .....	52
4.4.3.5. Gradient boosted decision trees (GBDT).....	53
4.4.4. Training and Validation .....	54
4.4.5. Testing .....	55
4.4.6. Evaluation Metric .....	55
4.5. RESULTS .....	57
4.5.1. Effect of Train Dataset Size .....	57
4.5.2. Effect of Spectra Acquisition Time .....	57
4.5.3. Run Time .....	59
4.5.4. Multi-class Classification Performance.....	60



4.5.5. Effect of Data Augmentation.....	63
4.6. REMARKS .....	63
5. CONCLUSIONS .....	66
APPENDICES	
A. RADIOISOTOPE DETECTION PERFORMANCE IN TABLES.....	68
B. PYTHON CODE FOR RADIOISOTOPE DETECTION ML MODELS .....	72
C. PUBLICATIONS .....	76
REFERENCES .....	78
VITA.....	88

## LIST OF ILLUSTRATIONS

Figure	Page
1.1. A relational diagram of Artificial Intelligence (AI), Machine Learning (ML) and Deep Learning (DL) .....	3
2.1. Schematic diagram of an artificial neuron or a 'perceptron' .....	7
2.2. A simplified graph of classical fully connected neural network .....	8
2.3. Some example activation functions that are commonly used in artificial neural networks (datawow.io, 2019) .....	9
2.4. A simplified graph of recurrent neural network .....	10
2.5. An example of generating feature-maps from input data using a convolutional kernel .....	12
2.6. An example of ReLU activation function .....	12
2.7. An example of downsampling operation by max-pooling technique .....	13
2.8. Activations of a convolutional neural network at different layers (stanford.edu, 2019 (accessed July 7, 2019) .....	14
2.9. A simplified graph of a decision tree .....	16
3.1. Registration result in one pair of corresponding slices from CT scans of two patients. ....	18
3.2. Training sample preparation. ....	26
3.3. Neural network architectures. model A - from this work, model B - reference (Eppenhof and Plum, 2018) .....	28
3.4. Example of training sub-volume patches.....	30
3.5. Histogram of registration accuracy (mm), DIRLAB and VCU dataset. ....	31
3.6. AUC-ROC curve for model A and B on DIRLAB and VCU test data.....	32
3.7. Ground truth REI vs. Estimated REI on VCU dataset. ....	33
3.8. REI estimation. ....	35
4.1. Block diagram of a conventional radioisotope detection algorithm .....	43
4.2. Schematic diagram of the MCNP simulation (NNSA, 2019) .....	45

4.3. Simulated spectrum of the five radioisotopes (HEU, WGPu, I-131, Co-60, Tc-99m) using a NaI(Tl) detector. ....	47
4.4. The process of training spectrums generation from time-series detector data ....	48
4.5. Schematic diagram of (a) FCNN and (b) RNN model .....	51
4.6. Schematic diagram of hybrid neural network architectures .....	52
4.7. Schematic diagram of GBDT model in leaf-wise growth method .....	54
4.8. Schematic diagram of the overall experimental setup .....	54
4.9. Multi-window prediction and voting strategy for radioisotope identification .....	56
4.10. F1 score on test dataset predicted by five ML models, for feature set SC (a), DCT (b), and PR (c) on different training percentages .....	58
4.11. Example spectra of HEU+Tc-99m at 1s, 5s and 20s acquisition time .....	59
4.12. F1 score on test dataset predicted by five ML models, for feature set SC (a), DCT (b), and PR (c) on different spectra acquisition times. ....	60
4.13. Run-time of the five models using 5s spectra acquisition time. (a) training time, (b) testing time with increasing train data percentages.....	61
4.14. F1 score on test data for 7-class classification by five models using 5s spectrums and SC feature set .....	62
4.15. Confusion matrix for 7-class classification by HNN1 model using 5s spectrums.	63
4.16. F1 score for 4-category (background (0), SNM (1), medical (2) and industrial (3)) radioisotope identification using 5s spectrums at test dataset. ....	64

**LIST OF TABLES**

Table	Page
3.1. Effect of data augmentation on VCU test set .....	33
4.1. Number of radiological search data files in the simulation dataset .....	46
4.2. Peak energies considered for PR feature calculation .....	50
4.3. Parameter optimization for GBDT model .....	53
4.4. Effect of data augmentation for HNN1 model on test dataset .....	64

**NOMENCLATURE**

*ANN* Artificial Neural Network

*CNN* Convolutional Neural Network

*DCT* Discrete Cosine Transform

*DL* Deep Learning

*GBDT* Gradient Boosted Decision Trees

*LSTM* Long Short-Term Memory

*ML* Machine Learning

*PR* Peak Ratios

*QA* Quality assurance

*REI* Registration Error Index

*RNN* Recurrent Neural Network

# 1. INTRODUCTION

## 1.1. MACHINE LEARNING

Machine learning (ML) is a field of study that gives computers the capability to learn without being explicitly programmed. New computing technologies and availability of the significant amount of digital data created the opportunity to develop more advanced ML methods than the past. While many of the algorithms have been around for an extended period, the ability to apply complex mathematical models to big data, over and over, faster and faster, is a relatively recent development. ML algorithms are useful where it is difficult to program a mathematical problem straightforwardly. Some of the ML applications that have seen success in recent years include computer vision, natural language processing, medical diagnosis, self-driving cars, customer demand prediction, fraud detection, email filtering, and others.

The term machine learning was first used by Arthur Samuel (Samuel, 1959). Tom Mitchell (Mitchell, 1997) provided a formal definition of ML: "A computer program is said to learn from experience  $E$  with respect to some class of tasks  $T$  and performance measure  $P$  if its performance at tasks in  $T$ , as measured by  $P$ , improves with experience  $E$ ." This is more of an operational definition in nature than cognitive reasoning. This also represents the same idea as Alan Turing suggested in his article (TURING, 1950) that the question "Can machine think?" should be replaced with "Can machines do what we can do?". This implies that the ML concept is mostly learning from experiences rather than building an actual "thinking" machine.

ML algorithms build mathematical models that are based on some sample data, known as "training data." When the correct label of the training data is known in prior, the models can be trained in a supervised manner, otherwise can also be trained in an

unsupervised fashion. Some of the most commonly used machine learning algorithms include Naive Bayes, Linear/Logistic Regression, Artificial Neural Network, Support Vector Machine, K-Nearest Neighbour, Decision Tree, Random Forest, K-means Clustering and others. These algorithms are general-purpose methods and work on various representations of data.

Based on the pattern of training data, different ML models with different capabilities were developed in the past. Traditionally, the most common type of data found in ML applications is in *structured* format, meaning data that are highly organized in a relational manner such as, data presented in a spreadsheet logically, is an example of *structured* data. On the other side, *unstructured* data type is everything else that is not *structured*, including image, video or audio data. This kind of data is information-rich, but not as organized as *structured* data, For computers, *structured* data is far more easier to understand than *unstructured* data cases.

## 1.2. DEEP LEARNING

Deep learning (DL) is a form of machine learning that takes advantage of modern computing hardware in order to train large ML models using a significant amount of training data. DL methods are based on artificial neural networks, and recently achieved high performance in *unstructured* data cases (i.e., computer vision, language processing related problems). With the availability of sufficient amount of training data, DL methods can achieve near human-level accuracy at specific prediction tasks, including, image recognition (Russakovsky *et al.*, 2015), which would be otherwise challenging with a traditional ML algorithm. DL methods also require much less pre-processing of data, which makes them an attractive option as an ML technique.

Though DL methods are promising techniques, they require a vast amount of train data to generalize sufficiently well. In contrast, traditional ML algorithms, such as Gradient Boosted Decision Trees (GBDT) or Support Vector Machine (SVM) are simpler, more

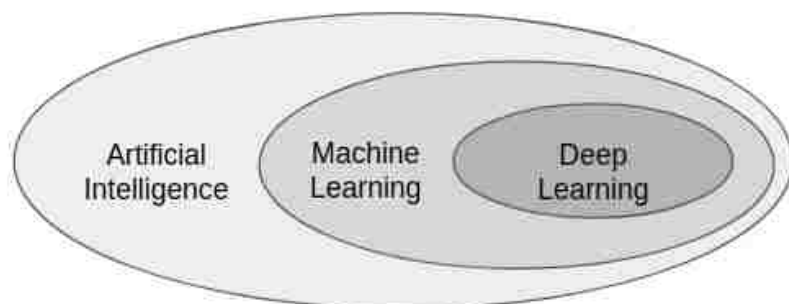


Figure 1.1. A relational diagram of Artificial Intelligence (AI), Machine Learning (ML) and Deep Learning (DL)

explainable methods than DL, and can work well with a little amount of train data. These methods are suitable for *structured* data cases, and widely used in real-world applications. Figure. 1.1 shows a relational diagram between Artificial Intelligence (AI), ML and DL fields.

### 1.3. DISSERTATION OVERVIEW

This dissertation presents two areas of ML applications that address both *unstructured* and *structured* data problems. The *unstructured* data case is analyzing nuclear images and the *structured* case is about automated radiation detection from gamma-spectra.

The first study presents an automatic algorithm for Quality Assurance (QA) on lung CT scan images. For example, if a patient is going through radiation treatment, CT scans are taken at different times in order to compare the changes happened inside the lung, because of the treatment. When two such CT images are brought together into alignment for comparison, that process is known as Deformable Image Registration (DIR). Computer algorithms are usually used to perform automated DIR. DIR algorithms do not produce acceptable results at every time, requiring a QA tool that is capable of identifying DIR errors. We have presented a method to identify registration errors made by DIR algorithms, using the DL method. Convolutional neural networks were employed to train the model. We



show that, even with a limited amount of train data, DL methods show promising results in challenging conditions. Moreover, we show a method to quantify registration error, called Registration Error Index (REI). The method developed can be applied to image registration problems from other domains or even for other body-parts.

The second study presents an ML method for automated detection and identification of radioisotopes using gamma-ray spectra. Gamma spectrum data is usually stored in an organized manner and can be viewed as *structured* data. Moreover, spectrum peak energies are sequentially related to each unique radioisotope. We have developed ML methods that maximize the use of information from gamma spectra by using hybrid neural networks. We have compared five ML models at different conditions and show that algorithms designed to address problems with a distinct pattern, such as gamma spectra, can produce more accurate and reliable results than traditional generalized methods.

The dissertation is organized in the following sections:

1. Related Works on Machine Learning: In this section, relevant ML algorithms for this dissertation study cases are described, More importantly, Convolutional Neural Network (CNN) is discussed in detail, along with other Artificial Neural Network methods, such as, Fully Connected Neural Network (FCNN) and Recurrent Neural Network (RNN). Apart from DL methods, a traditional decision tree based method, GBDT is also discussed.
2. Quality Assurance of Medical Image Registration with Convolutional Neural Network: This section describes the method of registration QA algorithm. More specifically, the process of training dataset construction from medical 3D images, method of training a 3D CNN, a method for error quantification and discussions on the advantages and limitations of the algorithm is presented.

3. Automated Radioisotope Detection using Hybrid Neural Networks: This section discusses some traditional methods of automated radioisotope detection algorithms, their limitations, and hypothesis on improving the methods. Next, we have described the method of designing hybrid neural networks for gamma spectra data. Further, we have presented a comparative study of three traditional methods and two newly developed hybrid methods.
4. Conclusion: Finally, we conclude the dissertation by summarizing the findings in the studies.

## 2. RELATED WORKS ON MACHINE LEARNING

This section describes necessary backgrounds on several machine learning methods that would make this dissertation self-contained. More importantly, Artificial Neural Networks (ANNs) are described in detail. Among ANN methods, Fully-connected Neural Network (FCNN), Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN) are discussed. Other supervised ML methods, such as Gradient Boosted Decision Tree (GBDT) is also reviewed.

### 2.1. ARTIFICIAL NEURAL NETWORK

Artificial Neural Network (ANN) is one of the most widely used machine learning methods in recent years. ANNs have been successfully applied in many disciplines, such as computer vision, language processing, time-series prediction, anomaly detection, fraud detection, weather prediction, playing board and video games, and many more.

An artificial neuron, commonly referred to as a "perceptron" (Rosenblatt, 1961), is a computational unit that receives information from another perceptron, manipulates it and pass the processed information to the next perceptron. When several perceptrons are connected systematically, they form a network-like structure. With the help of an optimization and training algorithm, the network can memorize or generalize patterns from complex data representations. Typically, artificial neurons are connected in layers. These connections are called 'edges' and edges have weights that adjust during a training process. The term 'training' is also referred to as 'learning' in some literature. Figure. 2.1 shows a schematic diagram of a perceptron and its connections.

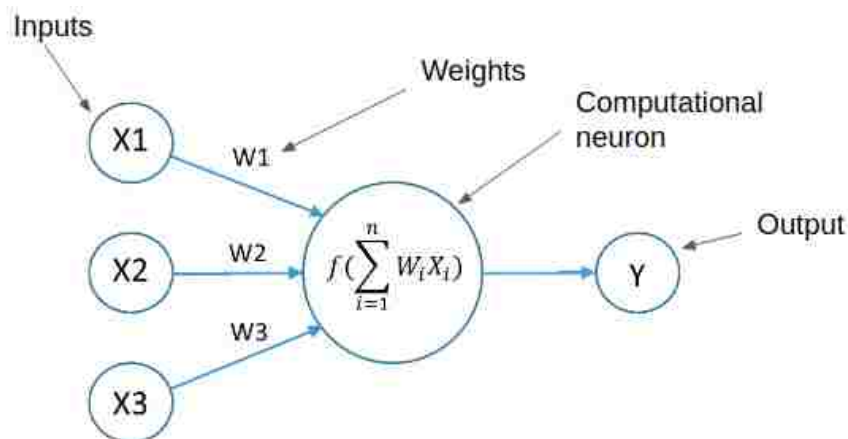


Figure 2.1. Schematic diagram of an artificial neuron or a 'perceptron'

There are several varieties of ANNs that are applied in practice to address different kinds of machine learning problems. Majority of the ANN approaches are supervised in nature, meaning they need a labeled dataset for training. Fully-connected Neural Network (FCNN), Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN) are some of the most common ones and described in the following sections.

**2.1.1. Fully Connected Neural Network.** Fully-connected Neural Network (FCNN) is the classical form of neural networks. Artificial neurons are arranged in a layer-wise manner, and each neuron in one layer connects to all neurons in the next layer. An FCNN with only one hidden layer can approximate any non-linear function. Typically, there is one input layer, one output layer, and there can be multiple hidden layers. Hidden layers produce an abstract representation of input vectors, which is more meaningful to the output layer. With the increasing number of neurons and the hidden layers, the number of connections and learnable parameters increases significantly. FCNNs can distinguish data that is not linearly separable. FCNN was a popular machine learning solution in the 1980s, but after that, they faced stiff competition from a much simpler method, Support Vector Machine (SVM). Figure. 2.2 shows a schematic diagram of a classical FCNN and its weight connections.

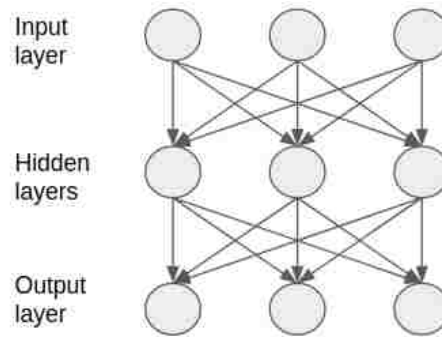


Figure 2.2. A simplified graph of classical fully connected neural network

**2.1.1.1. Training algorithm - backpropagation.** Backpropagation (BP) is a technique to train ANNs efficiently by following a gradient descent approach. Gradient descent calculates the gradient of the error with respect to the networks modifiable weights. BP exploits the chain rule in a recursive manner and updates weights of a neural network iteratively until it can perform the task for what it is being trained. BP works in four steps:

1. Forward pass
2. Loss calculation
3. Backward pass
4. Weights update

Gradient of error between two neurons,  $i$  and  $j$ , compared to the true label can be estimated by,

$$\frac{\partial E_p}{\partial w_{ji}} = \frac{\partial E_p}{\partial o_{pj}} \frac{\partial o_{pj}}{\partial w_{ji}} \quad (2.1)$$

where,  $E_p$  is the loss function that is optimized during training,  $w_{ji}$  is the weight between  $i$  and  $j$  and  $o_{pj}$  is the predicted value of target  $p$ .

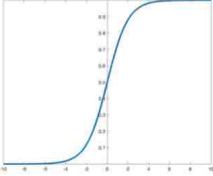
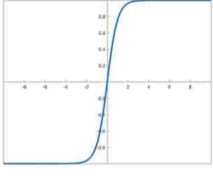
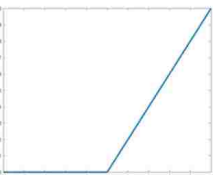
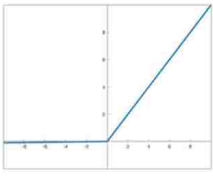
Activation function	Equation	Graph
Sigmoid	$S(x) = \frac{1}{1 + e^{-x}}$	
Tanh	$\tanh x = \frac{e^x - e^{-x}}{e^x + e^{-x}}$	
ReLU	$RELU(x) = \begin{cases} 0 & \text{if } x < 0 \\ x & \text{if } x \geq 0 \end{cases}$	
Leaky ReLU	$f(x) = \begin{cases} x & \text{if } x > 0 \\ 0.01x & \text{otherwise} \end{cases}$	

Figure 2.3. Some example activation functions that are commonly used in artificial neural networks (datawov.io, 2019)

**2.1.1.2. Activation functions.** Activation functions are used to calculate the output in a layer of a neural network. These functions usually scale the output of a neuron between 0 to 1 or reduce the complexity of numbers by discarding some less important values. Some example activation functions are presented in Figure. 2.3.

**2.1.2. Recurrent Neural Network.** The idea behind the Recurrent Neural Network (RNN) is to make use of sequential information. Traditional FCNN assumes that all inputs are independent of each other. But for many tasks such as speech recognition or text understanding, consideration of sequential relationship is very important for building predictive models. Unlike FCNNs, RNNs have temporal connection between neurons which

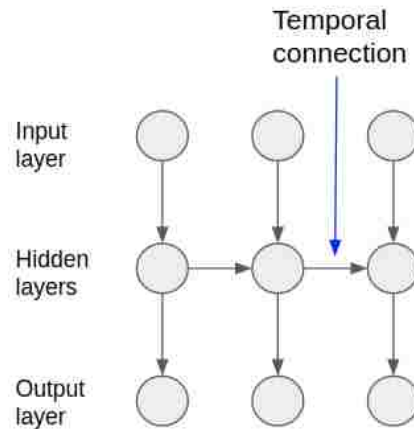


Figure 2.4. A simplified graph of recurrent neural network

makes them suitable for exploiting sequence relationship. RNNs have shown great success in Natural Language Processing (NLP) tasks (Graves, 2013; Graves *et al.*, 2014; Hermann *et al.*, 2015a; Mikolov *et al.*, 2013; Sutskever *et al.*, 2014) in the recent years. Figure. 2.4 shows a schematic diagram of a RNN and its connections.

**2.1.3. Convolutional Neural Network.** Convolutional neural network (CNN) is a class of deep neural network that have revolutionized many areas of machine learning including computer vision such as image classification (Goodfellow *et al.*, 2013; He *et al.*, 2016a,b; Hinton and Salakhutdinov, 2006; Krizhevsky *et al.*, 2012; Sermanet *et al.*, 2013; Szegedy *et al.*, 2016), object detection (Girshick *et al.*, 2016; He *et al.*, 2014; Redmon *et al.*, 2016; Ren *et al.*, 2015), image segmentation (Chandra and Kokkinos, 2016; Chen *et al.*, 2017; Jegou *et al.*, 2017; Kayalibay *et al.*, 2017; Long *et al.*, 2015; Moeskops *et al.*, 2017; Peng *et al.*, 2017; Zhao *et al.*, 2017) and speech recognition (Kalchbrenner *et al.*, 2014) related problems.

CNNs are regularized version of multi-layer perceptrons and primarily successful in unstructured data cases such as image, video or audio related problems. CNNs can extract abstract representations of input data that contains spatial or temporal relationship, and they are also known as 'shift-invariant' or 'space invariant' neural networks. Convolutional networks require many training data to generalize well.

CNNs usually needs little preprocessing compared to other image recognition algorithms. The network learns features by convolution operation, which otherwise would be hand-engineered. CNNs can replace the process of extracting hand-engineered features, which makes them an attractive option for data sets with an intricate pattern or sequence. Below the building blocks of a CNN are discussed.

**2.1.3.1. Convolutional layer.** Convolutional layer is the most important building block of a CNN. This layer consists of learnable filters (or kernels) that produce an activation feature map of the input volume. The filters convolve across the input space and compute the activation maps by taking a dot product between the filter and the input receptive field. Finally, the learned filters activate when it sees some similar pattern in the input data and detects the presence of a signal. By stacking all the activation maps next to each other, several abstract representations of the input data can be obtained. Figure. 2.5 shows the process of generating feature maps using convolutional kernels.

**2.1.3.2. Activation layer.** ReLU, abbreviated as Rectified Linear Unit, is the most commonly used activation function for CNNs. It's a function such that,

$$f(x) = \max(0, x) \quad (2.2)$$

ReLU effectively removes the negative entries from an activation map and sets them to zero. It is a non-linear activation and usually preferred for convolutional network training because they are several times faster than other activation functions such as hyperbolic tangent, or sigmoid and yet does not compromise much of the generalization ability. Figure. 2.6 shows the operation of ReLU activation function.



