9-7-2021

Efficient Neuromorphic Algorithms for Gamma-Ray Spectrum Denoising and Radionuclide Identification

Merlin Phillip Carson
Portland State University

Recommended Citation

Carson, Merlin Phillip, "Efficient Neuromorphic Algorithms for Gamma-Ray Spectrum Denoising and Radionuclide Identification" (2021). *Dissertations and Theses.* Paper 5812.

https://doi.org/10.15760/etd.7683

Efficient Neuromorphic Algorithms for Gamma-Ray Spectrum Denoising and Radionuclide Identification.

by

Merlin Phillip Carson

A thesis submitted in partial fulfillment of the
requirements for the degree of

Master of Science
in
Computer Science

Thesis Committee:
Christof Teuscher, Chair
Bart Massey
John Lipor

Portland State University
2021

Abstract

Radionuclide detection and identification are important tasks for deterring a potentially catastrophic nuclear event. Due to high levels of background radiation from both terrestrial and extraterrestrial sources, some form of noise reduction pre-processing is required for a gamma-ray spectrum prior to being analyzed by an identification algorithm so as to determine the identity of anomalous sources. This research focuses on the use of neuromorphic algorithms for the purpose of developing low power, accurate radionuclide identification devices that can filter out non-anomalous background radiation and other artifacts created by gamma-ray detector measurement equipment, along with identifying clandestine, radioactive material.

A sparse coding optimization solver, the Simple Spiking Locally Competitive Algorithm, is investigated and simulated for the tasks of radionuclide detection and identification. A convolutional neural network is used to filter the input signal to the identification algorithm to remove background radiation and detector noise. Both algorithms are designed to be neuromorphic, implemented in hardware using memristive devices, thus significantly reducing their necessary power consumption compared to software implementations.

The radionuclide identification algorithm is compared to Gamma Detector Response and Analysis Software, an industry standard package that is developed by Sandia National Laboratories. Our neuromorphic algorithm achieves a 91% accuracy with a high resolution detector and an 89% accuracy with a low resolution detector on the corresponding measured gamma-ray spectra test sets, both less than 2% below the benchmark, state of the art algorithm's performance on the same spectra. To determine the efficacy of using a neural network for background and noise reduction, identification results are compared between gamma-ray spectra with no noise reduction, the traditional standard of background subtraction, and using the presented convolutional neural network for denoising. Finally, the power consumption of the proposed neuromorphic algorithms is estimated and compared to the empirically determined power consumption of the Gamma Detector Response and Analysis Software, showing that they can achieve the same task with over a 99% reduction in power.

Acknowledgements

First, I would like to thank my advisor Christof Teuscher who over three and a half years ago gave me the opportunity to get involved in research related to neuromorphic and radiological algorithms. His advice and direction over the last several years has helped me become a much better researcher and technical writer. I would also like to thank Adam Hecht who has taught me so much about nuclear physics. Without the knowledge he has imparted to me over the last several years none of this research would have been possible. And I would like to thank Walt Woods who helped with the initial research and development of the underlying identification algorithm, introducing me to locally competitive algorithms.

Next, I would like to thank all my professors that helped advance my understanding of the mathematical and algorithmic fundamentals of machine learning and deep learning techniques along with pushing my research abilities. Specifically, I would like to thank Anthony Rhodes, Ted Willke, Suresh Singh, John Lipor, and Meysam Asgari. Additionally, I'd like to thank Karen Karavanic and Feng Liu for allowing me to do neural network projects in classes that are not focused on machine learning. I would also like to thank Bart Massey for always having an open door for talking about algorithms and signal processing with me. His initial support of my thesis topic and discussion of radionuclide algorithms gave me the confidence to kick-start the research and development for much of this work.

I would like to thank my former colleagues Nick Metzar, Matt Kotvis, and Karthiek

Bokka for getting me starting with developing denoising algorithms using deep learning. They all taught me so much and their trust and motivation allowed me to push my development skills more in the few years that I worked with them then I ever could have imagined.