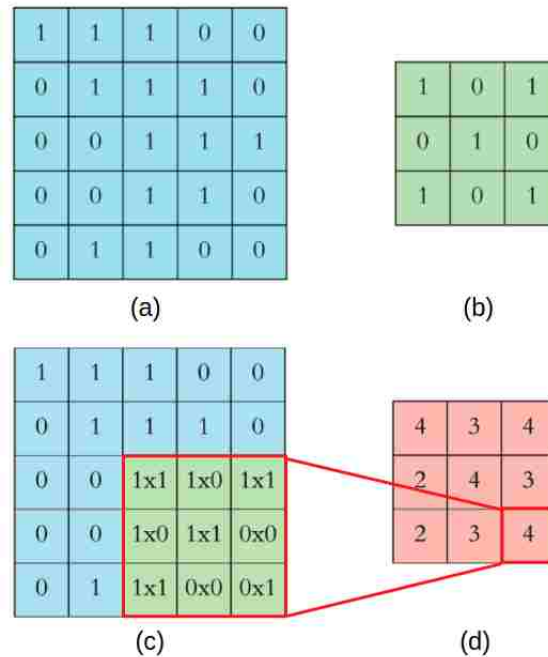


Figure 2.5. An example of generating feature-maps from input data using a convolutional kernel

**2.1.3.3. Pooling layer.** Pooling is another important concept in CNNs. Pooling layer downsamples data non-linearly by selecting a segment/sub-region from the input and taking the maximum or average from that segment. The segments are generally non-overlapping, and max-pooling is one of the common forms of the pooling operation. The

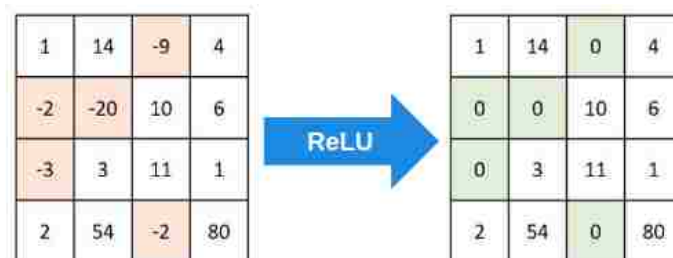


Figure 2.6. An example of ReLU activation function

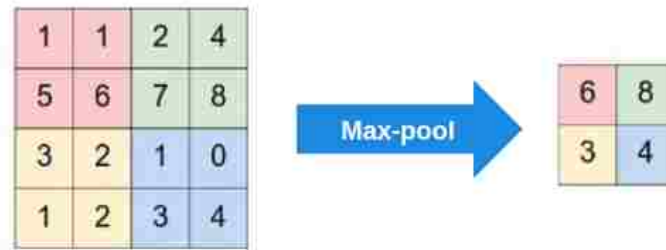


Figure 2.7. An example of downsampling operation by max-pooling technique

main purpose of pooling operation is to gradually downsample the input data so that the filters/kernels can operate at different resolutions of the data. Pooling layers can operate independently on the activation maps.

An important advantage of pooling layer is that it gradually reduces the input space, making the network operate in a decreased spatial size. This helps in learning more kernels at lower trainable parameters and memory cost. Traditionally, A CNN is trained by inserting pooling layers between successive convolutional layers. Figure. 2.7 shows an example of max-pooling operation.

**2.1.3.4. Fully connected layer.** Fully connected layers of neurons can be employed after several convolutions, activation, and pooling layers. These fully connected layers learn from the final convolutional representations of input data and are similar to FCNN architecture.

**2.1.3.5. Loss layer.** 'Loss layer' is usually the final or output layer of a neural network. It specifies how a training algorithm penalizes the difference between the predicted and true label. A loss function estimates the error at each training iteration, and the training algorithm works on minimizing the loss. Some commonly used loss functions include cross-entropy (CE), mean squared error (MSE), hinge, and others. Loss functions are chosen based on classification requirements. Softmax and sigmoid are the two commonly used activation functions for the loss layer. Figure. 2.8 shows the activation maps of the input at different layers of a CNN.

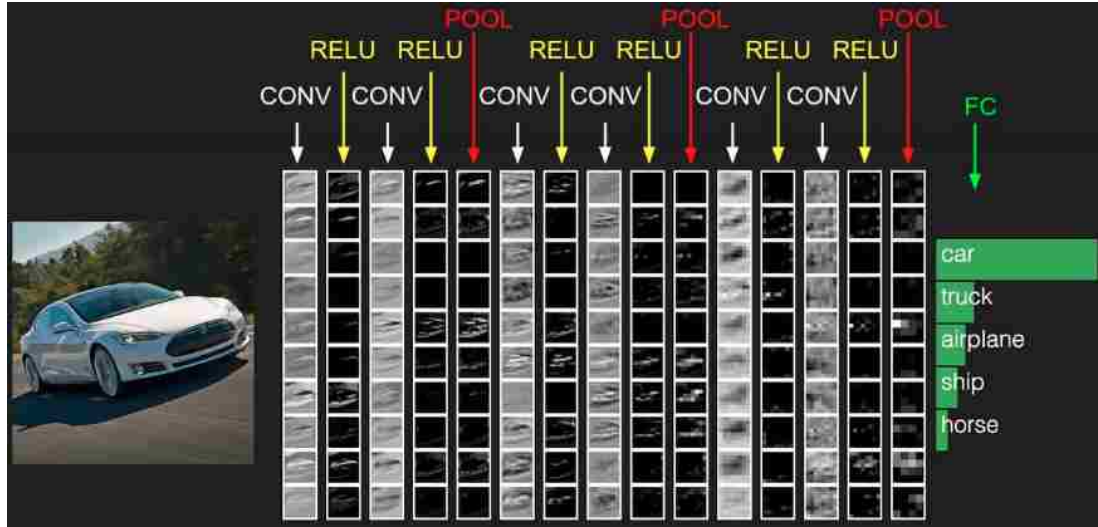


Figure 2.8. Activations of a convolutional neural network at different layers (stanford.edu, 2019 (accessed July 7, 2019))

If a vector of  $n$  predictions generated from a sample of  $n$  data points, and  $Y$  is the vector of observed values of the variable being predicted, then the within-sample MSE of the predictor is computed as,

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2 \quad (2.3)$$

The standard softmax function  $\sigma : \mathbb{R}^K \rightarrow \mathbb{R}^K$  is defined by the formula,

$$\sigma(\mathbf{z})_i = \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}} \text{ for } i = 1, \dots, K \text{ and } \mathbf{z} = (z_1, \dots, z_K) \in \mathbb{R}^K \quad (2.4)$$

**2.1.3.6. Hyperparameters.** The parameters that need to be defined before training a neural network are commonly termed as hyperparameters. For CNNs, some of the hyperparameters include the number of filters, filter size, number convolution layers, number of pooling layers and shape, connection type between layers, and some others. These parameters usually depend on dataset type and size and need empirical trials for finding a working set of hyperparameters.

**2.1.3.7. Regularization.** To successfully train a neural network, it often needs regularization to prevent overfitting. Some of the regularization methods include dropout (Srivastava *et al.*, 2014a), drop-connect (Wan *et al.*, 2013), L1 or L2 regularization and others. Dropout works by simply not updating the weights of a certain percentage of neurons at each iteration during training. This method prevents learning every possible pattern from input data, which reduces the possibility of overfitting.

Another successful method of increasing generalization of the neural network is data augmentation. Augmentation is the act of creating more training data by applying simple transformations/shift to the original data (i.e., random crop, translation, rotation, brightness shift, contrast shift, and among others). This technique works as a regularizer as neural network sees more data from different perspectives and tries to generalize among all the training data.

## **2.2. GRADIENT BOOSTED DECISION TREES**

Gradient boosted decision trees (GBDT) is one of the most popular machine learning algorithms for regression and classification tasks in recent years. GBDT achieves state-of-the-art results in the prediction tasks where data is in 'structured' form, meaning data is presented in a highly organized tabular format (unlike image, speech, or text).

Gradient boosting is based on the idea that an ensemble of weak learning models, such as decision trees, can produce better results than a single model. It builds the model in a stage-wise fashion and generalizes by optimizing a loss function. Figure 2.9 shows a simplified schematic diagram of a decision tree.

A weak hypothesis or a weak learner is defined as one that predicts at least slightly better than random choice. Hypothesis boosting is the idea of filtering observations, leaving the simple observations that a weak learner can handle, and further focuses on more difficult observations. The first realization of boosting algorithm was AdaBoost, an acronym for

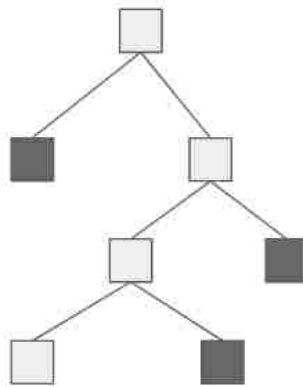


Figure 2.9. A simplified graph of a decision tree

Adaptive Boosting (Freund *et al.*, 1999). The weak learners in AdaBoost were decision trees, called decision stumps. Predictions were made by a majority vote of weak learners, weighted by their accuracy. AdaBoost was further generalized by treating it as a numerical optimization problem, where the objective was to minimize the loss using a gradient descent like procedure.

In general, GBDT is an algorithmic approach to split a dataset based on different conditions (i.e., if-else-then). The more complex the data pattern, the GBDT algorithm finds more ways to assign conditions on the data. These models do not consider any sequential relationship between input data points. Some of the popular gradient boosting algorithm using decision trees include Light Gradient Boosting Machine (LightGBM) (Ke *et al.*, 2017), CatBoost (Prokhorenkova *et al.*, 2018) and XgBoost (Chen and Guestrin, 2016) methods. There are several parameters in a GBDT model that needs to be defined to prevent overfitting on data, such as number of trees, tree depth, number of leaves, number of observations per split, minimum improvement to loss are some of the most important ones.

### 3. QUALITY ASSURANCE OF MEDICAL IMAGE REGISTRATION WITH CONVOLUTIONAL NEURAL NETWORK

#### 3.1. BACKGROUND AND SECTION OUTLINE

Image registration, the process of identifying corresponding features between two images in order to bring the images into alignment, has become a critical basic task in radiotherapy. Many applications rely on image registration, including image guidance, adaptive radiotherapy, multi-modality segmentation, and image-based assessment of therapy response.

Deformable image registration (DIR) is moving into the clinic for automated calculation of cumulative delivered dose for assessing the quality of treatment. However, regardless of which DIR algorithm is used for assessing cumulative dose, there is a lack of useful quality assurance (QA) tools and processes. Phantoms and publicly-available datasets are useful for algorithm validation and benchmarking, but cannot be used to directly infer the quality of an individual patient registration in the clinic. Deformably-mapped contour comparison and landmark analysis are useful, but also time-consuming and therefore not appropriate for first-line QA. Many clinics, therefore, fall back on manual, visual review of registration results. However, time and clinical resources are limited for such manual, intensive review of every registration performed for every patient. Instead, a first-line QA process is desired that can automatically assess the quality of the registration, and flag questionable results for in-depth review by a physicist, physician, or other experts.

One of the challenges in registration QA is to develop an evaluation method that describes the true underlying registration quality, rather than the noise (Pluim *et al.*, 2016). Various image similarity measure algorithms like the sum of squared difference (SSD), mutual information (MI), correlation ratio or least squares are found in the literature to express the quality of registration. However, the similarity of intensity is not an appropriate

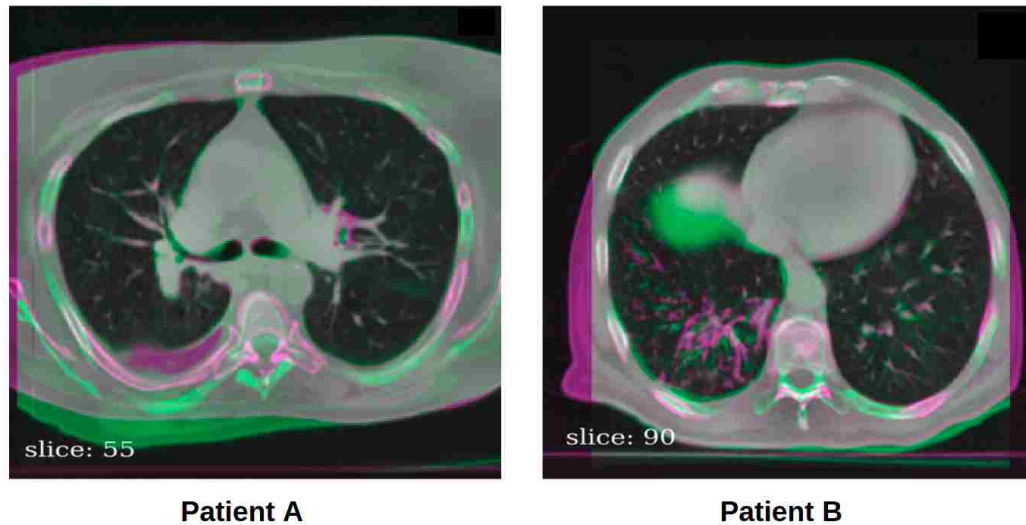


Figure 3.1. Registration result in one pair of corresponding slices from CT scans of two patients. The colors magenta and green are used in a complementary color overlap to represent matching quality in two corresponding slices after registration. Here, the registration result in the lung parenchyma for patient A might be acceptable because of the good alignment in the soft tissue region, even though there is misalignment in the bony structures. Conversely, the registration result for patient B might not be acceptable in the parenchyma as the soft tissues did not align well. A registration QA tool should be capable of identifying mismatched regions in both registration settings.

measure of accuracy (Rohlfing, 2012). To quantify local registration accuracy measurement, researchers have introduced and utilized the concept of target registration error (TRE) (Datteri and Dawant, 2012; Eppenhof and Pluim, 2018; Fitzpatrick and West, 2001; Fitzpatrick, 2009; Gierga *et al.*, 2008; Latifi *et al.*, 2013; Ma *et al.*, 2007; Moghari *et al.*, 2008; Neylon *et al.*, 2017; Seginer, 2011) or landmark registration error (LRE) (Muenzing *et al.*, 2012). One way to approximate TRE in the non-rigid setting is to measure the physical distance between corresponding landmark points in the registered image pair, also commonly known as landmark distance error (LDE) (Fischer *et al.*, 2010). Landmarks can be manually annotated by experts or semi-automatically generated (Murphy *et al.*, 2011a). Therefore, a quality measure derived from TRE is a more preferred method for expressing registration error as it considers physical/anatomical misalignment. Figure. 3.1 shows two examples of misalignment after registration between an image pair.

This work describes the feasibility of using a deep learning method to flag suspicious registration results for review by an expert or for re-registration. The goal is to develop a registration QA approach which can be used to infer registration quality across important regions of the anatomy rapidly. Here, we have used 3D convolutional neural networks (3D-CNN) to estimate registration error inside the lung parenchyma of pulmonary CT scans by classifying sub-regions (patches) from a full CT scan into acceptable or unacceptable classes of registration quality. In this work, we propose three major contributions: (a) a method to prepare the dataset for registration accuracy evaluation, (b) a 3D-CNN architecture that is computationally efficient and fast, and (c) a method to quantify registration error called registration error index (REI). In the following sections, we describe relevant works, methods of our algorithm, experimental procedure, and evaluation.

### 3.2. RELATED WORKS

Several methods have been reported in the literature to quantify registration errors in medical images. Some of them serve as general purpose algorithms, while others are problem specific. This previous work in automatic image registration quality assessment can be divided roughly into unsupervised and supervised approaches. In the unsupervised approach, an assessment of quality is made based on the data without using a learning method. In the supervised approach, a training dataset with a known estimation of registration quality is used to train a model to infer registration quality in an unseen, new dataset.

Unsupervised approaches include that by Datteri et al. (Datteri *et al.*, 2015), who proposed a method that uses a 'registration circuit' to measure the quality of registration. They have claimed that their measure of registration quality correlates well with true target registration error (TRE), and can be used to reduce registration errors in different algorithms. Additionally, Saygili et al. (Saygili *et al.*, 2016) proposed a method that does not require ground truth to estimate registration error and is independent of the registration algorithm



used. They have extracted various image features inside a voxel to match local patterns between the registered image pairs. They have reported a good agreement between local errors with their estimated error. Furthermore, Saleh et al. (Saleh *et al.*, 2014) proposed a DIR performance metric called the distance discordance metric (DDM), that measures registration uncertainty using a statistical sampling technique on deformation vector fields (DVF). The main limitation of this method is that it requires at least four registered images for the calculation.

In supervised approaches, Neylon et al. (Neylon *et al.*, 2017) reported a method that estimates DIR performance using a 3-layer deep neural network. They created a simulated head-neck dataset with ground truth annotations. Image similarity metrics (ISM) were calculated and fed into the neural network to estimate TRE. They have reported 95% accuracy within a single voxel to quantify DIR error.

Supervised registration quality estimation specifically for lung CT scans, has been studied as well. For example, Muenzing et al. (Muenzing *et al.*, 2012) proposed a supervised machine learning approach to estimate lung CT registration errors from statistical image features. Features were calculated from manually annotated landmark points to distinguish local alignment patterns. They classified the registration quality into three classes: good, poor and wrong alignment and reported 90% accuracy. They have used this method to further boost the performance of existing DIR algorithms (Muenzing *et al.*, 2014). Another approach proposed by Sokooti et al. (Sokooti *et al.*, 2016) is a regression forest based method to estimate registration errors in synthetically deformed chest CT scans. The authors have shown that regression is also a viable option to quantify errors as compared to classification (Muenzing *et al.*, 2012). Moreover, the authors claim an accuracy of 93.4% when they tested the algorithm to classify registrations incorrect, poor or wrong classes. Recently, Eppenhof et al. (Eppenhof and Pluim, 2018) proposed a sliding window 3D-CNN based approach to estimate registrations errors on lung CT scans. They have used synthetically deformed pair of a CT scan to train the network on 33x33x33 patches. They have evaluated

their approach on two standards: against a known, synthetic deformation field, and against manually annotated landmark points. They have reported a root-mean-square deviation of 0.51 mm from gold standard errors and 0.66 mm from ground truth landmark errors.

These previous supervised methods have been developed using hand-engineered features. However, it is challenging to utilize the full potential of a large amount of data by standard feature engineering-based methods, as it is more difficult to construct useful features which scale across imaging modality, registration algorithm, the magnitude of anatomical change, and body site. Recently, deep learning based methods have become more popular, as they can take advantage of the available data. Significant improvements have been reported in computer vision related problems by using convolutional neural networks (CNN) (Russakovsky *et al.*, 2015), which are well suited for medical image processing. Eppenhof *et al.* have employed convolutional neural networks to learn features predictive of registration accuracy (Eppenhof and Pluim, 2018). They used 3D-CNN and demonstrated model performance in the same deformation setting (breathing motion), achieving an impressive 0.66 mm predictive accuracy.

In this study, we use 3D convolutional neural networks to learn features to predict registration accuracy from image patches. We have trained our algorithm on publicly available pulmonary CT scans with one type of deformation (breathing motion), and tested it on in-house measured pulmonary CT scans with different anatomical change (day to day change in anatomy). To design a 3D-CNN for this purpose, we have closely re-implemented Eppenhof's (Eppenhof and Pluim, 2018) 3D-CNN architecture as a reference standard and compared its performance with ours. We show comparisons that our architecture performs with significantly less number of trainable parameters, which enables our approach to predict registration accuracy in the out of sample setting of different anatomical change.

### 3.3. MATERIALS AND METHODS

Below we describe the two separate lung CT datasets that were used for training and evaluating our method.

**3.3.1. Training Dataset: DIRLAB 4DCT Dataset.** Training dataset was generated from the publicly available <http://www.dir-lab.com/DIRLAB> 4DCT dataset that has images from ten patients. Each 4DCT scan contains five 3DCT images that cover the respiratory cycle between inhalation and exhalation. Out of ten patients, five were free of any lung disease (Castillo *et al.*, 2009) and the remaining five were treated for thoracic abnormalities (Castillo *et al.*, 2010). Images have either 256x256 or 512x512 voxels in-plane and between 94 to 138 slices. In-plane voxels have cross-sectional area of 0.97x0.97 to 1.16x1.16  $mm^2$ . The thickness of all voxels is 2.5 mm. For each 4DCT image, only the extreme inhale and exhale pair were made available for registration. The images were provided with 300 expert annotated landmark points distributed in the lung parenchyma each as ground truth (GT) for evaluation.

**3.3.2. Testing Dataset: 4D Lung CT Dataset.** Testing data for this work was gathered from patients treated with locally advanced non-small cell lung cancer at Virginia Commonwealth University (Guy *et al.*, 2018a). 4DCT or breath-hold scans of 17 patients were acquired during their course of radiotherapy on an institutional review board approved protocol after informed consent. The 4DCT patient data is also publicly available in the cancer imaging archive as part of the <https://wiki.cancerimagingarchive.net/display/Public/4D-Lung4D-Lung> Dataset (Hugo *et al.*, 2017). Image pairs were extracted for each patient. The simulation CT was selected as the baseline image and a mid-treatment CT selected from multiple re-simulation CTs acquired weekly during treatment. Note that all patients in this dataset had atelectasis (partially collapsed lung) with observed large anatomical changes due to resolution or progression of the atelectasis. The mid-treatment image was the one showing typically the largest amount of anatomical change, which represents a very different type of deformation than that seen in the DIRLAB dataset.

All images have  $512 \times 512$  voxels in-plane, and the number of slices ranges from 98 to 155. In-plane voxels have resolutions between  $0.98 \times 0.98$  to  $1.37 \times 1.37 \text{ mm}^2$ . Slice thickness ranged from 2 to 3 mm. For landmark generation, isiMatch software developed by Imaging Sciences Institute of the University Medical Center Utrecht (Utrecht, Netherlands) (Murphy *et al.*, 2011a) was used, as described in Guy *et al.* (Guy *et al.*, 2018a). Furthermore, the landmark sets underwent a rigorous QA process and were verified by an experienced physician (Guy *et al.*, 2019). On average, there were 169 landmarks per patient, with a standard deviation of 31. Testing data differed from training data in the way that it had large anatomical deformations compared to training data as described above, and that landmarks were annotated by different experts. Therefore, testing data represented a different deformation condition than in the training data.

**3.3.3. Quality Assurance Algorithm.** This section describes the method to quantify registration error between two sets of 3D volume images which have been registered with an arbitrary registration algorithm  $R$ . We employ a supervised deep learning approach, building a model to predict registration quality (based on LDE) from a pair of registered images. For training data, we determine registration quality based on thresholded mean LDE within a sub-volume as the ground truth and sub-volumes (patches) extracted from the registered images as the inputs. A neural network is used to construct the model. For inference of registration quality, a new pair of registered images is input into the model, which extracts sub-volumes from the images over the entire region of interest, infers registration quality for each sub-volume, and then reports registration quality for each sub-volume. Use of a sub-volume-based rather than a full image-based model was selected to maximize re-use of the imaging data, which should reduce the need for unique patients in the model training. This approach has been employed elsewhere, for example, in dictionary-based image reconstruction and other deep learning approaches.

For proof of principle, we assume pairwise registration between two images of the same modality, with a single transformation which deforms a moving image to match a fixed image. The fixed image is defined as  $I_f(\vec{x})$  and the moving image as  $I_m(\vec{h}(\vec{x}))$ , where the transform  $\vec{h}(\vec{x})$  maps 3D points  $\vec{x}$  in the fixed image domain to the point  $\vec{y}$  moving image domain,  $\vec{y} = \vec{h}(\vec{x})$ .  $I_f(\vec{x})$  represents the *fixed* image that does not go through any transformation during registration, while  $I_m(\vec{h}(\vec{x}))$  represents the *moving* image which goes through transformation according to a DIR algorithm  $R$  to align with  $I_f(\vec{x})$ . The details of the registration algorithm  $R$  are described in section 3.3.4.1.