Finally, and most importantly, I would like to thank my parents, Peter and Jeanne Carson, and brother, Nathan Carson, who have always been extremely supportive of all my life pursuits, especially in regards to my interest in computers, math, and science and my decision to pursue higher education.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

Due to the rapid increase in nuclear proliferation over the past several decades, asymmetric warfare has become a significant threat across the globe. Radionuclide detection and identification algorithms are paramount tools for the prevention of a preemptive strike. Radiation is commonplace, from both manufactured and *Naturally Occurring Radioactive Material* (NORM), with wide ranging uses from building materials to medical devices and treatments, even some healthy foods contain traceable amounts. Thus, radiation identification algorithms suffer due to the high amount of background radiation present in a measured gamma-ray spectrum. Therefore, it is desirable to have an efficient method for the removal of background so that unwanted *Special Nuclear Material* (SNM), used to create dangerous radiological devices, can be identified in low signal to noise gamma-ray spectra.

As the number of ports of entry and thus the number of people traversing these points has increased, there has been a significant increase in the demand for portal monitoring devices to dissuade and stop foreign terrorists from the smuggling of radioactive material across borders. Inspired by successful attacks such as 9/11, domestic terrorism and homegrown violent extremism have also become a major concern and focus for law enforcement agencies such as the FBI and *Defense Threat*

Reduction Administration (DTRA). For these types of threats, portal monitoring devices are no longer adequate since targets are often large outdoor gatherings. Additionally, the technology used for radiation detection and identification devices consume large amounts of power and thus need frequent charging or a constant power supply. So a low power, portable *Radiation Identification Device* (RID) is critical for long-term, independent monitoring.

Recent advances in neural network research have provided for the development of new, more efficient algorithms for the detection and identification of radionuclides. However, while these algorithms are significantly less computationally expensive then traditional methods, they have large numbers of parameters and require computationally complex matrix operations. Therefore, they still consume notable amounts of computational resources and thus power.

In 1971 Leon Chua described a potential fourth classical circuit element: the Memristor [2], a variable resistive nano-device which was theorized to have unique mathematical properties. Advances in manufacturing techniques over the last decade have allowed for the development of these devices into crossbar arrays [3], giving researchers the chance to empirically test and prove their computation and power efficiency [4]. More recently, artificial neural networks and convolutional neural networks have been built and tested using memristors [5, 6] due to their ability to emulate synaptic behavior, resulting in significantly faster and more power efficient versions of these traditionally software-based algorithms.

1.2 Overview of this Document

The focus of this thesis is low-power radiation detection for threat reduction. There are two main algorithms that are researched and referenced throughout this manuscript to achieve this task: radionuclide identification and gamma-ray spectrum denoising.

This chapter provides background on the gamma-ray spectrum and radionuclide identification. It also lists the contributions of this manuscript. Chapter 2 covers a neuromorphic radionuclide identification algorithm using traditional background removal, and quantifies its accuracy and power consumption in relationship to an industry standard tool. Chapter 3 covers a neuromorphic gamma-ray spectrum denoising algorithm and quantifies the model's training and evaluation metrics along with its power consumption. Chapter 4 compares the performance of the neuromorphic radionuclide identification algorithm using various denoising algorithms. The final chapter, Chapter 5, discusses major takeaways from this research and future work.

1.2.1 Gamma-Rays and the Photoelectric Effect Background

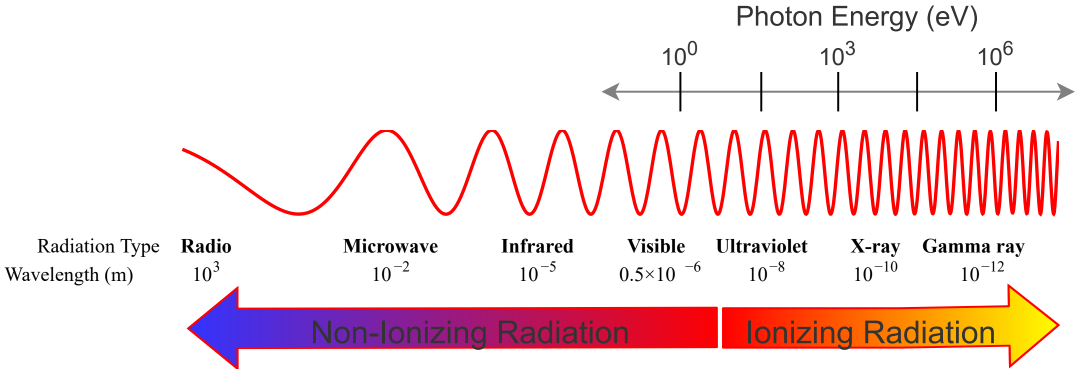

Figure 1.1: Electromagnetic radiation spectrum.

The electromagnetic spectrum describes the frequency of a wide range of electromagnetic radiation waves emitted by charged particles undergoing acceleration. On the low end, with long wavelengths, there are radio-waves. On the high end, with short wavelengths, there are gamma-rays. A variety of these types of waves and their wavelengths can be seen in Fig. 1.1. From just above visible light and below, these waves create what is known as non-ionizing radiation, which has high enough energy to transfer some to an electron when passing through it, but not high enough to remove it from its orbit around a nucleus. Increasing above non-ionizing radiation are types of electromagnetic waves that have higher frequencies and thus have enough energy to detach electrons for atoms and molecules. This is known as ionizing radiation which can results in drastic changes to physical material. In high enough doses, this type of radiation causes cell damage, radiation burns, and sickness. Even small exposures over a long period of time can lead to the growth of cancerous tissue.

Radioactive material consists of radionuclides, unstable atoms that emit high energy electromagnetic waves as the material decays. The most common energy emitted from this type of material are gamma-rays, discovered by French chemist and physicist Paul Ulrich Villard in 1900 while studying the radiation emanating from radium [7]. Gamma-rays occur in the photon energy range of 10^4 to 10^7 eV, or more specifically a few keV to $\sim$8 MeV. With this much energy, a gamma-ray incident with an electron can knock it out of its orbit around a nucleus, as depicted in Fig. 1.2, transferring all its energy to the electron in a process known as the photoelectric effect.

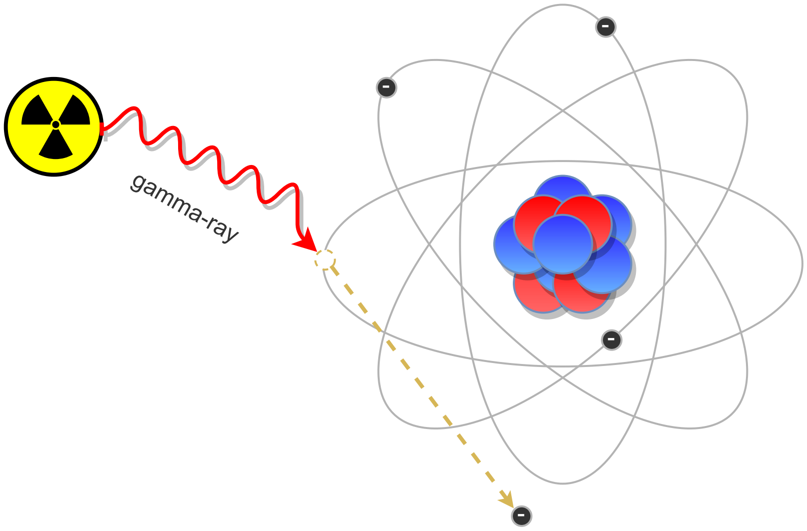

Figure 1.2: Photoelectric effect, where decaying nuclear material releases a gamma-ray that knocks an electron from the orbit of a nucleus, transferring all its energy to the electron.

1.2.2 Radionuclide Identification Background

Each type of radionuclide produces gamma-rays at multiple, specific energy levels, and thus creates a different spectrum of energy. However, the gamma-rays emitted from a decaying radionuclide are considered Poisson distributed, and thus a theoretical spectrum for a specific type of radionuclide consists of the probability of a gamma-ray at a specific energy level given some number of emissions, often referred to as hits or counts. Additionally, while as a whole each radionuclide type has a unique set of energy levels and associated probabilities, many radionuclides have common energy levels at which gamma-rays are emitted. So radionuclide identification cannot be determined by a specific measured energy level, and instead must consider measurements at several energy levels within the spectrum.

The photoelectric effect can be measured using a gamma-ray detector, and there are a variety of detector types which use different unique materials or compounds that have intrinsic properties that allow them to collect and measure photoelectric energy. However, each of these types of detectors have their own characteristic properties, such as energy resolution, which determines how accurately a specific energy level is detected, and detector efficiency, which determines how efficient the detector is at collecting energy at different levels. Therefore, radionuclide identification algorithms must be customized or tuned to work with the specific type of detector that will be used for gamma-ray measurements.