To calculate LDE between  $I_f(\vec{x})$  and  $I_m(\vec{h}(\vec{x}))$  after registration, we measure physical distance (in mm) between the set of  $k$  landmark positions identified by an expert in the moving image and the position of the corresponding landmark after registration by  $R$ , such that,

$$LDE = \|y_k - y'_k\| \quad (3.1)$$

where  $y'_k$  is the expert-identified position of landmark  $k$  in the moving image and  $y_k$  is the registration-mapped position of the landmark in the moving image.

To quantify registration error between two images, one can calculate the probability of error in each voxel of the registered volume image (regression problem). For quality assessment, classification of a patch into *acceptable* and *needs review* meets our needs, so we treat the problem as a classification problem and classify sub-volumes inside the full image into good (acceptable) or poor (needs review) quality classes. Based on preliminary experiments, we have chosen a sub-volume of size 32x32x32 inside the lung region of the CT scan to be classified based on its mean LDE value. Finally, we compute the overall registration quality of a patient's CT scan, registration error index (REI), by the following formula,

$$REI = \frac{N_{good}}{N_{total}} \quad (3.2)$$

where  $N_{total}$  is the total number of sub-volumes and  $N_{good}$  is the number of acceptable registered sub-volumes in the registered image.

**3.3.3.1. Image pre-processing.** One of our objectives was to develop an algorithm invariant to image intensities and resolution. For example, in clinical cases, images may come from different scanners and different settings, which may affect the performance of a computer algorithm. To satisfy this goal, we have rescaled the image intensity values between 0 to 1 based on the input image’s real data type (e.g., 8-bit, 16-bit or 32-bit) (Lowekamp *et al.*, 2013). Moreover, different images might have a different spatial resolution (e.g., voxel size) in all x,y,z directions. Standard practice is to resample the images to  $1 \times 1 \times 1 \text{ mm}^3$  voxel resolution. However, we chose not to normalize the spatial resolution as different resolution images provide us a natural way for incorporating variations (augmentations) during neural network training.

The images were normalized according to the formula (Lowekamp *et al.*, 2013):

$$I_{out(x,y,z)} = (I_{in(x,y,x)} - I_{in,min}) \frac{(I_{out,max} - I_{out,min})}{(I_{in,max} - I_{in,min})} + I_{out,min} \quad (3.3)$$

**3.3.3.2. Training dataset construction.** One method of generating training and testing samples is by using a sliding window approach, that propagates through the CT volume (Eppenhof and Pluim, 2018) and generates sub-volumes. In this study, we have chosen random sampling of sub-volumes from the registered images to generate samples. By employing random sampling, we were able to generate a reasonable amount of class-balanced training samples from each image pair, which is important for CNN training. We have chosen sub-volumes from 100 random locations inside the lung region for each image pair. Sampling was controlled to ensure that enough landmarks were found within a region to generate a reasonable estimate of mean LDE. A minimum of 5 landmarks were required

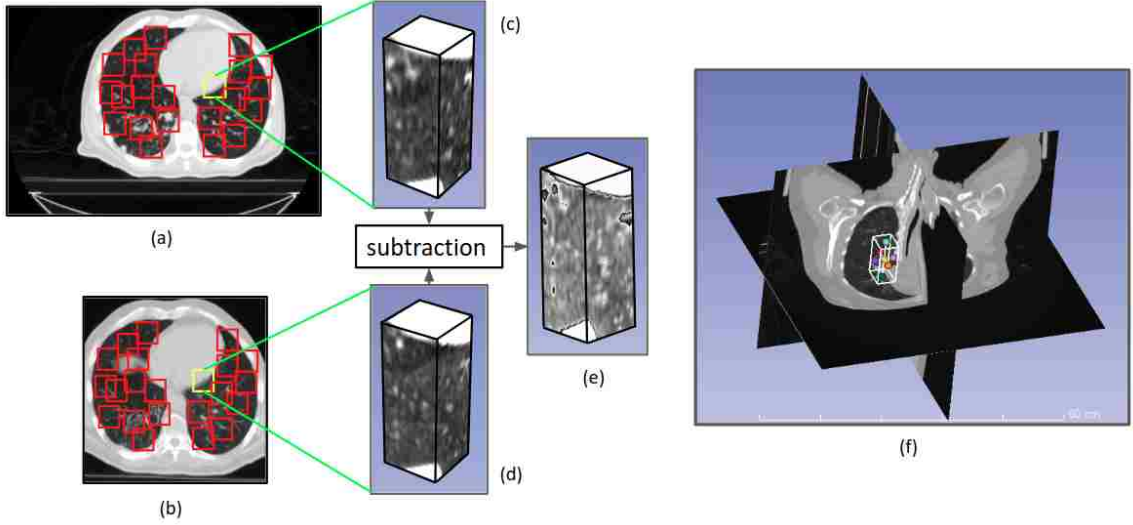


Figure 3.2. Training sample preparation. (a) a slice from the fixed image  $I_f(\vec{x})$ , (b) corresponding slice from the registered moving image  $I_m(\vec{h}(\vec{x}))$ , (c) volume view of a fixed patch  $I_{f,j}(\vec{x})$ , (d) volume view of corresponding patch the moving image  $I_{m,j}(\vec{h}(\vec{x}))$  from same location as  $I_{f,j}(\vec{x})$ , (e) subtracted view of (c) and (d). Neural network input is a composite image of (c), (d), and (e), (f) a sub volume inside the full image.

in each sub-volume, or a new random sample was drawn.  $I_{f,j}(\vec{x})$  and  $I_{m,j}(\vec{h}(\vec{x}))$  are the  $j$ -th sub-volumes from  $I_f(\vec{x})$  and  $I_m(\vec{h}(\vec{x}))$ , respectively. To construct the input samples for the neural network, we formed a composite image of three sub-volumes:  $I_{f,j}(\vec{x})$ ,  $I_{m,j}(\vec{h}(\vec{x}))$  and absolute difference of these two images,  $|I_{f,j}(\vec{x}) - I_{m,j}(\vec{h}(\vec{x}))|$ . Therefore, each input sample was a four-dimensional array of shape  $32 \times 32 \times 32 \times 3$ . Figure. 3.2 shows the process of creating training and testing samples.

**3.3.3.3. Neural network architecture.** Some of the challenging aspects of selecting a 3D-CNN are high computation cost requirement, size of training data required for generalization, and lack of pretrained networks for domain-specific problems in medical imaging. Different CNN architecture designs were investigated by various research groups to address medical imaging problems (Dou *et al.*, 2016, 2017; Eppenhof and Pluim, 2018; Ghafoorian *et al.*, 2017; Kamnitsas *et al.*, 2017; Sokooti *et al.*, 2017). A well-designed architecture should be computationally efficient, without overly compromising on the ac-

curacy. In this study, we have investigated two CNN architectures, and have compared the performance between them. One architecture was adapted and re-implemented from Eppenhof's (Eppenhof and Plum, 2018) proposed model (model B in Figure. 3.3), and we treat this model as our reference model. This network was slightly modified to work as a classification model, whereas the original implementation was for the regression model, and it has around 1.3M trainable parameters. Next, we have experimented with a six-layer neural network (model A in Figure. 3.3) that has around 308K trainable parameters. Input for the 3D-CNN was a 4D image (3 channel image of 3D sub-volumes) where the first channel came from the fixed image, the second channel from the transformed moving image, and the third channel from the mathematical subtraction between the above two images, as described above. For convolution, 3x3x3 kernels were used. After every convolutional layer, feature maps were down-sampled by 2x2x2 max-pooling layers, except for the last convolutional layer. We have also extracted additional feature maps by four consecutive 2x2x2 average-pooling of the input layer and concatenated it to the same sized feature maps from the convolutional layers. This additional feature of pooling was aimed to improve the performance, providing a better means of information flow. We have extracted bottleneck features from the last convolutional layer by global max-pooling operation (96 feature maps). Finally, a fully-connected layer of 32 neurons was used before the classification layer.

**3.3.3.4. Training method.** The neural network was trained using the Adam (Kingma and Ba, 2014) optimizer, and binary cross-entropy was used as the loss function. The network was trained for 50 epochs with the batch size of 32 samples. Different image augmentations, such as random rotation, random translation, and random flip were used to increase the generalization ability of the neural network. Batch normalization (Ioffe and Szegedy, 2015) was also used to improve training performance. Dropout (Srivastava



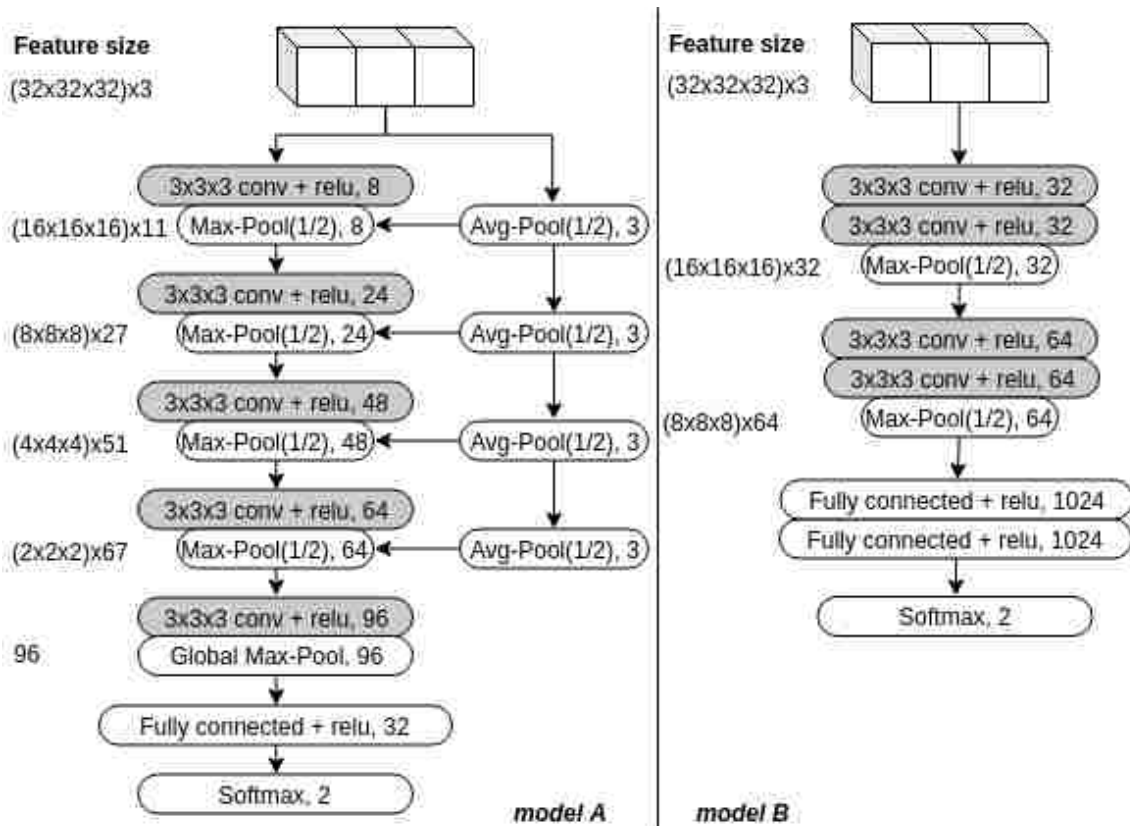


Figure 3.3. Neural network architectures. model A - from this work, model B - reference (Eppenhof and Plum, 2018)

*et al.*, 2014b) and L2 regularizer was applied to prevent overfitting. The hyperparameters, such as learning rate, number of convolution layers, number of filters were chosen from experimental results.

**3.3.4. Experiments.** This section describes the process of experiments and parameters used.

**3.3.4.1. Registration.** Using the SimpleElastix toolkit (Marstal *et al.*, 2016), b-spline based DIR was performed to register the moving to the fixed image, for each image pair. To generate a large number of registrations with varying quality, the hyperparameters of the registration were also randomly sampled for each registration. Sampled hyperparameters included: the similarity metric [*SSD, MattesMutualInformation*], the weight between the similarity term and a regularization term [ $10^0, 10^6$ ], final b-spline grid spacing

[2, 20] mm, and number of random samples for stochastic gradient descent based optimization [2000, 10000]. All registrations were performed using the similarity cost plus the regularization cost in a multiresolution framework (four levels of 8, 4, 2, 1x downsampling) where both b-spline grid size and image resolution were downsampled, with automatic parameter estimation, and with 256 iterations per resolution level. Transform bending energy was used as the regularization term.

This random sampling of registration hyperparameters and image region can be used to generate a substantial number of registrations with only moderate correlation as judged by the wide variation of registration quality as measured by registration accuracy (see Figure. 3.5). We have created 100 registration pairs out of the 10 patient images available for the training set and 33 pairs out of the 17 patients available for the test set.

**3.3.4.2. Landmarks.** By registering and applying the resulting transformation to the landmark in one image, the expected location of the landmark in the other image is generated. This expected location can be compared to the known location, and the distance calculated as an estimate of registration error in the region surrounding the landmark, known as the *landmark distance error*. In this dataset, the registration (dis)accuracy for a given registration was represented by the mean landmark distance error over all landmarks contained within the sampled fixed image region.

**3.3.4.3. Train and test set generation.** Training set Rather than having hundreds of millions of image pairs available, we made samples by extracting sub-volume patches from the ten image pairs available from the DIRLAB 4DCT dataset. Due to the similarity of lung parenchyma across patients at the patch level, and with the ability to vary registration quality by altering the registration hyperparameters, we are able to generate a large number of samples for training. First, each sample was generated by randomly selecting an image pair from one of the ten patients. Next, a 32x32x32 voxel region was randomly selected from the end of the inhalation image, and set as the *fixed* image. As mentioned previously, this region was tested to ensure at least five landmarks were present for the stability of

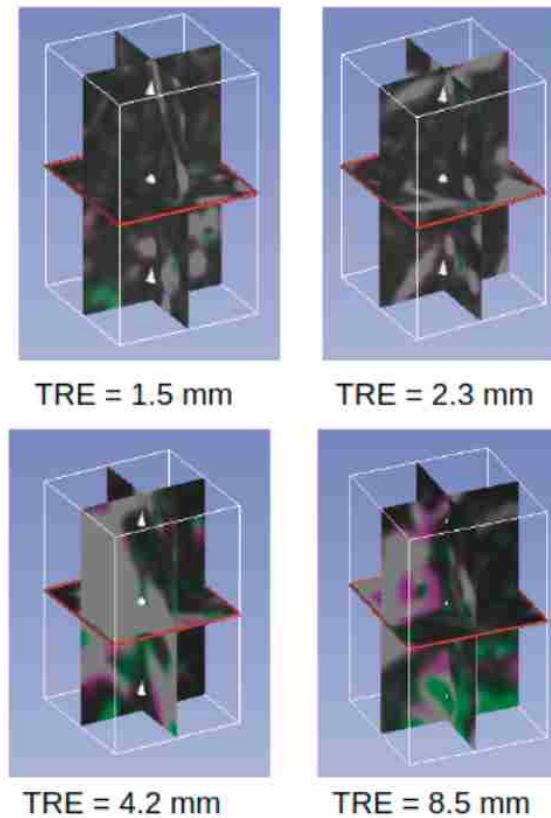


Figure 3.4. Example of training sub-volume patches

LDE estimation. The same region from the end of the exhalation image was selected as the *moving* image, and both images were saved to disk. For the current dataset, 100 sample sub-volume pairs were generated per patient. Therefore, we have generated a total of 10,000 patches for training.

Testing set Testing set was generated in the same way as the training set, but using VCU Lung CT data. Out of 33 registrations, we have generated 3,300 patches for testing.

To set a threshold for classifying registered sub-volumes as good, or poor quality, we can use the histogram of registration accuracy in Figure. 3.5 as well as estimation of acceptability for clinical use. Registrations with mean landmark distance error below 3.5mm were assigned as 'good' registrations. This threshold is similar to voxel size, which is a reasonable goal for registration accuracy for clinical use and is supported by the histogram

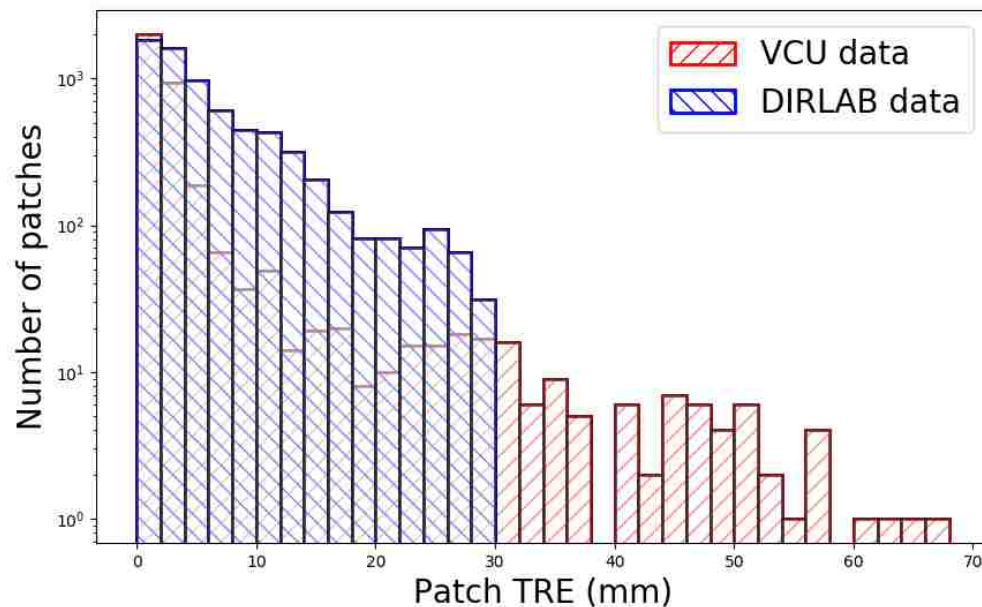


Figure 3.5. Histogram of registration accuracy (mm), DIRLAB and VCU dataset.

analysis. Selection of a different threshold would require re-training of the model, but not re-registration or re-sampling of the data. Figure. 3.4 illustrates four example patches at varying mean LDE.

**3.3.4.4. Implementation.** We have used keras (Chollet *et al.*, 2015) deep learning library for neural network construction and training. The algorithm was tested on both CPU and GPU based machines. It took about 72 hours to train the network for 50 epochs using an Intel 8-core i7 processor. When an NVIDIA Titan X GPU was used along with the CPU, training time was reduced to 48 hours. No significant speedup was observed while using a GPU, as 3D image processing was heavily carried out on CPUs, and thus it became a limiting factor for the GPU to get data from CPU.

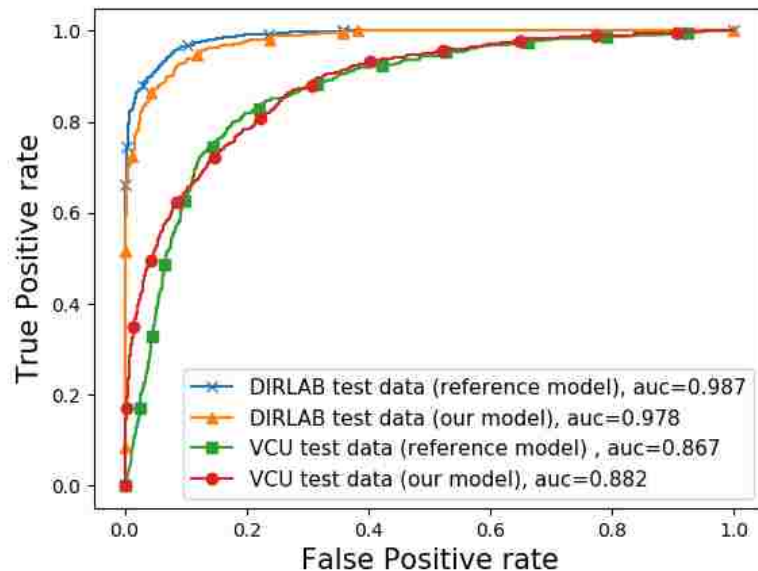


Figure 3.6. AUC-ROC curve for model A and B on DIRLAB and VCU test data

### 3.4. RESULTS

We have measured the performance of our method in terms of AUC (Area Under The Curve)-ROC (Receiver Operating Characteristics) curve. AUC-ROC is one of the standard metrics used for evaluating the performance of classification models. The curve is plotted having true-positive rate against the false-positive rate. AUC-ROC value can range from 0 to 1. The higher the value, the better the model is at distinguishing between classes. We have evaluated our model in two stages. At first, we have trained the models using 70% of DIRLAB data and tested it on remaining 30% of data. Both model A and model B achieved about 0.98 AUC-ROC. Next, we have tested the same models on VCU data and recorded a performance of 0.882 for our model (model A) and 0.867 for the reference model (model B). Therefore, our model outperforms the reference model by 1.02% AUC-ROC with 4.3 times fewer parameters. This improvement mostly comes from the carefully designed CNN architecture, realistic data augmentation, and overfit prevention strategies. Figure. 3.6 shows the AUC-ROC plot for all test cases.

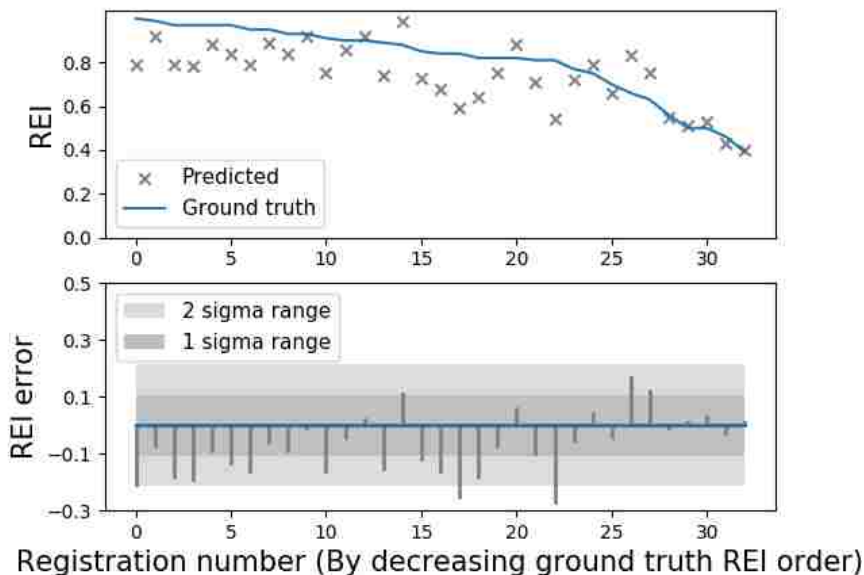


Figure 3.7. Ground truth REI vs. Estimated REI on VCU dataset.