In addition to the above mentioned difficulty in identifying radionuclides based on shared energy levels and characteristic properties of the measurement devices, there is also an issue with noise that must be considered. First and foremost, there is background radiation all around us. Building materials such as sandstone, concrete, brick, natural stone, gypsum, and granite all contain naturally-occurring radioactive elements like radium, uranium, and thorium. Healthy foods such as Brazilian nuts and bananas contain traceable amounts of radium and potassium-40 absorbed from below the soil through plant roots. A variety of radionuclides are used in medical diagnostics and therapeutics. Radiation is even coming from the sky in the form of cosmic-rays from the sun and outside our solar system, some as old as the universe itself created during the Big Bang. There is also some amount of inherent noise in gamma-ray detectors themselves, especially at low keV levels where detector efficiency is high and there is overlap between X-rays and gamma-rays. Therefore, it is imperative that some form of noise reduction and background radiation removal is performed to a gamma-ray spectrum in order to determine the presence of anomalous and dangerous radionuclides so they are not masked by noise and background.

1.3 Contributions

This research focuses on two major tasks, radionuclide identification and gamma-ray spectrum denoising. The radionuclide identification is an application of the *Simple Spiking Locally Competitive Algorithm* (SSLCA) [1], developed by Portland State University alumnus Walt Woods. All research related to applying the SSLCA to radionuclide identification appears in Carson *et al.* [8] which was published on March 16th of 2021 in the journal IEEE Transactions on Nuclear Science. The gamma-ray spectrum denoising algorithm is my original idea and solely developed by myself using convolutional neural networks, inspired by past research projects I have worked on related to audio and image denoising.

1.3.1 Radionuclide Identification using Gamma-Ray Spectra

There exist a variety of methods for identifying radionuclides from gamma-ray spectrum. However, this research presents the first published work on using neuromorphic computing for the task. For my part in this research, I will present the development of a radionuclide template dictionary to prepare the SSLCA for this type of classification. I compare the power consumption of the neuromorphic implementation of the SSLCA to that of an industry standard, *State of the Art* (SOTA) identification tool *Gamma Detector Response and Analysis Software* (GADRAS), showing that the SSLCA consumes significantly less power. Finally, I present empirical results for both the SSLCA and GADRAS, showing that the SSLCA is capable of near SOTA performance identifying radionuclides from gamma-ray spectrum.

1.3.2 Gamm-Ray Spectrum Denoising for Radionuclide Identification

To my knowledge there is no published work or application of using neural networks for gamma-ray spectrum denoising. Therefore, I discuss past uses of convolutional neural networks for other types of signal denoising tasks and why they are a natural choice for this application. I also discuss why gamma-ray spectrum denoising is a critical procedure and the advantages and disadvantages to both this approach and the traditional approach of background subtraction. I go over the training data generation procedure that I have developed. I cover the model hyperparameters that I have chosen, along with the tuning and training process that I have implemented. And finally, I test the model by comparing the identification results with no noise reduction, background subtraction, and my gamma-ray spectrum denoising algorithm using the SSLCA.

Chapter 2

Radionuclide Identification

2.1 Simple Spiking Locally Competitive Algorithm Overview

The *Simple Spiking Locally Competitive Algorithm* (SSLCA) [1] is an extension of the *Locally Competitive Algorithm* (LCA) proposed by Rozell *et al.* in 2008 [9]. The algorithms are optimal solvers for sparse coding problems, linearly decomposing a composite input signal into its constituent parts, based on a set of templates. It is a neuromorphic algorithm that processes data in a fashion inspired by the spiking of neurons that occurs in biological brains. Memristors are used as a compact, energy-efficient representation of synapses [10], attenuating the spiking signal that is passed through the system. Memristors are nanodevices that change their resistance based on the amount and direction of voltage passed through them over time, and maintain their state when no power is present [11]. Inhibitory forces are used to create competition in the system, preventing nodes from firing that are already represented by the combination of active nodes. This results in sparse data representation and a reduction in power consumption.