Table 3.1. Effect of data augmentation on VCU test set

<b>Data Augmentation</b>	<b>AUC-ROC</b>	<b>CPU time(sec)</b>
No augmentation	0.865	60±10
Train	0.872	60±10
Train + Test(5X)	0.882	180±20

Effect of data augmentation is shown in Table. 3.1. We have experimented with both train and test time augmentations. Data augmentation improved the performance close to 2% compared to without augmentation. During testing, predicting a patch using multiple augmentations similar to train (i.e., random rotation, translation, flip) provided a substantial performance gain. However, by applying test time augmentation five times, computation time increased nearly three times.

Additionally, we have evaluated our method using REI (Registration Error Index). Figure. 3.8 shows both quantitative and qualitative comparison between ground truth and estimated classes. We have presented three examples of patient cases where registration

error was respectively low, high, and estimated REI differs most from ground truth REI. REI values can range from 0 to 1, where 1 represents a perfectly acceptable registration, and 0 represents an unacceptable registration. We have only classified 100 patches within the lung volume to calculate REI. This approach made our method fast yet reliable for registration quality quantification. From Figure. 3.8 (I) and (II), a good agreement between ground truth(GT) and estimated REI was observed. For Figure. 3.8 (I), GT and estimated REI was 0.90 and 0.92 respectively, and for (II) it was 0.39 and 0.40 respectively. However, for (III), we observed a relatively high gap between GT and estimated REI (0.81 and 0.53).

Finally, we plotted GT REIs against the estimated REIs for all the registrations in VCU dataset. Top chart of Figure. 3.7 shows the actual values of GT and estimated REIs, where the bottom chart shows the relative error between them. GT REI values for VCU dataset ranged from 0.99 to 0.39, and estimated values ranged from 0.99 to 0.37. The combined standard uncertainty (i.e., measured standard deviation) of the estimated REI lies within  $(\text{True REI} \pm 0.11)$  (i.e.,  $\pm 11\%$  from GT REI value) with a level of confidence of approximately 68%. Moreover, estimated REI values were higher than GT REI values for most of the registrations (73% of the registrations), which implies that our model has less probability to predict a poor registration as good than in reverse. This observation is essential, as in clinical practice, it lowers the risk of clearing a poor registration for further use without the review of an expert.

### 3.5. DISCUSSION

In this section, we have presented a method that estimates registration error from 3D volume images. This is a supervised machine learning algorithm which learns from expert annotated images. Specifically, we have utilized annotated landmark points to calculate mean LDE for different sub-volumes(patches) inside the lung region. Then the LDE measures were used as ground-truth for training a 3D-CNN model which classifies

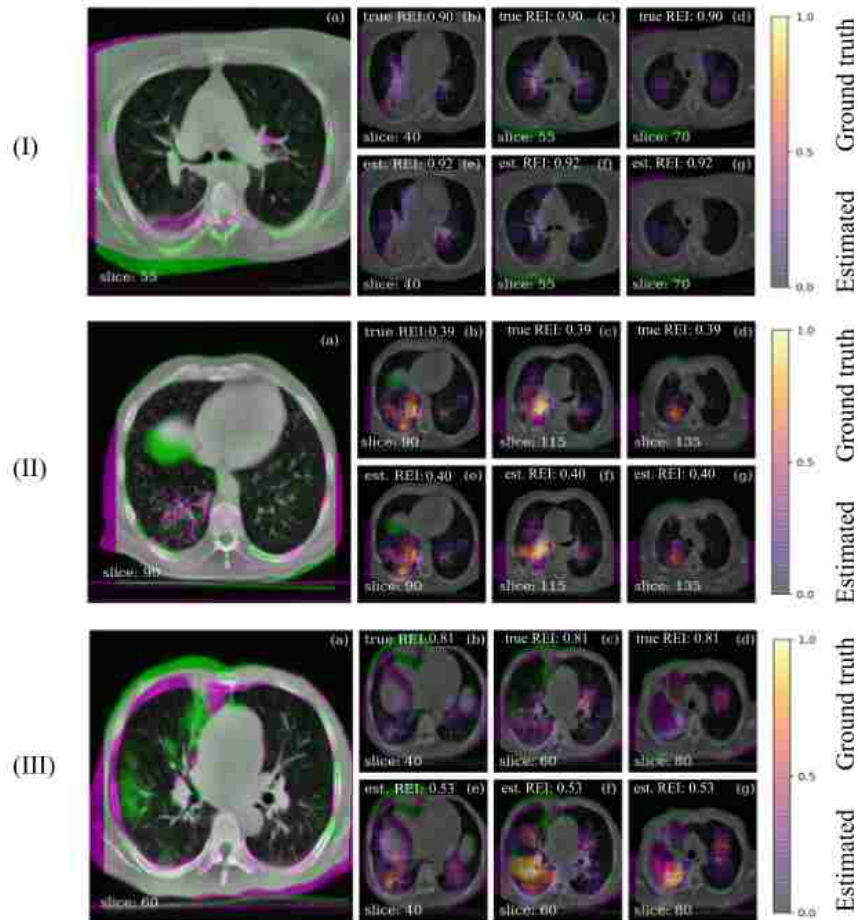


Figure 3.8. REI estimation. (I) An example registration where error was low. First row represents 3 slices of same scan at different depths with ground truth error heat-map. Second row represents the same 3 slices with predicted error heat-maps. The heat-maps show the probability of registration error in respective slices. (II) An example where error was high, and (III) An example where estimated REI differs most from ground truth REI

a patch as good (acceptable) or poor (needs review) class. Our proposed 3D-CNN model showed promising results on out-of-train dataset images while training on a relatively small number (i.e., ten patient scans from DIRLAB dataset) of image pairs.

Training on DIRLAB dataset and testing on VCU dataset provided us an opportunity to evaluate the strengths and weaknesses of our proposed algorithm in a more realistic manner. As expected, we have observed a drop in performance in the VCU test set compared to DIRLAB test set (AUC-ROC 0.98 to 0.88). There are several reasons that might explain



this result. First, we trained the model using only inhale and exhale pairs of CT scans (DIRLAB data), while the VCU data consists of images of different treatment days and contained larger deformations between scans compared to DIRLAB. Second, the landmark annotation schema between the two datasets differs (300 landmarks per scan in DIRLAB compared to 169 on average in VCU, landmarked by different physician experts using different software tools for annotation). Third, the registration algorithms used between DIRLAB and VCU dataset were different. For DIRLAB, a conventional b-spline algorithm with standard intensity-based cost function was employed using SimpleElastix (Marstal *et al.*, 2016), while for VCU, an in-house developed algorithm 'CALIPER' (Guy *et al.*, 2018a) was applied. VCU variations were therefore highly underrepresented at training, and we observed the drop in performance partly because of this reason. However, by comparing the testing methods, we can conclude that even if the unseen testing data came from very different environments and patient conditions, our method generalized well in challenging scenarios.

We carried out several experiments to find a computationally efficient neural network architecture. An input patch size of  $32 \times 32 \times 32$  worked best in our experiments. A smaller patch size (i.e.,  $16 \times 16 \times 16$ ) contains less context information and also achieved lower accuracy during training. Moreover, it is difficult to train a deep neural network with small image size, as max-pooling operation quickly reduce the feature map in CNNs. On the other hand, by using larger patches (i.e.,  $64 \times 64 \times 64$ ), we found it not favorable to localize registration errors on lung scans as this is a relatively large area to calculate mean LDE. As experts may need to look for registration errors on relatively small radiation treated areas, we have found it optimal to use  $32 \times 32 \times 32$  patches for training and testing. Furthermore, we were able to reduce the number of trainable parameters compared to a competing method (Eppenhof and Pluim, 2018) by designing a modified CNN on  $32 \times 32 \times 32$  input size.

We have also experimented with different input image processing techniques. We used a 3-channel input for the model training where the first channel came from the fixed image, second from the moving image and third from the absolute difference of the fixed and moving image. Training with only the first two channels performed little lower (accuracy: 0.916) compared to all 3-channel input (accuracy: 0.924). The third channel can also be viewed as an attention map for the CNN, which provides meaningful information related to registration error with very little additional computation cost. We have also experimented with adding deformation vector field (DVF) as a channel. However, it did not contribute to the validation accuracy significantly. To make the pipeline simple, we did not incorporate any DVF information in training as the CNN was strong enough to extract features from the raw inputs.

There are a few works reported in the literature to address registration QA based on supervised approaches (Eppenhof and Pluim, 2018; Muenzing *et al.*, 2012, 2014; Neylon *et al.*, 2017; Sokooti *et al.*, 2016). Most of the supervised methods either use hand engineered features (Muenzing *et al.*, 2012, 2014; Sokooti *et al.*, 2016) or lack studies on real clinical data (Neylon *et al.*, 2017). A more recent work by Eppenhof *et al.* (Eppenhof and Pluim, 2018) is a closely relevant method to our approach that used a 3D-CNN to predict registration error without the need for any engineered feature calculation. Similar to our method, they used a patch size of 33x33x33 for training and testing. However, their method did not require any expert annotation for training, which was a favorable advantage. They created the registration image pairs from a single image by synthetically deforming it into two images. This was a potential drawback of their algorithm as synthetic transformations may not represent the variations that occur in clinical cases. To address this issue, we created our training dataset from clinical CT scan pairs that would be registered. Image pairs generated from a matching setting (e.g., inhale-exhale pair, synthetic pair from a single image) may provide high accuracy while this performance may not hold for other settings (e.g., image pair before and after radiation treatment).

Furthermore, our study provides an in-depth analysis of inter-setting variability during testing. Using a smaller network with fewer parameters, but the similar performance was selected to prevent overfitting and enable our network to generalize to out-of-sample data. This was demonstrated, as described above, by good performance on a very different dataset under different registration conditions. Moreover, to account for inference time, we preferred a classification approach over the regression approach used by Eppenhof *et al.*, as it requires less time while providing similar information to guide clinical review of unacceptable registration results. In our test cases, we estimated registration quality by using only 100 sub-volumes, which took  $60 \pm 10$  seconds for every registration pair. Therefore, our method provides an improved pipeline for registration QA by mimicking real clinical scenarios in a practically applicable manner.

One of the major limitations of this study was that it needed expert annotation for training. Expert annotation is costly, and opinion can vary person to person, although there are methods to improve throughput and reduce annotation time per case (Guy *et al.*, 2018b; Murphy *et al.*, 2011b). If there were not enough examples, an algorithm could incorporate some unwanted biases. Even though our algorithm learned reasonably well from a limited number of patient images, we can observe the effect of bias from Figure. 3.8 (III). Qualitative interpretation of this patient case suggests that there are mismatches in different areas inside the lung region. Our model estimated a high error even though the mismatched portion might be relevant/irrelevant to an expert. In practical scenarios, these discrepancies can be quickly settled with expert knowledge as this type of cases lie in the gray area between acceptable and needs review classes.

### 3.6. REMARKS

Our proposed method is general, fast, scanner type independent and can be extended to various other image registration problems. Our experiments suggest that it is not required to have thousands or millions of images available for training a deep neural network

successfully for this task. Our model was trained on only a handful of annotated images and performed reliably in a different test setting. Moreover, our method can be applied to images of different organs or body parts as well (i.e., brain, abdomen, nervous system), although this would likely require additional site-specific training data. A similar approach applies to MRI, PET, SPECT images and also for multi-modality images, although this would similarly require additional training data.

## 4. AUTOMATED RADIOISOTOPE DETECTION USING HYBRID NEURAL NETWORKS

### 4.1. BACKGROUND AND SECTION OUTLINE

Automatic radioisotope identification in real-time is important in a wide variety of applications including prevention of nuclear terrorism (Connor *et al.*, 2016; Mucci, 2016; Sanada and Torii, 2015), monitoring of environmental contamination (Hosoda *et al.*, 2016; Rangaswamvi *et al.*, 2018; Uekusa *et al.*, 2015) or medical data analysis (Sahiner, 2017). Different types of radioactive materials are screened during a radiological search operation, such as, Special Nuclear Materials (SNM) (e.g., high enriched uranium, weapons-grade plutonium), medical isotopes (e.g., patients recently treated with radiation/nuclear medicine may contain Iodine-131 or Technetium-99), industrial isotopes (e.g., Cobalt-60, Selenium-75 is used for gamma radiography and non-destructive testing) or Naturally Occurring Radioactive Materials (NORM) (e.g., Carbon-14, Tritium). The search teams rely on automatic algorithms that provide feedback about the presence of any radiation anomaly. However, accurate identification of radioisotope is difficult in practice and complicated by a large variation in background conditions, poor resolution of radiation detectors, calibration shift, overlapping photo-peak energies, source strength, shielding conditions and among others. Therefore, in addition to design better hardware, a sophisticated computer algorithm is also a needed tool for automatic detection of radioactive materials.

The problem considered in this study assumes that the radioisotope detection system is travelling in a search vehicle. Unlike stationary type systems, mobile systems need to be lightweight, small-sized and consume as less power as possible. These create limiting factors in choosing the hardware. It is preferable that the detector would have high efficiency and low cost. Though semiconductor detectors (i.e., HPGe) provide superior energy resolution, crystal scintillator detectors are preferred where efficiency and portability are essential. For

example, thallium-doped sodium iodide detector, NaI(Tl), is a common choice as a portable detector because of its reasonable cost. However, NaI detectors have relatively weak energy resolution and temperature dependence, which makes it weak to resolve many of the photopeaks in a spectrum (Stinnett *et al.*, 2017). Nonetheless, sophisticated algorithms can still provide the possibilities to improve the detection performance (Blackadar, 2005) from low-resolution detector data.

Recently published articles show impressive results in automatic isotope identification methods (Bobin *et al.*, 2016; Elmaghraby *et al.*, 2019; He *et al.*, 2018; Kamuda *et al.*, 2017; Zhang *et al.*, 2019) using artificial neural networks. Most of the studies reported their experiments in a simplified laboratory setting, which led to high accuracy results. Furthermore, the performance of these algorithms can be limiting where low computation cost is desired. A little amount of reported work discussing mobile radiation detection systems and their different testing conditions make it unsuitable to judge a particular state-of-the-art method in this study area (Stinnett *et al.*, 2017). Therefore, more studies are needed to improve detection accuracy assuming real-life environment and real-time scanning.

In this work, we report a supervised method that learns from gamma spectra using Hybrid Neural Network (HNN) models. Specifically, we have developed two hybrid neural networks, HNN1 and HNN2, where former is the combination of FCNN and CNN and later is FCNN and RNN. We have compared the HNN models with three other (FCNN, RNN, and Gradient boosted decision trees (GBDT)) ML models to evaluate our method. Moreover, we compared three different feature calculation methods (i.e., automatic, semi-automatic, manual) that are relevant to gamma spectroscopy. Finally, the robustness and stability of the techniques were evaluated at different testing conditions.

## 4.2. RELATED WORKS

Automatic radioisotope detection algorithms can be roughly categorized into two types, condition-based and machine learning based approaches. In condition-based methods, an isotope can be identified by measuring the photo-peak energy information from the gamma spectrums. This does not require any training method, and usually, physics-based rules were applied for identification. On the other hand, in machine learning based approaches, gamma spectrums and their respective isotope type need to be known in advance in order to train a predictive model. Then the trained model can be used to estimate the probability of the presence of a radioisotope in an unseen environment.

Some of the condition based approaches (Burr and Hamada, 2009) include simple library comparison methods (Alamantios *et al.*, 2013; Batdorf *et al.*, 2009), region of interest (ROI) methods, template matching methods (Mitchell and Harding, 2014) and some others (Runkle *et al.*, 2006). In the library comparison method, the centroid energy of each photo-peak is considered, but not the area or branching ratio. A slight shift in detector response or calibration may result in inaccurate detections. In the ROI method, one or multiple ROIs were selected from each nuclide. One of the major drawbacks of this method is that it fails if different nuclides contain overlapping ROIs. In the template matching method, a few example spectrums of different radioisotopes were generated at varying strength and/or background conditions and then matched with a test spectrum. Condition-based approaches did not usually produce reliable results in real-time applications (Burr and Hamada, 2009; Sullivan *et al.*, 2007).

Machine learning approaches have been investigated in radioisotope identification for nearly three decades. In this approach, artificial neural networks (ANN), support vector machine (SVM), nearest neighbors, decision trees, bayesian, normalized cut are some of the successfully applied methods. However, ANNs, such as fully-connected neural networks

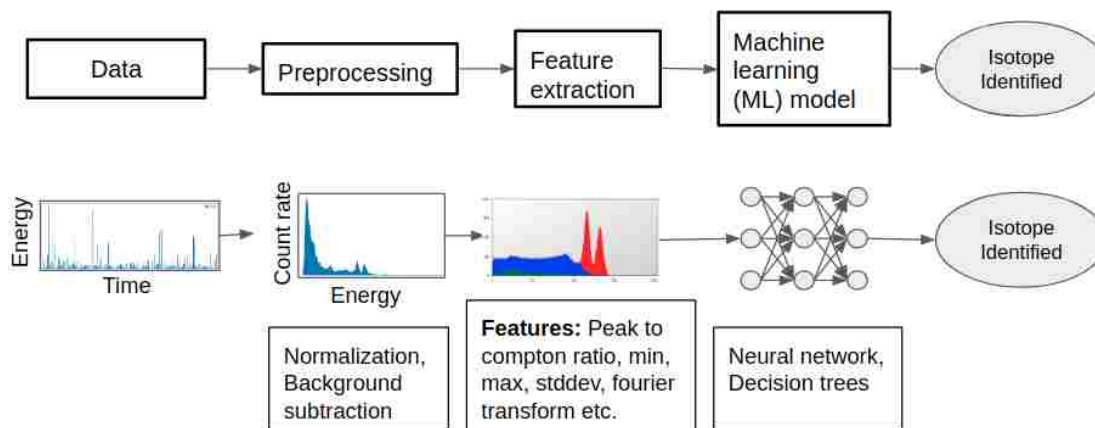


Figure 4.1. Block diagram of a conventional radioisotope detection algorithm

(FCNNs) are most commonly used in interpreting radiation data as they usually provide higher accuracy than other methods (Bobin *et al.*, 2016; Elmaghraby *et al.*, 2019; He *et al.*, 2018; Kamuda *et al.*, 2017; Zhang *et al.*, 2019).

A classical approach to train machine learning models using gamma-ray spectra is to use the numerical values from each channel of the spectrum directly (Bobin *et al.*, 2016; Kim *et al.*, 2019; Sahiner, 2017). Some researchers have also investigated feature engineering approaches like taking wavelet transform (Stinnett *et al.*, 2017) or discrete cosine transform (He *et al.*, 2018) of the spectra and feed the feature vectors to the machine learning models. However, feature calculation can be time-consuming during the development phase and also computationally costly at deployment, which makes some of the algorithms less attractive in real-time applications. Figure. 4.1 illustrates a classical approach of radioisotope detection algorithm.

One of the significant limitations of FCNN models is that they do not consider input data points (i.e., feature vectors) as sequentially related, whereas gamma spectrum photo-peaks possess relationship among them. Therefore, FCNNs can not exploit full information of gamma spectrums, and better methods need to be investigated.



Recent developments deep learning achieved significant performance gain in computer vision-related problems (Krizhevsky *et al.*, 2012; Russakovsky *et al.*, 2015), as well as speech (Graves *et al.*, 2013) and text (Hermann *et al.*, 2015b) recognition problems. Convolutional neural networks (CNNs) and recurrent neural networks (RNNs) are the most successful methods for unstructured data types (i.e., visual imagery, audio, text), where they can exploit the sequential information (i.e., temporal/spatial relationship) of the data. Unlike FCNNs, CNNs or RNNs usually need a large amount of training data to produce satisfactory results.

### 4.3. MATERIALS

**4.3.1. Dataset Description.** The gamma-ray detector data for this study was obtained from publicly available website *www.challenge.gov* (NNSA, 2019), an US government initiative for research and development purposes. Several US federal and non-federal agencies partnered together to generate the data (i.e., National Nuclear Security Administration, NASA, Los Alamos National Laboratory, Oak Ridge National Laboratory and Lawrence Berkeley National Laboratory). The dataset contains simulated radiological search data that would be typically generated by moving through the streets along a mid sized US city.

For the simulations, Monte Carlo N-Particle code (C.J. Werner, 2018) was employed to generate data files in a time-series manner. A standard NaI(Tl) detector with 7.5% resolution at 661 keV, traveling in a search vehicle, was assumed. Traveling speed of the detector can vary between 1 to 13.4 meters per second. Radioisotope type, strength, shielding conditions, and background conditions (the type of roadways/buildings/structures present) may differ from case to case as well. In total, 9700 data files with publicly known ground truth were provided in the dataset. A schematic diagram of the simulation setup is shown in Figure. 4.2.

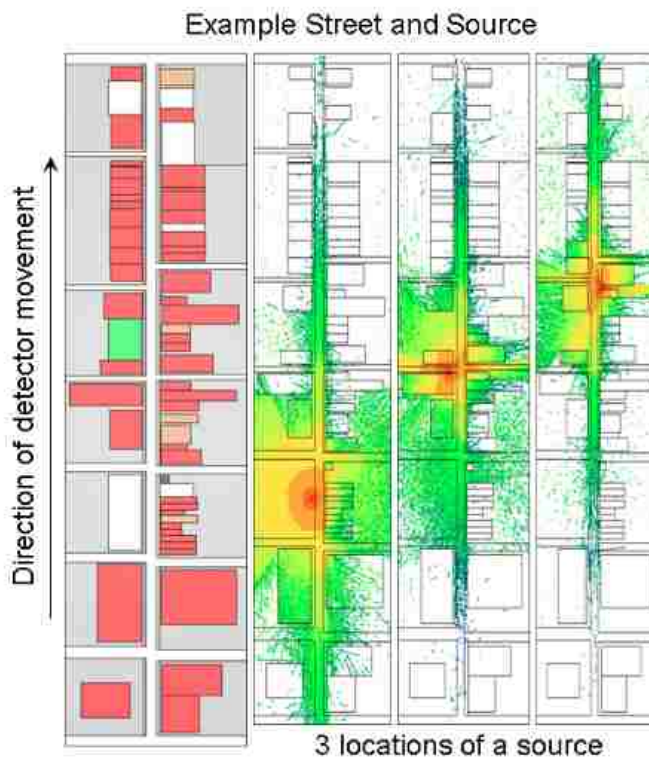


Figure 4.2. Schematic diagram of the MCNP simulation (NNSA, 2019)

**4.3.2. Radioisotope Types.** The dataset contains six different types of source combinations: Highly enriched uranium (HEU), Weapons-grade plutonium (WGPu), Iodine ( $^{131}\text{I}$ , a medical isotope), Cobalt ( $^{60}\text{Co}$ , an industrial isotope), Technetium ( $^{99m}\text{Tc}$ , a medical isotope) and a combination of  $^{99m}\text{Tc}$  and HEU. A normalized spectrum of these sources in both shielded and unshielded conditions are presented in Figure. 4.3.