Fig. 2.1 shows the proposed architecture for a RID which uses the SSLCA as a classifier. Energy from a radionuclide source is collected with a gamma-ray detector. The pulses of the gamma-ray detector are histogrammed as a spectrum by a *Multi-Channel Analyzer* (MCA) and fed to the input of the SSLCA. The input is converted

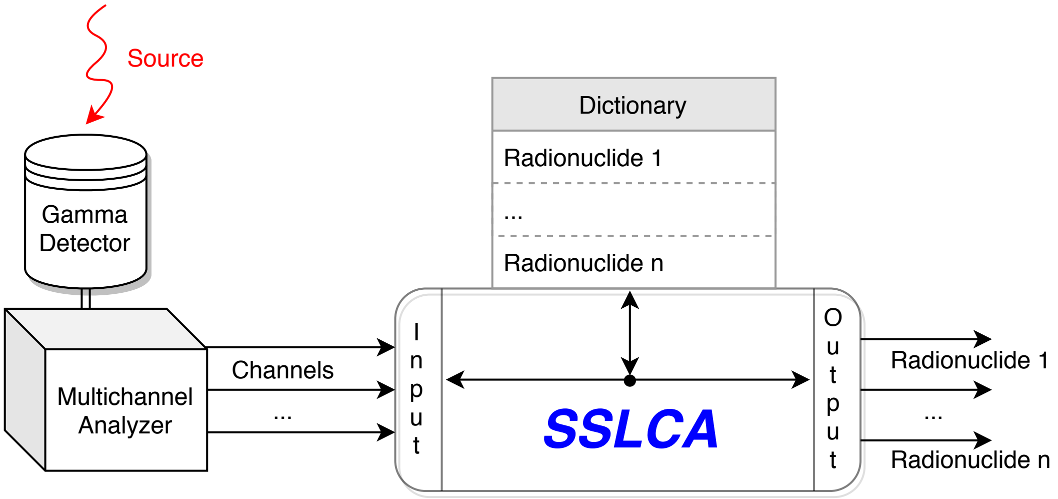

Figure 2.1: The proposed architecture for a RID utilizing the SSLCA as a classifier. Energy from a radionuclide source is collected within a gamma-ray detector. The pulses from the detector are passed to a multichannel analyzer. The output channels of the MCA are fed to the input of the SSLCA. The signal is compared to a dictionary of radionuclides. A separate output is triggered for each radionuclide identified in the signal.

to a spike train, only triggering the inputs when a signal that is not represented by an activated output is present. The signal is passed through a memristive crossbar, performing an approximation of a dot product with the entries in a radionuclide template dictionary. When a strong enough signal is reached at an output, representing a high similarity between the incoming signal and an entry in the radionuclide dictionary, a unique output is triggered for each radionuclide that has a high likelihood of being present in the signal.

Fig. 2.2 shows the signal flow through the SSLCA. Input spikes are handled by Row and Column Headers attached directly to a nanowire crossbar with memristors at each junction. The Row Headers allow these input spikes during encoding and set appropriate voltages to change memristor states. The Column Headers, the neurons, fire when sufficiently stimulated. Output neuron spikes pass current back through the nanowire crossbar, charging capacitors in the Row Headers. This inhibitory force prevents over-represented input spikes from reaching the crossbar for the duration of

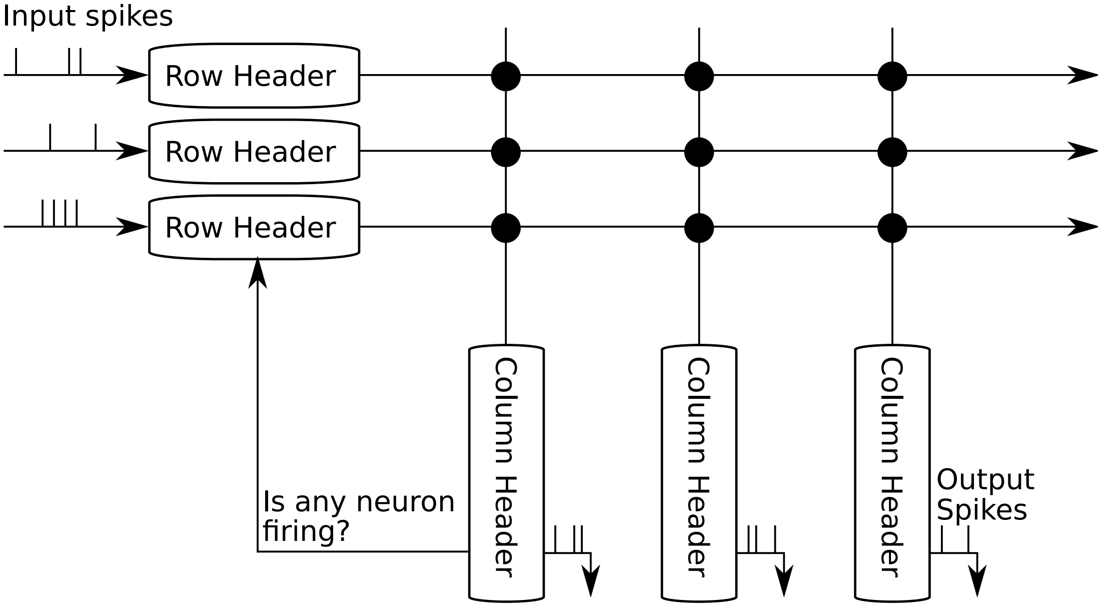

Figure 2.2: Simple Spiking implementation of the LCA, reproduced from [1], with permission. Input spikes are fed into the Row Headers. Voltage passes through the Row Headers to a nanowire crossbar with memristors at each junction. Current is used to charge and discharge neurons in each Column Header.

the output spike. The circuitry within the Row and Column Headers that exhibit this behavior can be seen in Fig. 2.3.

The SSLCA uses sparse coding optimization to minimize a weighted combination of coefficients. Sparse coding achieves low average activity levels by forcing many of the coefficients in the system to be zero through regularization and thresholding. This results in dimensionality reduction, creating a new basis that represents the original input, thus reducing the complexity and power consumption of the system. Unlike algorithms that optimize sparsity at each time step, resulting in dramatic changes to small variations in stimuli, the SSLCA creates inertia by optimizing over a time window. This inertia promotes more proportional changes to outputs when presented with smooth, time-varying changes to the input.

The SSLCA is based on the time-varying ODE derived as part of Rozell *et al.*'s work on the LCA [9], reproduced in Eq. (2.1), which accumulates an internal state