The sources represent both harmless medical/industrial isotopes and threatening weapon producing isotopes. These radioisotopes contain one or more closely overlapping photo-peaks, and some of the crucial photo-peaks can appear at very low energies (<200 keV). Low energy peaks can be highly suppressed by shielding materials, making the detection task more challenging in real-time. Table 4.1 shows the number of time-series data files for each radioisotope.

Table 4.1. Number of radiological search data files in the simulation dataset

<b>ID</b>	<b>Source type</b>	<b>No. of samples</b>
0	Background	4900
1	HEU	800
2	WGpu	800
3	$^{131}\text{I}$	800
4	$^{60}\text{Co}$	800
5	$^{99m}\text{Tc}$	800
6	HEU + $^{99m}\text{Tc}$	800

#### 4.4. METHODS

First, gamma spectrums were created from time-series detector data. Next, three different methods were applied for feature extraction from the spectrums for machine learning model training. Then, five different machine learning models were tested for radioisotope classification performance. Below we describe these processes in relevant sections.

**4.4.1. Training Samples Generation.** Simulation data in this study was generated in a way that photon energies(keV) were recorded while the detector was in motion (i.e., time-series data). If a source was present in a case, it can be identified from a relatively small fraction of the time-series data. Therefore, small segments of the data were selected for creating the gamma spectrums.

Ideally, the most desirable segment size would be the minimum number of counts or time needed for a detector to identify the presence of a source. However, as this calculation depends on the source type, intensity, shielding condition and also the traveling speed of the detector, it is challenging to find the exact number of counts or time required for constructing the preferred sized segment or time-window.

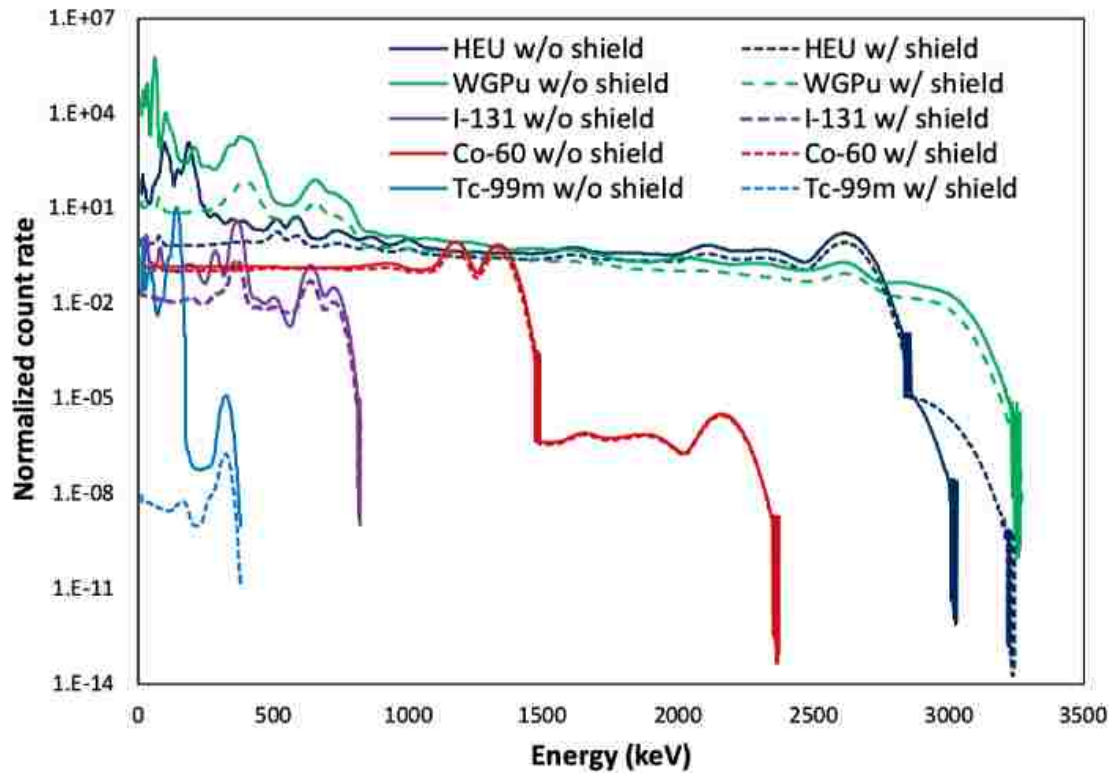


Figure 4.3. Simulated spectrum of the five radioisotopes (HEU, WGPu, I-131, Co-60, Tc-99m) using a NaI(Tl) detector. From the spectrums it is observed that, radioisotopes posses overlapping peak energies and many of the important peaks appear below 200 keV.

To approach this optimization problem, time-series segments of different acquisition times were collected. Time-windows were ranged from 0.2s - 20s counts. If a source was present, we have collected six segments from near source location; otherwise, six segments from random locations were gathered. Figure. 4.4 shows a schematic diagram of the training sample generation process. Approximately 60000 training spectrums were generated from 9700 time-series data available. About half of the spectrums did not contain any source. Rest half of the spectrums equally represented the six sources.

**4.4.2. Feature Selection.** Three different types of features were extracted from the gamma-spectrums for machine learning models input and for comparison. The features were chosen such that they represent automatic (i.e., SC), semi-automatic (i.e., DCT) and manual (i.e., PR) feature extraction methods.

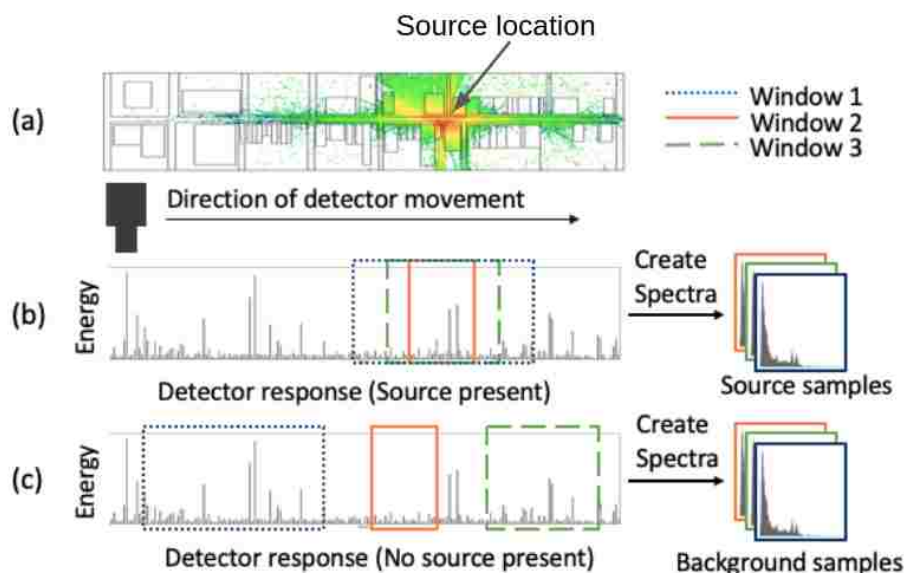


Figure 4.4. The process of training spectrums generation from time-series detector data

**4.4.2.1. Spectrum counts (SC).** SC features are the number of counts in each channel of the gamma spectra. This features can be interpreted as raw data as no explicit feature calculation was carried out on this method. For the experiments, 100 channel spectrums were used, and SC features were fed to machine learning models after normalization and scaling of the channel counts. We refer to this method 'automatic' as the features are essentially raw data without

**4.4.2.2. Discrete cosine transform (DCT).** DCT converts a signal into the frequency domain, and the transformation kernel is a cosine function as presented in Eq. (4.1). The advantage of DCT is that it can pack the energy of the spatial sequence into a few frequency coefficients. DCT is faster and presents high compaction of energy than Fourier transforms using only real numbers. DCT is successfully used in image/audio compression techniques, and it is also found to be useful in radioisotope detection algorithms (He *et al.*, 2018). In total, 128 feature vectors were extracted from 100-channel gamma spectrums using DCT. We refer to this method 'semi-automatic' as one still needs to find out which transform method may work well for gamma spectroscopy.

$$t(x, u) = C(u) \sqrt{\frac{2}{N}} \cos \frac{(2x + 1)u\pi}{2N} \quad C(u) = \begin{cases} \frac{1}{\sqrt{2}} & u = 0 \\ 1 & \text{other} \end{cases} \quad (4.1)$$

where  $u$  is an integer in the range of 0 to  $N - 1$

The spectrum can be considered as a  $N$  dimensional vector of  $y$ , then  $y(x)[x = 0, 1, 2, \dots, N - 1]$  are the discrete signals of the spectrum. The signal of the  $Y(u)$  spectrum can be expressed by Eq. (4.2).

$$Y(u) = C(u) \sqrt{\frac{2}{N}} \sum_{x=0}^{N-1} y(x) \cos \frac{(2x + 1)u\pi}{2N} \quad (u, x = 0, 1, 2 \dots N - 1) \quad (4.2)$$

**4.4.2.3. Peak ratios (PR).** In this method, 51 numerical features were calculated from the gamma-spectra based on the peak ratios. Bin counts of important energy values(keV) associated with the sources considered in this study (Table 4.2) were used to calculate peak-to-peak ratios (4.3) and peak-to-compton ratios (4.4). These ratios play an important role in radiation detection and measurement methods and provide useful information to distinguish between different radioactive sources (Knoll, 2010). Next, we have joined the 51 peak-ratio features with the 100-channel counts to create 151 feature vector for machine learning model training. We refer to this method 'manual' as the peak ratios are handcrafted features and depend on the radioisotope to be identified. Calculation of these features does not increase the computational burden significantly but still helps in extracting meaningful information from a gamma spectrum.

$$\text{Peak to peak ratio} = \frac{\text{counts}_p}{\text{counts}_{ref}} \quad (4.3)$$

$$\text{Peak to compton ratio} = \frac{\text{counts}_p}{\text{counts}_{comp}} \quad (4.4)$$

Table 4.2. Peak energies considered for PR feature calculation

ID	Source type	Ref. peaks (keV)	Other peaks (keV)
1	HEU	511, 585	17, 97, 187, 2615
2	WG Pu	381, 407	33, 61, 77, 101, 207, 653
3	$^{131}\text{I}$	365	31, 81, 185, 285, 637
4	$^{60}\text{Co}$	1173, 1333	-
5	$^{99m}\text{Tc}$	141	19, 45, 321
6	HEU + $^{99m}\text{Tc}$	141, 511, 585	17, 19, 45, 97, 187, 321

where,  $counts_p$  represents counts under an important photo peak P for source X,  $counts_{ref}$  represents counts under the reference photo peak for source X and  $counts_{comp}$  represents average counts of compton plateau for peak P of source X.

Reference peak was determined such that this peak should appear in both shielded and un-shielded conditions for a specific source.

**4.4.3. Classifiers.** Five different classifiers were tested in this study for comparison. Four of the classifiers were neural network based and the other one was decision tree based.

**4.4.3.1. Fully-connected neural network (FCNN).** FCNNs are classical type of neural networks and studied extensively in automated radioisotope identification algorithms (Elmaghraby *et al.*, 2019; Kamuda *et al.*, 2017; Olmos *et al.*, 1992; Wiedner *et al.*, 2017; Zhang *et al.*, 2019; Zhao *et al.*, 2017). Artificial neurons in FCNN are connected depth-wise (or layer-wise), creating abstract representation of input data. FCNNs do not assume any sequential relationship among the data points, as the neurons do not have a connection in the temporal direction.

In this study, we have re-implemented an FCNN model adapted from the work of He *et al.* (He *et al.*, 2018). The FCNN consists of one hidden layer with eight neurons, and we consider this as a baseline model.

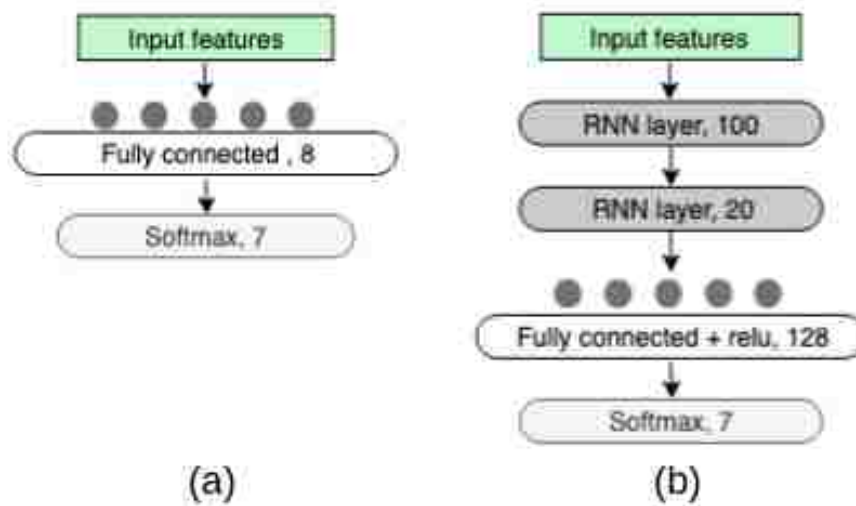


Figure 4.5. Schematic diagram of (a) FCNN and (b) RNN model

**4.4.3.2. Recurrent neural network (RNN).** Unlike FCNN, neurons in RNNs exhibit temporal connections, which makes them useful for sequence learning applications (Graves, 2013; Graves *et al.*, 2014; Hermann *et al.*, 2015a; Mikolov *et al.*, 2013; Sutskever *et al.*, 2014). We have implemented a RNN based on long short-term memory (LSTM) (Hochreiter and Schmidhuber, 1997) cells. LSTM provides a solution for vanishing gradient problem in RNNs. The RNN has 2-hidden layers before the classification layer. Hyperparameters, such as the number of hidden layers and LSTM cells were optimized experimentally. Figure. 4.5 shows schematic diagram of FCNN and RNN models.

**4.4.3.3. Hybrid neural network 1 (HNN1).** Recent developments in deep learning and convolutional neural networks (CNNs) achieved significant performance gain in computer vision-related problems (Russakovsky *et al.*, 2015). CNNs are known as automatic feature extractors from data. By principal, CNNs generate internal representations from the full sequence of data. CNNs performance is well studied in unstructured data cases (i.e., images, videos, speech); however there is a lack of studies about their capabilities in structured data cases (i.e., tabular data).



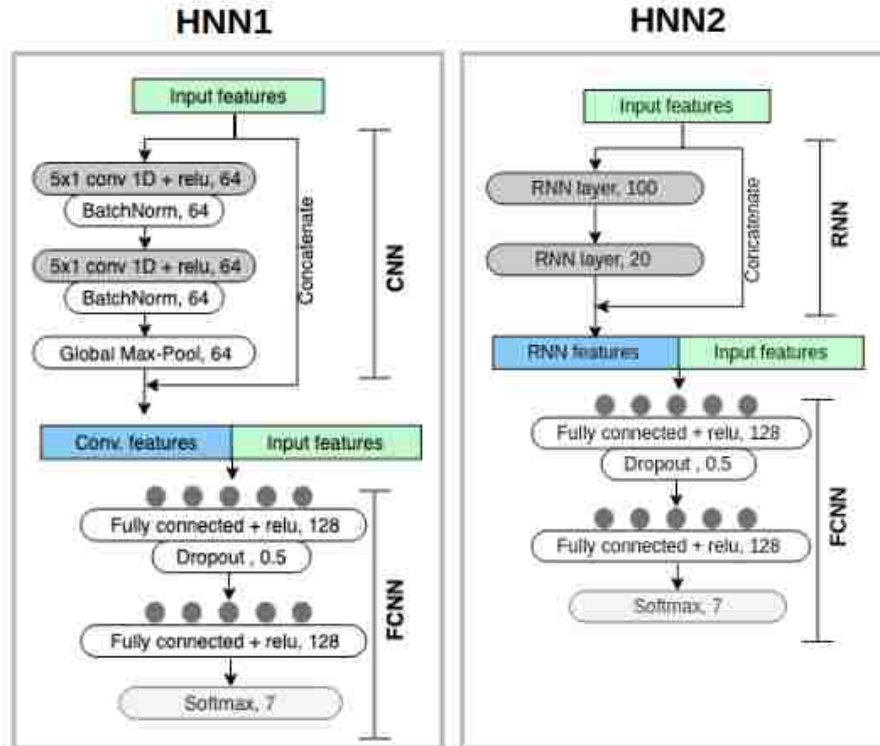


Figure 4.6. Schematic diagram of hybrid neural network architectures

Inspired by the working principle of CNNs, we have designed a hybrid neural network 1 (HNN1) architecture that consists of CNN and FCNN structures. First, 64 convolutional feature vectors were learned from the input features by a 2-layer CNN. Next, these 64 features were added to the original input features to produce a larger feature space. These feature vectors were then fed to a 2-layer FCNN to produce final predictions. The HNN1 was trained end-to-end, like a single model. Figure. 4.6 shows the schematic diagram of HNN1 architecture.

**4.4.3.4. Hybrid neural network 2 (HNN2).** HNN2 is mostly similar to HNN1, except it uses RNN layers instead of convolutional layers for feature extraction. A two-layer RNN with LSTM cells was employed on this architecture. RNN layers are responsible for extracting sequence information from the data. Figure. 4.6 shows the schematic diagram of HNN2 architecture.

Table 4.3. Parameter optimization for GBDT model

Parameter	Search range for optimization
Max. bin size	20-300
Number of leaves in each tree	1-20
Min. data in leaf	1-20
Max. depth of tree	1-20
Bagging fraction	0.2-1.0
Feature fraction	0.1-1.0

The neural network models were implemented using python language and Keras (Chollet *et al.*, 2015) deep learning library.

**4.4.3.5. Gradient boosted decision trees (GBDT).** Decision tree is an algorithm that finds ways to split a dataset based on different conditions. It is a tree-like model of decisions and consequences. The algorithm can usually contain if-else-then statements. Gradient boosted version of decision trees (GBDT) is an ensemble of several weak models like several basic decision trees.

Gamma spectra contains conditional relationship among its energy values. For example, an isotope might have multiple energy peaks which distinguishes them from others. Thus, GBDT useful in gamma spectroscopy (Barzilov and Novikov, 2015; Parkinson *et al.*, 2016) techniques. However, GBDT does not assume any sequential or temporal relationship, which is a weakness of this model. Figure. 4.7 shows a schematic diagram of GBDT model on a leaf wise growth method.

We have trained a GBDT model using the same input features as used in the neural network models. LightGBM (Wang *et al.*, 2017) library was employed for model training. LightGBM grows trees in a leaf-wise manner where best tree grows first. It chooses the leaf with max delta error to grow. We have employed mean squared error as an objective function. To limit overfitting, hyperparameters were tuned before the final experiments. Some important parameter search space is shown in Table 4.3.

Overall experimental setup is illustrated in Figure. 4.8

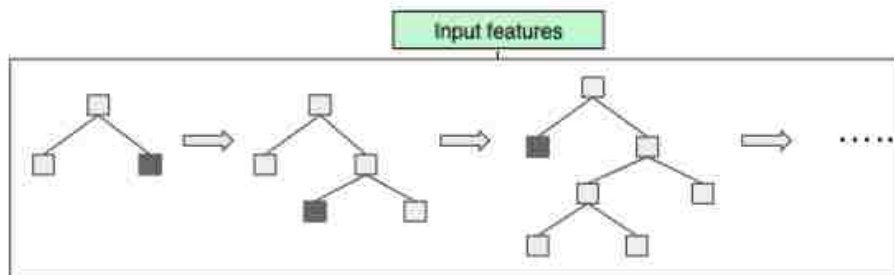


Figure 4.7. Schematic diagram of GBDT model in leaf-wise growth method

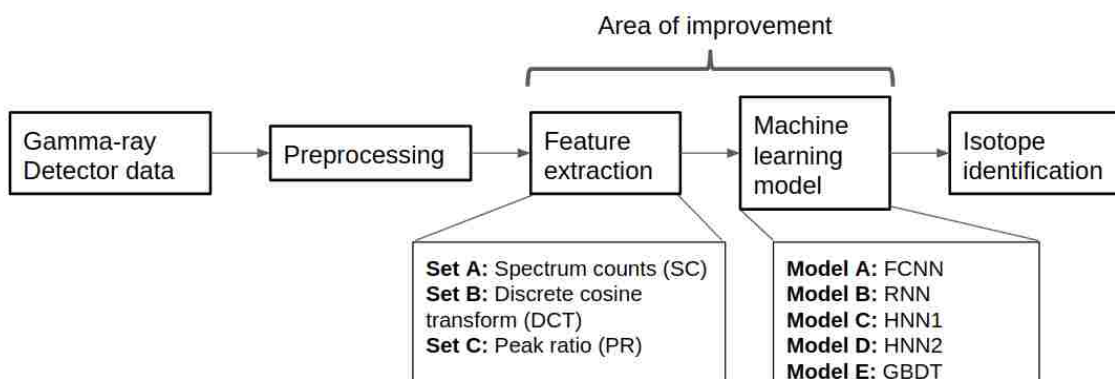


Figure 4.8. Schematic diagram of the overall experimental setup

**4.4.4. Training and Validation.** We have divided the dataset into training, validation, and testing set by 80%-10%-10% ratio. Random stratified split was applied to create the three sets. All the sets contain a balanced percentage of radioisotope samples. The model was trained until the validation accuracy did not improve for 50 training iterations. We had checkpointed the model weights when validation accuracy reached the maximum value. Finally, we have reported the model performance on test data by taking the average test F1 score among different training sessions.

The neural network was trained using categorical cross-entropy loss function. Adam(Kingma and Ba, 2014) optimizer was employed to minimize the training loss. Furthermore, batch normalization(Ioffe and Szegedy, 2015) was applied to train the model faster. Moreover, dropout(Srivastava *et al.*, 2014b) was utilized to minimize overfitting of the model.

Assuming  $p \in \{y, 1 - y\}$  and  $q \in \{\hat{y}, 1 - \hat{y}\}$ , dissimilarity between  $p$  and  $q$  can be calculated by measuring cross entropy between them. Eq. (4.5) shows the loss calculation method that was optimized for neural network training.

$$H(p, q) = - \sum_i p_i \log q_i = -y \log \hat{y} - (1 - y) \log(1 - \hat{y}) \quad (4.5)$$

**4.4.5. Testing.** For testing, we considered time-series detector data for identification of a source. Detector data was scanned at fixed time intervals. For this study, we have chosen two-second intervals for scanning. The model predicted the probability of a source using multiple different time-windows at successive positions. Multi-window prediction strategy can also be viewed as a technique called test time augmentation (TTA). TTA usually improves the performance of prediction algorithms; however, inference time also increases with this method. Finally, the decision of a source being present was made if the same source ID was detected consecutively. Further, we calculated the approximate location of the source by taking the mid-point of consecutive detected anchor points. Figure. 4.9 shows a schematic diagram of the testing procedure.

**4.4.6. Evaluation Metric.** The performance of the methods was quantified using the F1 measure. F1 measure is a balance between precision and recall and commonly used in multi-class classification problems. Precision can be interpreted as "how many selected items are relevant" while recall is "how many relevant items are selected." F1 measure can range from 0 to 1, where 1 means perfect precision and recall. Specifically, the macro-

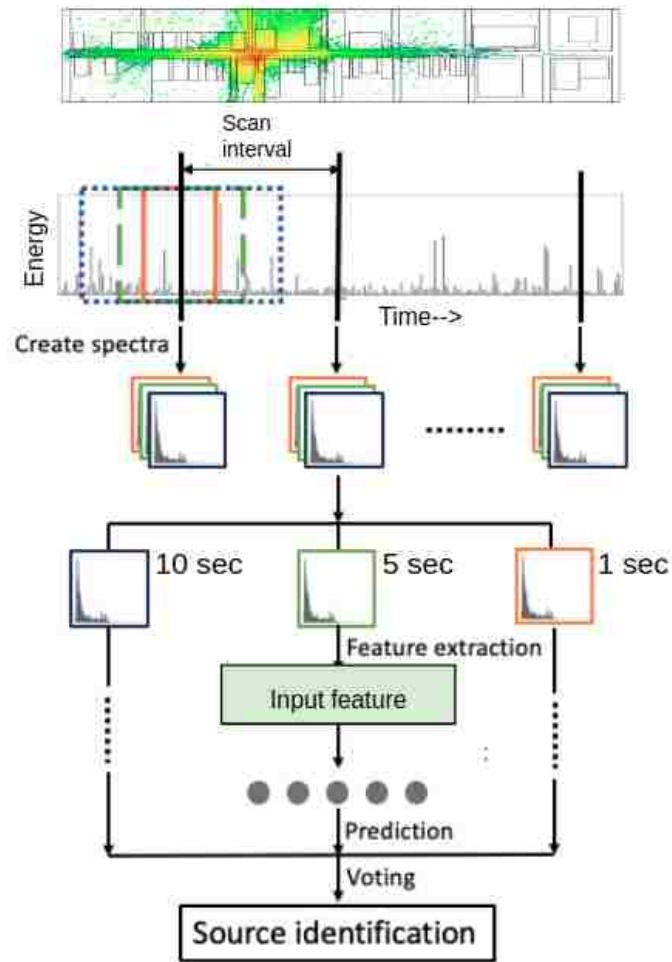


Figure 4.9. Multi-window prediction and voting strategy for radioisotope identification

F1 metric was used, which assigns equal importance to every class during the evaluation despite the class imbalance. Model training was repeated 20 times to calculate the estimated variance of the testing results.

$$\text{Precision} = \frac{\text{True positive}}{\text{True positive} + \text{False positive}} \quad (4.6)$$

$$\text{Recall} = \frac{\text{True positive}}{\text{True positive} + \text{False negative}} \quad (4.7)$$

$$F1 = \frac{2 \cdot \textit{precision} \cdot \textit{recall}}{\textit{precision} + \textit{recall}} \quad (4.8)$$

## 4.5. RESULTS

**4.5.1. Effect of Train Dataset Size.** An ML models robustness can be evaluated by measuring its performance on limited training data conditions. A robust model's accuracy should not decrease significantly with the decrease in training data samples. Moreover, a stable model is one that demonstrates less variance in prediction performance among different training sessions.

Figure. 4.10 demonstrates the mean F1 score of the 5 models when trained with varying train dataset sizes (i.e., 5%, 10%, 20%, 40%, 80% of train dataset). The models were trained and evaluated using 5-second spectrums. At 80% training size, all models show comparable accuracy. We observe that HNN1 and HNN2 models show high F1 score at low train percentages and achieve 5-20% improvement compared to the other three models when trained with only 5% of train data. The standard deviation for the HNN models is below 1% for difficult training conditions (i.e., low train percentages). However, RNN models show high standard deviation compared to others, which implies that they are not very stable predictive models.

When we compare the three feature sets, DCT and PR sets show higher accuracy for all models compared to SC features. DCT and PR sets provide 2-8% improvement over SC feature at 5% training data conditions for all the models. This implies that semi-automatic or manual feature extraction does help in the generalization of ML models.

**4.5.2. Effect of Spectra Acquisition Time.** Assuming a moving detector, a low data acquisition time (i.e., 0.2s) may result in highly noisy spectra, whereas high acquisition time (i.e., 20s) may suppress the signals. Figure. 4.11 show example spectrums of HEU+Tc-99m radioisotope at three different acquisition times (i.e., 1s, 5s and 20s).

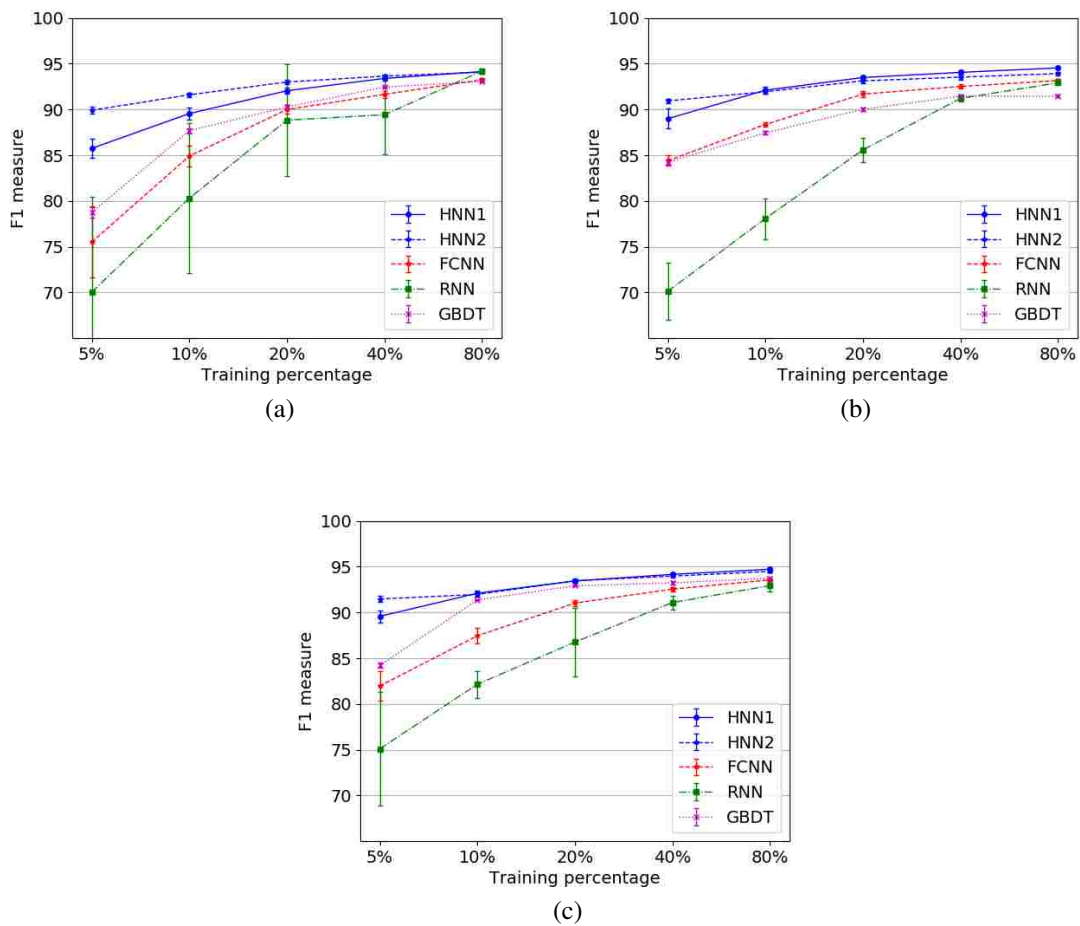
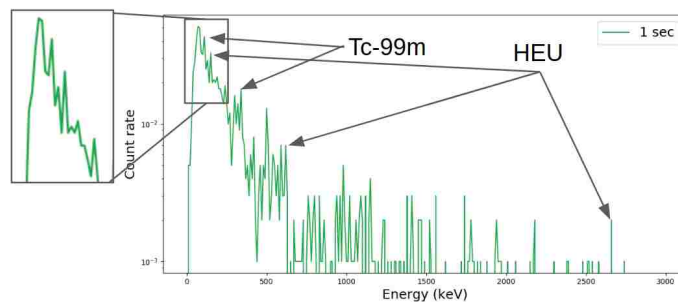


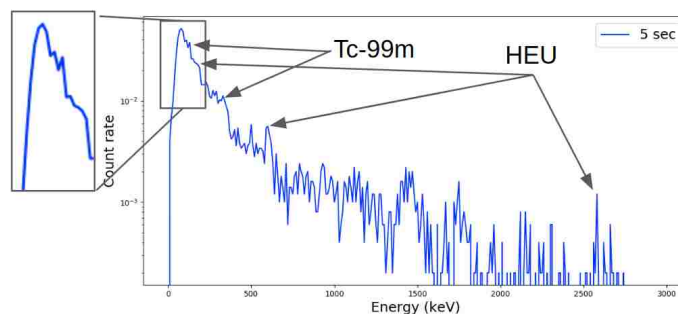
Figure 4.10. F1 score on test dataset predicted by five ML models, for feature set SC (a), DCT (b), and PR (c) on different training percentages

Figure 4.12 demonstrates a comparison among the models at varying data acquisition times. HNN models show higher accuracy than other models in most of the cases. At 0.2s acquisition time, HNN models achieve 2-12% improvement in F1 score than other three models for all feature sets. At the 20s, we observe 1-5% improvement of HNN models. HNN1 performed better than HNN2 in most of the conditions.

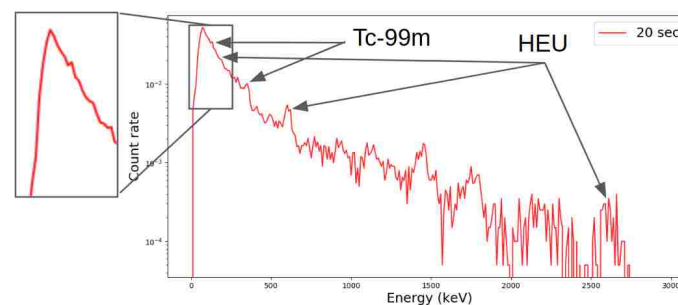
We also observe that PR features improved the performance of all models than the other two feature sets. PR and DCT features achieved closely 3-5% gain in F1 score at 0.2s spectra conditions compared to SC. All experiments were carried out using 80% train data.



(a) 1s spectra



(b) 5s spectra



(c) 20s spectra

Figure 4.11. Example spectra of HEU+Tc-99m at 1s, 5s and 20s acquisition time

**4.5.3. Run Time.** Figure. 4.13 demonstrates training and testing time for all five models. FCNN models are the fastest to train and test primarily because of their relatively simple architecture. RNN and HNN2 take almost 120x more time to train than FCNN models. HNN1 and GBDT models are five times slower than FCNN models during training. During testing, neural network models run time remained constant with the increment of training dataset size and usually faster than GBDT models. HNN1 is two times faster than HNN2 during test time. We observed a rise in inference time for GBDT models as the



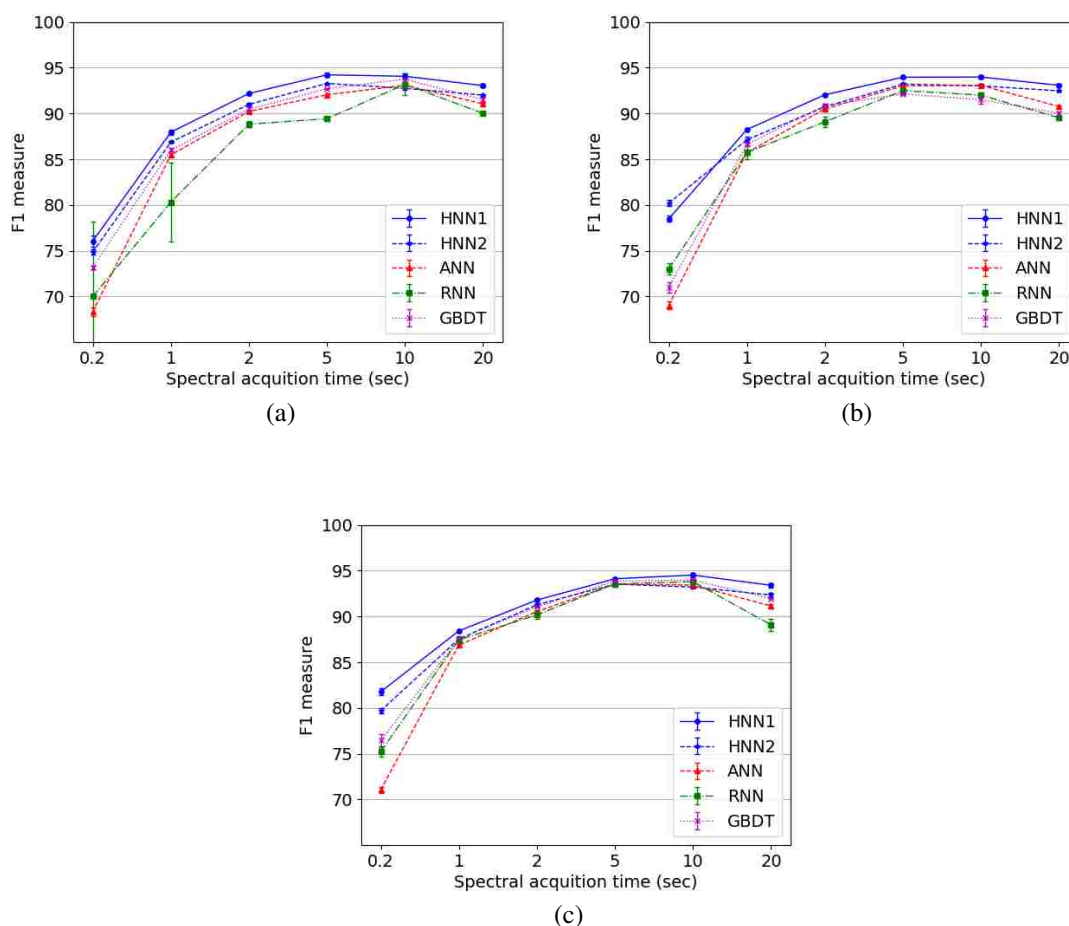


Figure 4.12. F1 score on test dataset predicted by five ML models, for feature set SC (a), DCT (b), and PR (c) on different spectra acquisition times. 80% train data was used for the experiments

training data increases. This is explainable as more data means more complex conditions can be fitted with the GBDT model, and the trees grow more in depth with more states, which in turn, increases the run time.

**4.5.4. Multi-class Classification Performance.** Figure 4.14 demonstrates F1 score for each individual radioisotope in the dataset for all five models. Models were trained using 80% of train data, 5s spectrums with SC feature set. We observed a relatively high accuracy for background, I-131, and Co-60 (ID 0, 3 and 4 respectively) radioisotopes compared to the other four combinations. Lower accuracy for HEU, WGPu, Tc-99m, and HEU+Tc-99m can

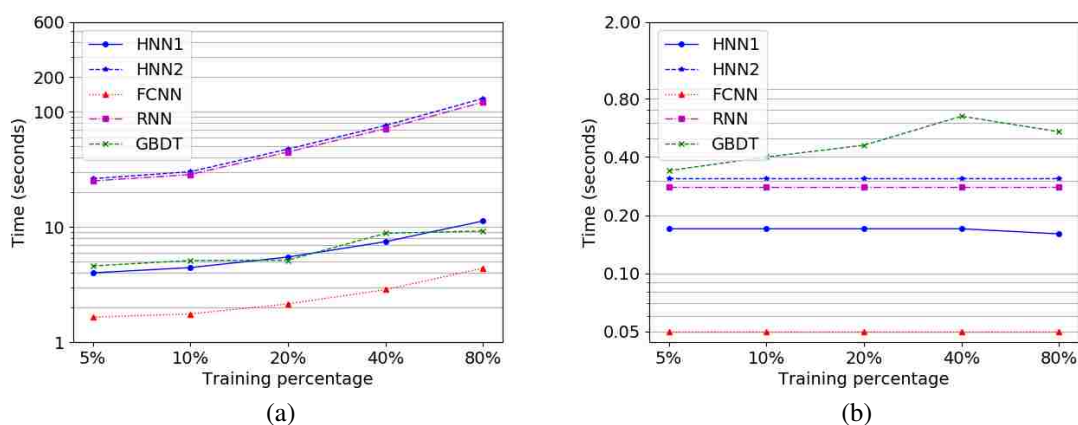


Figure 4.13. Run-time of the five models using 5s spectra acquisition time. (a) training time, (b) testing time with increasing train data percentages

be explained by analyzing their most active peak energies. These four radioisotopes have high activity peaks below 200 keV, which can be greatly affected by shielding conditions. Therefore, low peak energy isotopes are relatively difficult to detect.

To compare among the models, HNNs achieve higher F1 score than other models, especially for difficult radioisotopes and combinations (HEU, WGPu, Tc-99m, and HEU+Tc-99m). For multi-isotope scenario (ID 6: HEU+Tc-99m), HNNs achieve at least 2% improvement over the other models. Both HNN models perform comparably with a lower standard deviation than others. Moreover, for ID 0, 3, and 4, performance for all models are closely comparable.

Figure. 4.15 presents the confusion matrix of the 7-class classification results for HNN1 model. As expected, the majority of the misclassifications occurred between HEU and HEU+Tc-99m. Moreover, we observed that if a radioisotope is misclassified, it is more probable to be identified as background than other isotopes.

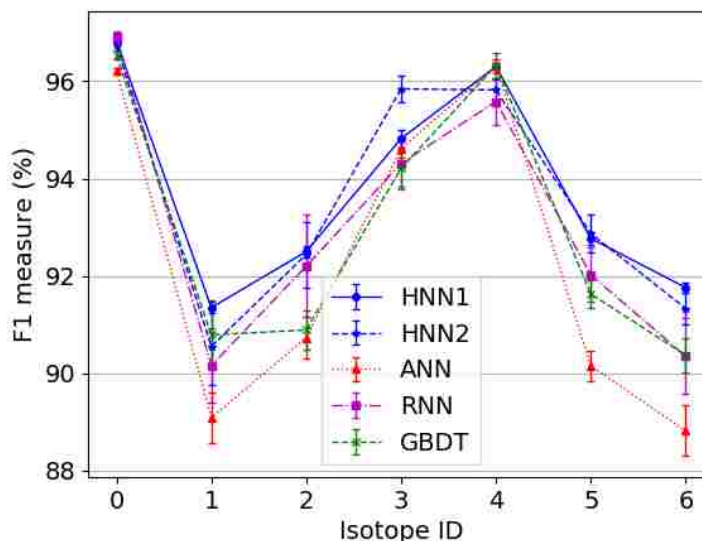


Figure 4.14. F1 score on test data for 7-class classification by five models using 5s spectrums and SC feature set

To further evaluate the classification performance of the models, we categorized the 6 radioisotopes into 3 categories: SNM (HEU, WGPu and HEU+Tc-99m), medical (I-131, Tc-99m) and industrial (Co-60). We observe that, detection accuracy for SNM was improved about 2% when considered as one category. One of the reasons behind this improvement possibly due to the peak similarity among the SNMs.

Most of the SNMs possess many of their essential photopeaks below the energy of 200 keV. The photopeaks may overlap if the detector resolution is low, and shielding conditions may suppress the presence of low energy peaks considerably. Therefore, detector response is vulnerable to similar conditions for SNMs. These characteristics make them alike for machine learning models and provide better results when considered together.

Figure 4.16 shows the results of all five models for four category classification. HNN models present higher accuracy and lower standard deviation than other models for most of the cases. Similar to previous results, we observe that when a category is misclassified, it has a higher probability to be classified as background (Figure 4.16(b)).

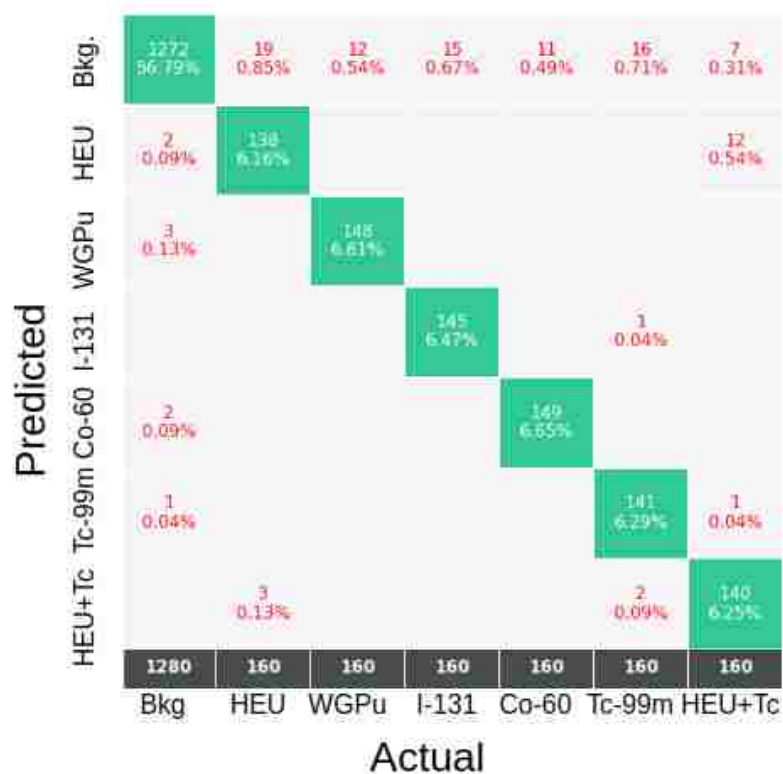


Figure 4.15. Confusion matrix for 7-class classification by HNN1 model using 5s spectrums

**4.5.5. Effect of Data Augmentation.** Table 4.4 shows the result of data augmentation using HNN1 model. Using 5 different augmentations (i.e., no. of time windows) for training and testing, prediction results were improved for more than 1% compared to using only 5s spectrums. Moreover, the uncertainty of the predictions became lower as the number of augmentations was increased. However, test time augmentation would increase the inference time as well.

#### 4.6. REMARKS

In this study, we have compared five state-of-the-art ML methods and three sets of feature calculation methods for automated radioisotope detection. Hybrid neural networks (HNNs) were developed based on FCNN, RNN, and CNN architectures. It is observed

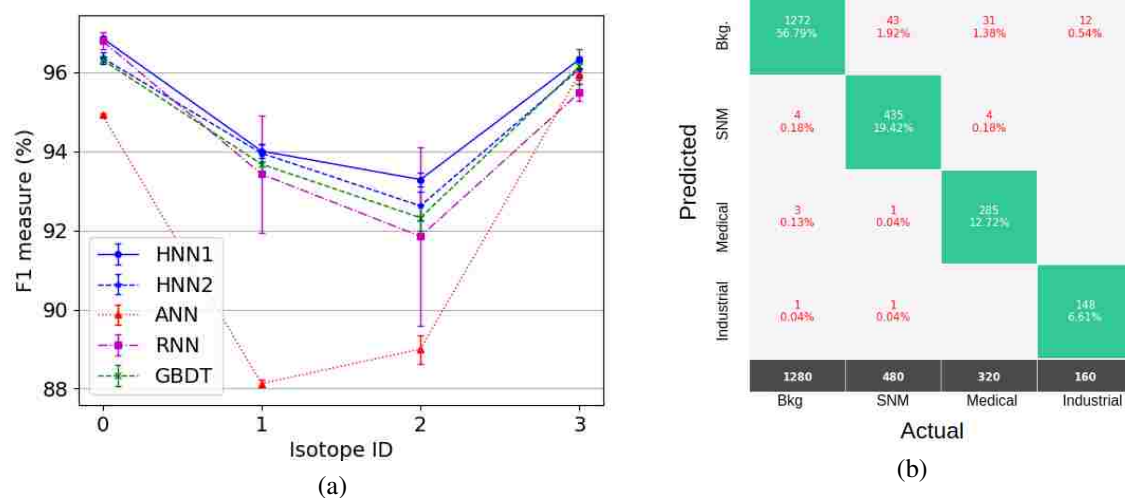


Figure 4.16. F1 score for 4-category (background (0), SNM (1), medical (2) and industrial (3)) radioisotope identification using 5s spectrums at test dataset. (a) individual F1 score for all five models, (b) confusion matrix for HNN1 model

Table 4.4. Effect of data augmentation for HNN1 model on test dataset

No. of augmentation	Acquisition time (s)	F1 score (%)
1	5	92.23 $\pm$ 0.68
2	5, 10	92.71 $\pm$ 0.32
3	5, 10, 20	93.04 $\pm$ 0.11
4	2, 5, 10, 20	93.27 $\pm$ 0.05
5	1, 2, 5, 10, 20	93.33 $\pm$ 0.08

that HNN models are more accurate and robust than traditional models at a wide range of parameter choices. Between the hybrid models, HNN1 (FCNN-CNN) showed comparable or better accuracy than HNN2 (FCNN-RNN) model for most of the cases, while running two times faster than the former. HNN models were evaluated based on the training dataset size and spectra acquisition time. In both of the cases, the models show 2-12% improvement in challenging prediction scenarios compared to best traditional models. Moreover, for hard

radioisotope combinations, such as multi-isotopes (i.e., HEU+Tc-99m) and their parent isotopes are detected 1-2% more accurately with HNN1 model. With data augmentation, HNN1 model was able to achieve 93.33% F1 score in test data.

One of the limitations of this study was that only simulated spectra were used for training and testing. The algorithm needs further verification using real-detector measured spectra. Besides, for best performance, handcrafted feature calculation remained important. Future ML models are expected to be capable of eliminating the explicit feature calculation step as this requires both domain knowledge and extra computation cost.

Overall, FCNN-CNN models found to be working best in the automated radioisotope classification task. This implies that deep learning method, such as CNN, is a useful and promising technique for gamma-ray spectroscopy. This also demonstrates that ML models tailored to a specific problem often leads to a better generalization of the algorithms.

## 5. CONCLUSIONS

This dissertation explores two application areas of machine learning where radiation generated data was analyzed. Both of the cases were treated as classification problems. Convolutional Neural Network (CNN) and Hybrid Neural Network (HNN) were developed for these purposes. We conclude that deep learning methods are powerful techniques for both image understanding and spectra unfolding problems.

In the first study, we have evaluated a method to quantify registration error from lung CT scans using a fast and lightweight deep learning algorithm. We have built and trained our algorithm using real-life clinical registrations, without the need for any synthetic data generation. The method was evaluated in a challenging scenario, by training on inhale-exhale image pairs but testing on image pairs from different treatment days, from a different source, and with a different registration algorithm. Results show the robustness of our algorithm despite the presence of large deformations in test data which were not present in the training data. Our approach is general, as it learns from raw data with no feature engineering required, which implies we can apply the same method to assess the quality of registrations across dataset from various imaging domains and possibly even across different body parts.

In the second study, five machine learning models were compared for their performance in automated radioisotope detection problem. Hybrid neural network models found to be more accurate in most of the cases compared to the conventional models at a wide range of parameter choices. This implies that machine learning models tailored to a specific problem often leads to better generalization. Furthermore, for all five models, we observed that explicit feature calculation performed better compared to using only raw data as features. This indicates that careful processing of data helps in building more accurate models, which would be unique for different cases.

Machine learning (ML) methods are useful techniques to automate varieties of tasks in real-world scenarios. With the increasing volume of digital data, more opportunities are creating for automated applications. ML technology is an essential tool for taking advantage of digital data and building the future world better and safer.



**APPENDIX A.**

**RADIOISOTOPE DETECTION PERFORMANCE IN TABLES**

Table 1. Effect of train data percentage on F1 score for five models

Model	Train percentage	SC		DCT		PR	
		mean (%)	std (%)	mean (%)	std (%)	mean (%)	std (%)
FCNN	80	93.25	0.22	93.16	0.21	93.56	0.12
	40	91.65	0.38	92.51	0.26	92.55	0.22
	20	89.99	0.46	91.68	0.29	91.02	0.32
	10	84.91	1.10	88.36	0.28	87.45	0.88
	5	75.51	3.86	84.39	0.57	81.96	1.60
RNN	80	94.19	0.29	92.91	0.28	92.93	0.63
	40	89.42	4.34	91.21	0.38	91.08	0.76
	20	88.83	6.11	85.58	1.33	86.76	3.77
	10	80.31	8.22	78.08	2.23	82.15	1.47
	5	70.03	10.45	70.13	3.15	75.10	6.18
GBDT	80	93.05	0.07	91.45	0.14	93.73	0.20
	40	92.45	0.11	91.46	0.10	93.24	0.22
	20	90.27	0.20	90.01	0.17	92.91	0.09
	10	87.68	0.27	87.45	0.21	91.37	0.04
	5	78.75	0.59	84.23	0.30	84.21	0.27
HNN1	80	94.14	0.16	94.54	0.12	94.72	0.14
	40	93.39	0.23	94.05	0.26	94.16	0.19
	20	92.04	0.30	93.49	0.21	93.44	0.23
	10	89.55	0.67	92.10	0.35	92.10	0.35
	5	85.75	1.01	88.89	1.08	89.57	0.66
HNN2	80	94.08	0.18	93.92	0.21	94.48	0.15
	40	93.64	0.21	93.53	0.28	93.97	0.20
	20	93.00	0.24	93.12	0.29	93.46	0.19
	10	91.60	0.21	91.93	0.22	91.97	0.23
	5	89.91	0.38	90.94	0.24	91.47	0.32

Table 2. Effect of spectra acquisition time on F1 score for five models

Model	Acquisition time (s)	SC		DCT		PR	
		mean (%)	std (%)	mean (%)	std (%)	mean (%)	std (%)
ANN	0.2	68.33	0.47	69.03	0.44	71.06	0.33
	1	85.47	0.22	85.73	0.29	86.88	0.15
	2	90.22	0.21	90.49	0.20	90.53	0.14
	5	93.04	0.12	93.01	0.11	93.52	0.14
	10	93.08	0.16	93.07	0.17	93.43	0.09
	20	91.04	0.18	90.75	0.15	91.14	0.16
RNN	0.2	70.00	0.00	73.01	0.64	75.23	0.54
	1	80.31	33.36	85.74	0.75	87.44	0.43
	2	88.83	0.33	89.08	0.56	90.17	0.41
	5	89.42	0.25	92.49	0.41	93.55	0.21
	10	93.19	1.17	92.00	0.29	93.80	0.07
	20	90.00	0.14	89.53	0.29	89.09	0.65
GBDT	0.2	73.17	0.62	71.10	0.55	76.43	0.71
	1	86.01	0.27	86.62	0.15	87.56	0.14
	2	90.46	0.15	90.86	0.19	91.04	0.05
	5	93.71	0.21	92.13	0.11	93.87	0.07
	10	93.75	0.28	91.50	0.41	93.97	0.20
	20	91.50	0.20	90.01	0.23	91.96	0.09
HNN1	0.2	76.05	0.15	78.47	0.33	81.77	0.39
	1	87.95	0.20	88.24	0.13	88.42	0.13
	2	92.18	0.32	92.04	0.16	91.79	0.09
	5	94.22	0.06	93.96	0.13	94.12	0.07
	10	94.06	0.07	93.98	0.22	94.52	0.11
	20	93.04	0.34	93.07	0.14	93.40	0.24
HNN2	0.2	74.85	0.29	80.19	0.32	79.70	0.28
	1	86.87	0.05%	87.12	0.35	87.53	0.19
	2	90.99	0.19%	90.73	0.20	91.27	0.18
	5	93.28	0.09%	93.21	0.14	93.53	0.10
	10	92.74	0.15%	93.01	0.19	93.21	0.17
	20	92.00	0.13%	92.47	0.12	92.34	0.24

Table 3. F1 score (%) on individual radioisotopes for five models

Model		Source ID						
		0	1	2	3	4	5	6
FCNN	mean	96.21	89.09	90.73	94.61	96.31	90.15	88.82
	std	0.07	0.52	0.43	0.17	0.15	0.30	0.52
RNN	mean	96.92	90.16	92.20	94.31	95.58	92.02	90.35
	std	0.09	0.76	1.05	0.52	0.49	0.56	0.78
GBDT	mean	96.53	90.79	90.89	94.21	96.32	91.63	90.37
	std	0.09	0.70	0.39	0.37	0.25	0.29	0.34
HNN1	mean	96.81	91.35	92.50	94.82	96.30	92.77	91.76
	std	0.06	0.11	0.12	0.18	0.07	0.12	0.11
HNN2	mean	96.73	90.55	92.44	95.84	95.82	92.87	91.33
	std	0.09	0.80	0.67	0.27	0.22	0.38	0.33

Table 4. F1 score (%) on categorical radioisotopes for five models

Model		Source category			
		Background	SNM	Medical	Industrial
FCNN	mean	94.92	88.13	89.00	95.96
	std	0.03	0.10	0.36	0.14
RNN	mean	96.79	93.42	91.85	95.48
	std	0.21	1.49	2.26	0.22
GBDT	mean	96.31	93.67	92.32	96.14
	std	0.04	0.29	0.32	0.45
HNN1	mean	96.86	94.01	93.28	96.32
	std	0.05	0.17	0.18	0.07
HNN2	mean	96.36	93.95	92.62	96.08
	std	0.14	0.22	0.37	0.22

## **APPENDIX B.**

### **PYTHON CODE FOR RADIOISOTOPE DETECTION ML MODELS**

## 1. PYTHON CODE FOR NEURAL NETWORK MODELS USING KERAS LIBRARY

```

1 # FCNN model
2 def init_model():
3     inp = Input(shape=(len(X[0]), 1))
4
5     a = Flatten()(inp)
6     a = Dense(8, kernel_initializer="uniform")(a)
7
8     output = Dense(7, activation="softmax", kernel_initializer="uniform")(a)
9     model = Model(inp, output)
10
11     return model
12
13
14 # RNN model
15 def init_model():
16     inp= Input(shape=(len(X[0]), 1))
17
18     a = CuDNNLSTM(100, return_sequences=True)(inp)
19     a = CuDNNLSTM(20)(a)
20     a = Dense(128, activation="relu", kernel_initializer="uniform")(a)
21
22     output = Dense(7, activation="softmax", kernel_initializer="uniform")(a)
23     model = Model(inp, output)
24
25     return model
26
27
28 # HNN1 model
29 def init_model():
30     inp = Input(shape=(len(X[0]), 1))
31
32     a = Conv1D(64, 5, activation="relu", kernel_initializer="uniform", )(inp)
33     a = BatchNormalization()(a)
34     a = Conv1D(64, 5, activation="relu", kernel_initializer="uniform", )(a)
35     a = BatchNormalization()(a)
36     max_pool = GlobalMaxPool1D()(a)

```

```

37
38     b = Flatten()(inp)
39     ab = concatenate([ max_pool, b])
40
41     a = Dense(128, activation="relu", kernel_initializer="uniform")(ab)
42     a = Dropout(0.5)(a)
43     a = Dense(128, activation="relu", kernel_initializer="uniform")(a)
44
45     output = Dense(7, activation="softmax", kernel_initializer="uniform")(a)
46     model = Model(inp, output)
47
48     return model
49
50
51 # HNN2 model
52 def init_model():
53     inp= Input(shape=(len(X[0]), 1))
54
55     a = CuDNNLSTM(100, return_sequences=True)(inp)
56     a = CuDNNLSTM(20)(a)
57
58     b = Flatten()(inp)
59     ab = concatenate([a, b])
60
61     a = Dense(128, activation="relu", kernel_initializer="uniform")(ab)
62     a = Dropout(0.5)(a)
63     a = Dense(128, activation="relu", kernel_initializer="uniform")(a)
64
65     output = Dense(7, activation="softmax", kernel_initializer="uniform")(a)
66
67     model = Model(inp, output)
68
69     return model

```

## 2. PARAMETERS FOR GBDT MODEL USING LIGHTGBM LIBRARY

```
1 params = {
2     'bagging_freq': 5,
3     'bagging_fraction': 0.6,
4     'boost_from_average': 'true',
5     'boost': 'gbdt',
6     'feature_fraction': 0.15,
7     'learning_rate': 0.0085,
8     'max_depth': 5,
9     'metric': 'multi_error',
10    'min_data_in_leaf': 5,
11    'min_sum_hessian_in_leaf': 10.0,
12    'num_leaves': 5,
13    'tree_learner': 'serial',
14    'objective': 'multiclass',
15    'verbosity': 1,
16    'num_class': 7,
17    'max_bin': 200,
18    'random_seed': random_seed
19 }
```



**APPENDIX C.**

**PUBLICATIONS**

## 1. PUBLICATION DISCLOSURE

Sections three and four of this dissertation will be published in peer-reviewed journal in the future. The probable journal titles are as follows:

Shaikat Galib, Hyoung Koo Lee, Christopher Guy, Matthew Riblett, Geoffrey Hugo, A Fast and Scalable method for Quality Estimation of Deformable Image Registration on Lung CT scans using Convolutional Neural Networks, Journal of Medical Physics.

Shaikat Galib, Palash Bhowmik , Ashish Avachat, Hyoung Koo Lee, A Comparison of Machine Learning Methods for Automated Radioisotope Identification using NaI Gamma-ray Spectra, Applied Radiation and Isotopes

## REFERENCES

- Alamaniotis, M., Heifetz, A., Raptis, A. C., and Tsoukalas, L. H., 'Fuzzy-Logic Radioisotope Identifier for Gamma Spectroscopy in Source Search,' *IEEE Transactions on Nuclear Science*, 2013, **60**(4), pp. 3014–3024, ISSN 0018-9499, doi: 10.1109/TNS.2013.2265307.
- Barzilov, A. and Novikov, I., 'Material classification by analysis of prompt photon spectra induced by 14-mev neutrons,' *Physics Procedia*, 2015, **66**, pp. 396 – 402, ISSN 1875-3892, doi:<https://doi.org/10.1016/j.phpro.2015.05.049>, the 23rd International Conference on the Application of Accelerators in Research and Industry - CAARI 2014.
- Batdorf, M., Hensley, W., Seifert, C., Kirihara, L., Erikson, L., and Jordan, D., 'Isotope identification in the GammaTracker handheld radioisotope identifier,' in '2009 IEEE Nuclear Science Symposium Conference Record (NSS/MIC),' IEEE, ISBN 978-1-4244-3961-4, 2009 pp. 868–872, doi:10.1109/NSSMIC.2009.5402463.
- Blackadar, J., 'Automatic isotope identifiers and their features,' *IEEE Sensors Journal*, 2005, **5**(4), pp. 589–592, ISSN 1530-437X, doi:10.1109/JSEN.2005.846374.
- Bobin, C., Bichler, O., Lourenço, V., Thiam, C., and Thévenin, M., 'Real-time radionuclide identification in  $\gamma$ -emitter mixtures based on spiking neural network,' *Applied Radiation and Isotopes*, 2016, **109**, pp. 405–409, ISSN 0969-8043, doi: 10.1016/J.APRADISO.2015.12.029.
- Burr, T. and Hamada, M., 'Radio-Isotope Identification Algorithms for NaI  $\gamma$  Spectra,' *Algorithms*, 2009, **2**, pp. 339–360, ISSN 1999-4893, doi:10.3390/a2010339.
- Castillo, E., Castillo, R., Martinez, J., Shenoy, M., and Guerrero, T., 'Four-dimensional deformable image registration using trajectory modeling,' *Physics in Medicine and Biology*, 2010, **55**(1), pp. 305–327, ISSN 0031-9155, doi:10.1088/0031-9155/55/1/018.
- Castillo, R., Castillo, E., Guerra, R., Johnson, V. E., McPhail, T., Garg, A. K., and Guerrero, T., 'A framework for evaluation of deformable image registration spatial accuracy using large landmark point sets,' *Physics in Medicine and Biology*, 2009, **54**(7), pp. 1849–1870, ISSN 0031-9155, doi:10.1088/0031-9155/54/7/001.
- Chandra, S. and Kokkinos, I., 'Fast, exact and multi-scale inference for semantic image segmentation with deep gaussian CRFs,' in 'Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics),' volume 9911 LNCS, ISBN 9783319464770, ISSN 16113349, 2016 pp. 402–418, doi:10.1007/978-3-319-46478-7\_25.

- Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., and Yuille, A. L., 'DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs,' *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017, pp. 1–1, ISSN 0162-8828, doi:10.1109/TPAMI.2017.2699184.
- Chen, T. and Guestrin, C., 'Xgboost: A scalable tree boosting system,' in 'Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining,' ACM, 2016 pp. 785–794.
- Chollet, F. *et al.*, 'Keras,' <https://keras.io>, 2015.
- C.J. Werner, e. a., 'Mcnp6.2 release notes,' Los Alamos National Laboratory, 2018, **report LA-UR-18-20808**.
- Connor, D., Martin, P., and Scott, T., 'Airborne radiation mapping: overview and application of current and future aerial systems,' *International journal of remote sensing*, 2016, **37**(24), pp. 5953–5987.
- datawow.io, 'Interns explain basic neural network - data wow,' <https://blog.datawow.io/interns-explain-basic-neural-network-ebc555708c9>, 2019, (Accessed on 07/09/2019).
- Datteri, R. D. and Dawant, B. M., 'Estimation and reduction of target registration error.' *Medical image computing and computer-assisted intervention : MICCAI ... International Conference on Medical Image Computing and Computer-Assisted Intervention*, 2012, **15**(Pt 3), pp. 139–46.
- Datteri, R. D., Liu, Y., D'Haese, P.-F., and Dawant, B. M., 'Validation of a non-rigid registration error detection algorithm using clinical MRI brain data.' *IEEE transactions on medical imaging*, 2015, **34**(1), pp. 86–96, ISSN 1558-254X, doi: 10.1109/TMI.2014.2344911.
- Dou, Q., Chen, H., Yu, L., Zhao, L., Qin, J., Wang, D., Mok, V. C., Shi, L., and Heng, P.-A., 'Automatic Detection of Cerebral Microbleeds From MR Images via 3D Convolutional Neural Networks,' *IEEE Transactions on Medical Imaging*, 2016, **35**(5), pp. 1182–1195, ISSN 0278-0062, doi:10.1109/TMI.2016.2528129.
- Dou, Q., Yu, L., Chen, H., Jin, Y., Yang, X., Qin, J., and Heng, P.-A., '3D deeply supervised network for automated segmentation of volumetric medical images,' *Medical Image Analysis*, 2017, **41**, pp. 40–54, ISSN 1361-8415, doi: 10.1016/J.MEDIA.2017.05.001.
- Elmaghraby, E. K., Tohamy, M., and Comsan, M., 'Determination of isotopes activity ratio using gamma ray spectroscopy based on neural network model,' *Applied Radiation and Isotopes*, 2019, **148**, pp. 19–26, ISSN 0969-8043, doi: 10.1016/J.APRADISO.2019.03.014.

- Eppenhof, K. A. J. and Pluim, J. P. W., 'Error estimation of deformable image registration of pulmonary CT scans using convolutional neural networks,' *Journal of Medical Imaging*, 2018, **5**(02), p. 1, ISSN 2329-4302, doi:10.1117/1.JMI.5.2.024003.
- Fischer, B., Dawant, B. M., and Lorenz, C., *Biomedical image registration : 4th International Workshop, WBIR 2010, Lübeck, Germany, July 11-13, 2010. Proceedings*, Springer-Verlag, 2010, ISBN 3642143652.
- Fitzpatrick, J. and West, J., 'The distribution of target registration error in rigid-body point-based registration,' *IEEE Transactions on Medical Imaging*, 2001, **20**(9), pp. 917–927, ISSN 02780062, doi:10.1109/42.952729.
- Fitzpatrick, J. M., 'Fiducial registration error and target registration error are uncorrelated [7261-1],' 2009, doi:10.1117/12.813601.
- Freund, Y., Schapire, R., and Abe, N., 'A short introduction to boosting,' *Journal-Japanese Society For Artificial Intelligence*, 1999, **14**(771-780), p. 1612.
- Ghafoorian, M., Karssemeijer, N., Heskes, T., Bergkamp, M., Wissink, J., Obels, J., Keizer, K., Leeuw, F.-E. d., Ginneken, B., Marchiori, E., and Platel, B., 'Deep multi-scale location-aware 3D convolutional neural networks for automated detection of lacunes of presumed vascular origin,' *NeuroImage: Clinical*, 2017, **14**, pp. 391–399, ISSN 2213-1582, doi:10.1016/J.NICL.2017.01.033.
- Gierga, D. P., Riboldi, M., Turcotte, J. C., Sharp, G. C., Jiang, S. B., Taghian, A. G., and Chen, G. T. Y., 'Comparison of target registration errors for multiple image-guided techniques in accelerated partial breast irradiation.' *International journal of radiation oncology, biology, physics*, 2008, **70**(4), pp. 1239–46, ISSN 0360-3016, doi:10.1016/j.ijrobp.2007.11.020.
- Girshick, R., Donahue, J., Darrell, T., and Malik, J., 'Region-based convolutional networks for accurate object detection and segmentation,' *IEEE transactions on pattern analysis and machine intelligence*, 2016, **38**(1), pp. 142–158.
- Goodfellow, I. J., Warde-Farley, D., Mirza, M., Courville, A. C., and Bengio, Y., 'Maxout Networks.' *ICML (3)*, 2013, **28**, pp. 1319–1327.
- Graves, A., 'Generating sequences with recurrent neural networks,' *arXiv preprint arXiv:1308.0850*, 2013.
- Graves, A., Mohamed, A.-r., and Hinton, G., 'Speech recognition with deep recurrent neural networks,' in 'Acoustics, speech and signal processing (icassp), 2013 IEEE international conference on,' *IEEE*, 2013 pp. 6645–6649.
- Graves, A., Wayne, G., and Danihelka, I., 'Neural Turing machines,' *arXiv preprint arXiv:1410.5401*, 2014.

- Guy, C. L., Weiss, E., Christensen, G. E., Jan, N., and Hugo, G. D., 'CALIPER: A deformable image registration algorithm for large geometric changes during radiotherapy for locally advanced non-small cell lung cancer,' *Medical Physics*, 2018a, **45**(6), pp. 2498–2508, ISSN 00942405, doi:10.1002/mp.12891.
- Guy, C. L., Weiss, E., Jan, N., Christensen, G. E., and Hugo, G. D., 'Technical Note: A method for quality assurance of landmark sets for use in evaluation of deformable image registration accuracy of lung parenchyma,' *Medical Physics*, 2018b, **46**(2), p. mp.13336, ISSN 0094-2405, doi:10.1002/mp.13336.
- Guy, C. L., Weiss, E., Jan, N., Christensen, G. E., and Hugo, G. D., 'A method for quality assurance of landmark sets for use in evaluation of deformable image registration accuracy of lung parenchyma,' *Medical physics*, 2019, **46**(2), pp. 766–773.
- He, J., Tang, X., Gong, P., Wang, P., Wen, L., Huang, X., Han, Z., Yan, W., and Gao, L., 'Rapid radionuclide identification algorithm based on the discrete cosine transform and BP neural network,' *Annals of Nuclear Energy*, 2018, **112**, pp. 1–8, ISSN 0306-4549, doi:10.1016/J.ANUCENE.2017.09.032.
- He, K., Zhang, X., Ren, S., and Sun, J., 'Spatial pyramid pooling in deep convolutional networks for visual recognition,' in 'European Conference on Computer Vision,' Springer, 2014 pp. 346–361.
- He, K., Zhang, X., Ren, S., and Sun, J., 'Deep residual learning for image recognition,' in 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,' 2016a pp. 770–778.
- He, K., Zhang, X., Ren, S., and Sun, J., 'Identity mappings in deep residual networks,' *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2016b, **9908 LNCS**, pp. 630–645, ISSN 16113349, doi:10.1007/978-3-319-46493-0\_38.
- Hermann, K. M., Kočiský, T., Grefenstette, E., Espeholt, L., Kay, W., Suleyman, M., and Blunsom, P., 'Teaching Machines to Read and Comprehend,' 2015a, pp. 1–9, ISSN 10495258, doi:10.1109/72.410363.
- Hermann, K. M., Kocisky, T., Grefenstette, E., Espeholt, L., Kay, W., Suleyman, M., and Blunsom, P., 'Teaching machines to read and comprehend,' in 'Advances in Neural Information Processing Systems,' 2015b pp. 1693–1701.
- Hinton, G. E. and Salakhutdinov, R. R., 'Reducing the dimensionality of data with neural networks.' *Science (New York, N.Y.)*, 2006, **313**(5786), pp. 504–7, ISSN 1095-9203, doi:10.1126/science.1127647.
- Hochreiter, S. and Schmidhuber, J., 'Long short-term memory.' *Neural computation*, 1997, **9**(8), pp. 1735–80, ISSN 0899-7667, doi:10.1145/2740908.2742760.

- Hosoda, M., Inoue, K., Oka, M., Omori, Y., Iwaoka, K., and Tokonami, S., 'Environmental radiation monitoring and external dose estimation in aomori prefecture after the fukushima daiichi nuclear power plant accident,' *Health Physics*, 2016, **51**(1), pp. 41–50.
- Hugo, G. D., Weiss, E., Sleeman, W. C., Balik, S., Keall, P. J., Lu, J., and Williamson, J. F., 'A longitudinal four-dimensional computed tomography and cone beam computed tomography dataset for image-guided radiation therapy research in lung cancer,' *Medical Physics*, 2017, **44**(2), pp. 762–771, ISSN 00942405, doi: 10.1002/mp.12059.
- Ioffe, S. and Szegedy, C., 'Batch normalization: Accelerating deep network training by reducing internal covariate shift,' arXiv preprint arXiv:1502.03167, 2015.
- Jegou, S., Drozdal, M., Vazquez, D., Romero, A., and Bengio, Y., 'The One Hundred Layers Tiramisu: Fully Convolutional DenseNets for Semantic Segmentation,' in 'IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops,' volume 2017-July, ISBN 9781538607336, ISSN 21607516, 2017 pp. 1175–1183, doi:10.1109/CVPRW.2017.156.
- Kalchbrenner, N., Grefenstette, E., and Blunsom, P., 'A convolutional neural network for modelling sentences,' arXiv preprint arXiv:1404.2188, 2014.
- Kamnitsas, K., Ledig, C., Newcombe, V. F., Simpson, J. P., Kane, A. D., Menon, D. K., Rueckert, D., and Glocker, B., 'Efficient multi-scale 3D CNN with fully connected CRF for accurate brain lesion segmentation,' *Medical Image Analysis*, 2017, **36**, pp. 61–78, ISSN 1361-8415, doi:10.1016/J.MEDIA.2016.10.004.
- Kamuda, M., Stinnett, J., and Sullivan, C. J., 'Automated Isotope Identification Algorithm Using Artificial Neural Networks,' *IEEE Transactions on Nuclear Science*, 2017, **64**(7), pp. 1858–1864, ISSN 0018-9499, doi:10.1109/TNS.2017.2693152.
- Kayalibay, B., Jensen, G., and van der Smagt, P., 'CNN-based Segmentation of Medical Imaging Data,' Technical report, 2017.
- Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., and Liu, T.-Y., 'Lightgbm: A highly efficient gradient boosting decision tree,' in 'Advances in Neural Information Processing Systems,' 2017 pp. 3146–3154.
- Kim, J., Park, K., and Cho, G., 'Multi-radioisotope identification algorithm using an artificial neural network for plastic gamma spectra,' *Applied Radiation and Isotopes*, 2019, **147**, pp. 83–90, ISSN 0969-8043, doi:10.1016/J.APRADISO.2019.01.005.
- Kingma, D. and Ba, J., 'Adam: A method for stochastic optimization,' arXiv preprint arXiv:1412.6980, 2014.
- Knoll, G. F., *Radiation detection and measurement*, John Wiley, 2010, ISBN 9780470131480.

- Krizhevsky, A., Sutskever, I., and Hinton, G. E., 'Imagenet classification with deep convolutional neural networks,' in 'Advances in neural information processing systems,' 2012 pp. 1097–1105.
- Latifi, K., Zhang, G., Stawicki, M., van Elmp, W., Dekker, A., and Forster, K., 'Validation of three deformable image registration algorithms for the thorax,' *Journal of Applied Clinical Medical Physics*, 2013, **14**(1), pp. 19–30, ISSN 15269914, doi:10.1120/jacmp.v14i1.3834.
- Long, J., Shelhamer, E., and Darrell, T., 'Fully convolutional networks for semantic segmentation,' in 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,' 2015 pp. 3431–3440.
- Lowekamp, B. C., Chen, D. T., Ibáñez, L., Blezek, D., and Johnson, H. J., 'The Design of SimpleITK,' 2013, doi:10.3389/fninf.2013.00045.
- Ma, B., Moghari, M. H., Ellis, R. E., and Abolmaesumi, P., 'On fiducial target registration error in the presence of anisotropic noise.' *Medical image computing and computer-assisted intervention : MICCAI ... International Conference on Medical Image Computing and Computer-Assisted Intervention*, 2007, **10**(Pt 2), pp. 628–35.
- Marstal, K., Berendsen, F., Staring, M., and Klein, S., 'SimpleElastix: A User-Friendly, Multi-lingual Library for Medical Image Registration,' in '2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW),' IEEE, ISBN 978-1-5090-1437-8, 2016 pp. 574–582, doi:10.1109/CVPRW.2016.78.
- Mikolov, T., Chen, K., Corrado, G., and Dean, J., 'Efficient estimation of word representations in vector space,' *arXiv preprint arXiv:1301.3781*, 2013.
- Mitchell, D. J. and Harding, L. T., 'GADRAS isotope ID user's manual for analysis of gamma-ray measurements and API for Linux and Android,' Technical report, Sandia National Laboratories (SNL), Albuquerque, NM, and Livermore, CA (United States), 2014, doi:10.2172/1177049.
- Mitchell, T. M., 'Artificial neural networks,' *Machine learning*, 1997, **45**, pp. 81–127.
- Moeskops, P., Veta, M., Lafarge, M. W., Eppenhof, K. A., and Pluim, J. P., 'Adversarial training and dilated convolutions for brain MRI segmentation,' in 'Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics),' volume 10553 LNCS, ISBN 9783319675572, ISSN 16113349, 2017 pp. 56–64, doi:10.1007/978-3-319-67558-9\_7.
- Moghari, M. H., Ma, B., and Abolmaesumi, P., 'LNCS 5242 - A Theoretical Comparison of Different Target Registration Error Estimators,' Technical report, 2008.
- Mucci, A., 'Drone tours in security systems,' 2016, uS Patent App. 14/516,651.



- Muenzing, S. E., van Ginneken, B., Murphy, K., and Pluim, J. P., 'Supervised quality assessment of medical image registration: Application to intra-patient CT lung registration,' *Medical Image Analysis*, 2012, **16**(8), pp. 1521–1531, ISSN 1361-8415, doi:10.1016/J.MEDIA.2012.06.010.
- Muenzing, S. E., van Ginneken, B., Viergever, M. A., and Pluim, J. P., 'DIRBoost—An algorithm for boosting deformable image registration: Application to lung CT intra-subject registration,' *Medical Image Analysis*, 2014, **18**(3), pp. 449–459, ISSN 1361-8415, doi:10.1016/J.MEDIA.2013.12.006.
- Murphy, K., van Ginneken, B., Klein, S., Staring, M., de Hoop, B., Viergever, M., and Pluim, J., 'Semi-automatic construction of reference standards for evaluation of image registration,' *Medical Image Analysis*, 2011a, **15**(1), pp. 71–84, ISSN 1361-8415, doi:10.1016/J.MEDIA.2010.07.005.
- Murphy, K., van Ginneken, B., Reinhardt, J. M., Kabus, S., Ding, K., Deng, X., Cao, K., Du, K., Christensen, G. E., Garcia, V., Vercauteren, T., Ayache, N., Commowick, O., Malandain, G., Glocker, B., Paragios, N., Navab, N., Gorbunova, V., Sporring, J., de Bruijne, M., Han, X., Heinrich, M. P., Schnabel, J. A., Jenkinson, M., Lorenz, C., Modat, M., McClelland, J. R., Ourselin, S., Muenzing, S. E. A., Viergever, M. A., De Nigris, D., Collins, D. L., Arbel, T., Peroni, M., Li, R., Sharp, G. C., Schmidt-Richberg, A., Ehrhardt, J., Werner, R., Smeets, D., Loeckx, D., Song, G., Tustison, N., Avants, B., Gee, J. C., Staring, M., Klein, S., Stoel, B. C., Urschler, M., Werlberger, M., Vandemeulebroucke, J., Rit, S., Sarrut, D., and Pluim, J. P. W., 'Evaluation of registration methods on thoracic CT: the EMPIRE10 challenge,' *IEEE Transactions on Medical Imaging*, 2011b, **30**(11), pp. 1901–1920, doi:10.1109/TMI.2011.2158349.
- Neylon, J., Min, Y., Low, D. A., and Santhanam, A., 'A neural network approach for fast, automated quantification of DIR performance,' *Medical Physics*, 2017, **44**(8), pp. 4126–4138, ISSN 00942405, doi:10.1002/mp.12321.
- NNSA, 'Detecting radiological threats in urban areas - challenge,' <https://www.topcoder.com/challenges/30085346>, 2019, (Accessed on 05/09/2019).
- Olmos, P., Diaz, J., Perez, J., Garcia-Belmonte, G., Gomez, P., and Rodellar, V., 'Application of neural network techniques in gamma spectroscopy,' *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 1992, **312**(1-2), pp. 167–173, ISSN 0168-9002, doi:10.1016/0168-9002(92)90148-W.
- Parkinson, P. M. S., Xu, H., Yu, P. L. H., Salvetti, D., Marelli, M., and Falcone, A. D., 'CLASSIFICATION AND RANKING OFFERMILAT GAMMA-RAY SOURCES FROM THE 3fgl CATALOG USING MACHINE LEARNING TECHNIQUES,' *The Astrophysical Journal*, 2016, **820**(1), p. 8, doi:10.3847/0004-637x/820/1/8.

- Peng, C., Zhang, X., Yu, G., Luo, G., and Sun, J., 'Large Kernel Matters – Improve Semantic Segmentation by Global Convolutional Network,' 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 1743–1751, doi: 10.1109/CVPR.2017.189.
- Pluim, J. P., Muenzing, S. E., Eppenhof, K. A., and Murphy, K., 'The truth is hard to make: Validation of medical image registration,' in '2016 23rd International Conference on Pattern Recognition (ICPR),' IEEE, ISBN 978-1-5090-4847-2, 2016 pp. 2294–2300, doi:10.1109/ICPR.2016.7899978.
- Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V., and Gulin, A., 'Catboost: unbiased boosting with categorical features,' in 'Advances in Neural Information Processing Systems,' 2018 pp. 6638–6648.
- Rangaswamvi, D., Sannappa, J., and Srinivasa, E., 'Estimation of radiological dose from radon, thoron and their progeny levels in the dwellings of shivamogga district, karnataka, india,' in 'Proceedings of the thirty-third IARP international conference on developments towards improvement of radiological surveillance at nuclear facilities and environment: book of abstracts,' 2018 .
- Redmon, J., Divvala, S., Girshick, R., and Farhadi, A., 'You only look once: Unified, real-time object detection,' in 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,' 2016 pp. 779–788.
- Ren, S., He, K., Girshick, R., and Sun, J., 'Faster r-cnn: Towards real-time object detection with region proposal networks,' in 'Advances in neural information processing systems,' 2015 pp. 91–99.
- Rohlfing, T., 'Image similarity and tissue overlaps as surrogates for image registration accuracy: widely used but unreliable,' IEEE Transactions on Medical Imaging, 2012, **31**(2), pp. 153–163, ISSN 02780062, doi:10.1109/TMI.2011.2163944.
- Rosenblatt, F., 'Principles of neurodynamics. perceptrons and the theory of brain mechanisms,' 1961, doi:10.21236/ad0256582.
- Runkle, R., Tardiff, M., Anderson, K., Carlson, D., and Smith, L., 'Analysis of spectroscopic radiation portal monitor data using principal components analysis,' IEEE Transactions on Nuclear Science, 2006, **53**(3), pp. 1418–1423, ISSN 0018-9499, doi:10.1109/TNS.2006.874883.
- Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L., 'ImageNet Large Scale Visual Recognition Challenge,' International Journal of Computer Vision, 2015, **115**(3), pp. 211–252, ISSN 0920-5691, doi:10.1007/s11263-015-0816-y.
- Sahiner, H., *Gamma spectroscopy by artificial neural network coupled with MCNP*, Ph.D. thesis, 2017.

- Saleh, Z. H., Apte, A. P., Sharp, G. C., Shusharina, N. P., Wang, Y., Veeraraghavan, H., Thor, M., Muren, L. P., Rao, S. S., Lee, N. Y., and Deasy, J. O., ‘The distance discordance metric—a novel approach to quantifying spatial uncertainties in intra- and inter-patient deformable image registration,’ *Physics in medicine and biology*, 2014, **59**(3), pp. 733–746, ISSN 1361-6560, doi:10.1088/0031-9155/59/3/733, 24440838[pmid].
- Samuel, A. L., ‘Some studies in machine learning using the game of checkers,’ *IBM Journal of Research and Development*, 1959, **3**(3), pp. 210–229, ISSN 0018-8646, doi:10.1147/rd.33.0210.
- Sanada, Y. and Torii, T., ‘Aerial radiation monitoring around the fukushima dai-ichi nuclear power plant using an unmanned helicopter,’ *Journal of environmental radioactivity*, 2015, **139**, pp. 294–299.
- Saygili, G., Staring, M., and Hendriks, E. A., ‘Confidence Estimation for Medical Image Registration Based On Stereo Confidences,’ *IEEE Transactions on Medical Imaging*, 2016, **35**(2), pp. 539–549, ISSN 0278-0062, doi:10.1109/TMI.2015.2481609.
- Seginer, A., ‘Rigid-body point-based registration: The distribution of the target registration error when the fiducial registration errors are given.’ *Medical image analysis*, 2011, **15**(4), pp. 397–413, ISSN 1361-8423, doi:10.1016/j.media.2011.01.001.
- Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., and LeCun, Y., ‘OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks,’ 2013, ISSN 10636919, doi:10.1109/CVPR.2015.7299176.
- Sokooti, H., de Vos, B., Berendsen, F., Lelieveldt, B. P. F., Išgum, I., and Staring, M., ‘Non-rigid Image Registration Using Multi-scale 3D Convolutional Neural Networks,’ pp. 232–239, Springer, Cham, 2017, doi:10.1007/978-3-319-66182-7\_27.
- Sokooti, H., Saygili, G., Glocker, B., Lelieveldt, B. P. F., and Staring, M., ‘Accuracy Estimation for Medical Image Registration Using Regression Forests,’ pp. 107–115, Springer, Cham, 2016, doi:10.1007/978-3-319-46726-9\_13.
- Srivastava, N., Hinton, G., Krizhevsky, A., and Salakhutdinov, R., ‘Dropout: A Simple Way to Prevent Neural Networks from Overfitting,’ Technical report, 2014a.
- Srivastava, N., Hinton, G. E., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R., ‘Dropout: a simple way to prevent neural networks from overfitting.’ *Journal of Machine Learning Research*, 2014b, **15**(1), pp. 1929–1958.
- stanford.edu, *CS231n Convolutional neural networks for visual recognition*, 2019 (accessed July 7, 2019), <http://cs231n.github.io/convolutional-networks/>.
- Stinnett, J., Sullivan, C. J., and Xiong, H., ‘Uncertainty Analysis of Wavelet-Based Feature Extraction for Isotope Identification on NaI Gamma-Ray Spectra,’ *IEEE Transactions on Nuclear Science*, 2017, **64**(7), pp. 1670–1676, ISSN 0018-9499, doi:10.1109/TNS.2017.2676045.

- Sullivan, C., Garner, S., Lombardi, M., Butterfield, K., and Smith-Nelson, M., 'Evaluation of key detector parameters for isotope identification,' in '2007 IEEE Nuclear Science Symposium Conference Record,' IEEE, ISBN 978-1-4244-0922-8, 2007 pp. 1181–1184, doi:10.1109/NSSMIC.2007.4437217.
- Sutskever, I., Vinyals, O., and Le, Q. V., 'Sequence to sequence learning with neural networks,' in 'Advances in neural information processing systems,' 2014 pp. 3104–3112.
- Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z., 'Rethinking the inception architecture for computer vision,' in 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,' 2016 pp. 2818–2826.
- TURING, A. M., 'I.ÃŦCOMPUTING MACHINERY AND INTELLIGENCE,' *Mind*, 1950, **LIX**(236), pp. 433–460, ISSN 0026-4423, doi:10.1093/mind/LIX.236.433.
- Uekusa, Y., Nabeshi, H., Nakamura, R., Tsutsumi, T., Hachisuka, A., Matsuda, R., and Teshima, R., 'Surveillance of radioactive cesium in domestic foods on the japanese market (fiscal years 2012 and 2013),' *Shokuhin eiseigaku zasshi. Journal of the Food Hygienic Society of Japan*, 2015, **56**(2), pp. 49–56.
- Wan, L., Zeiler, M., Zhang, S., Cun, Y. L., and Fergus, R., 'Regularization of neural networks using dropconnect,' in S. Dasgupta and D. McAllester, editors, 'Proceedings of the 30th International Conference on Machine Learning,' volume 28 of *Proceedings of Machine Learning Research*, PMLR, Atlanta, Georgia, USA, 2013 pp. 1058–1066.
- Wang, D., Zhang, Y., and Zhao, Y., 'Lightgbm: An effective mirna classification method in breast cancer patients,' in 'Proceedings of the 2017 International Conference on Computational Biology and Bioinformatics,' ICCBB 2017, ACM, New York, NY, USA, ISBN 978-1-4503-5322-9, 2017 pp. 7–11, doi:10.1145/3155077.3155079.
- Wiedner, H., Peyrés, V., Crespo, T., Mejuto, M., García-Toraño, E., and Maringer, F. J., 'Application of an artificial neural network for evaluation of activity concentration exemption limits in NORM industry,' *Applied Radiation and Isotopes*, 2017, **126**, pp. 289–292, ISSN 0969-8043, doi:10.1016/J.APRADISO.2016.12.044.
- Zhang, C., Hu, G., Luo, F., Xiang, Y., Ding, G., Chu, C., Zeng, J., Ze, R., and Xiang, Q., 'Identification of SNM based on low-resolution gamma-ray characteristics and neural network,' *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 2019, **927**, pp. 155–160, ISSN 0168-9002, doi:10.1016/J.NIMA.2019.02.023.
- Zhao, H., Shi, J., Qi, X., Wang, X., and Jia, J., 'Pyramid Scene Parsing Network,' in '2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR),' IEEE, ISBN 978-1-5386-0457-1, 2017 pp. 6230–6239, doi:10.1109/CVPR.2017.660.

## VITA

Shaikat Mahmood Galib received a PhD degree in December 2019 in Nuclear Engineering from Missouri University of Science and Technology (Missouri S&T). His doctoral research investigates machine learning techniques in nuclear medicine and security applications. He received his BS degree in Mechanical Engineering from Islamic University of Technology, Bangladesh and has been awarded OIC scholarship. He worked in a 110 MW power plant and in an automotive service company in the past. He also received an MS degree in Nuclear Engineering from Missouri S&T. Apart from work, he enjoyed playing Cricket and appointed as a captain of Bangladesh team at Missouri S&T.