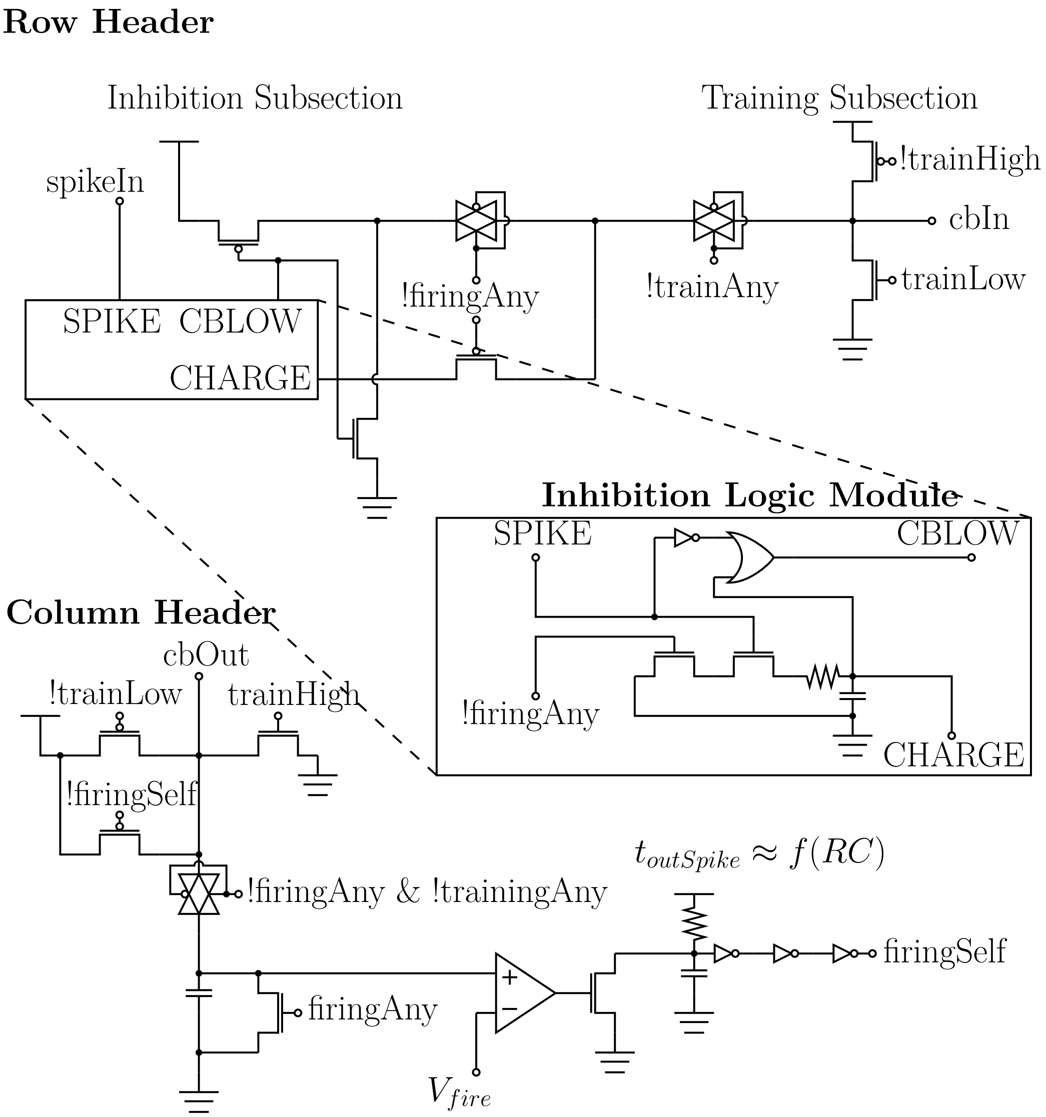

Figure 2.3: Row and Column Header circuit design to achieve inhibition in the SSLCA, reproduced from [1], with permission. The Row Headers keep input spikes from reaching the crossbar when inhibited and track the state of inhibitory forces. The CHARGE port is used for current sinking from the crossbar during output spikes. A capacitor in the Inhibition Logic Module prevents subsequent spikes from sending voltage to the crossbar. The Column Header has a transmission gate to send current to and from capacitors that represents each neuron's state. After a neuron fires, its capacitor is drained. An RC circuit with several NOT gates is used to amplify and digitize the output spikes.

that is used to fire output neurons:

$$\dot{u}_m = \frac{1}{\tau}\left[b_m(t) - u_m(t) - \sum_{n \neq m} G_{m,n} a_n(t)\right]. \tag{2.1}$$

The ODE specifies dynamics for the coefficients u_m to evolve as a dynamic system with respect to time. The vector $\boldsymbol{b}(t)$ is the inner product of the input to the system and the dictionary elements, approximated by the memristors. The vector $\boldsymbol{u}(t)$ contains the internal state variables, the charge on each column's header due to the spiking input. The vector $\boldsymbol{a}(t)$ is the set of time-varying active coefficients, a sparse approximation resulting from a soft threshold on each node's internal state. The term $G_{m,n}$ is the inhibition signal from active node m to any other node n. Finally, the term $\frac{1}{\tau}$ is a normalization factor, where τ represents the time scale. For additional theory relating to the underlying sparse coding work, refer to [9], and for more SSLCA implementation details, refer to [1].

The ODE amplifies the convex sections of the signal, increasing the probability of finding correlations between the input signal and a corresponding radionuclide template in the dictionary. Sparse coding optimization squashes the uninformative parts, reducing the significance of background noise and anomalies on predictions. These computations can be reduced to primitives that are implementable with analog devices, making hardware implementation possible. The use of memristors allow for low complexity scaling and results in a significant power reduction compared to the LCA.

2.2 Radionuclide Spectrum Template Preprocessing

The *National Nuclear Data Center* (NNDC), a part of the US Nuclear Data Program [12], collects data related to nuclear science from several national laboratories and universities. The data is evaluated, archived, and disseminated for the purpose of basic research and applied nuclear technology. The ANSI N42.34 standard [13] has identified a subset of the NNDC Nuclear Wallet Cards [14], lists of radionuclides and their associated gamma-ray energy levels, for testing RIDs due to common use in industrial, medical, NORM, and SNM categories. Many of these radionuclides may be diverted to a dirty bomb or used for other clandestine activity, while others are used to mask the presence of those used for nefarious purposes. For the purpose of threat reduction, and to avoid false alarms, the following radionuclides have been included: *Sodium-22* (^{22}Na), *Chromium-51* (^{51}Cr), *Manganese-56* (^{56}Mn), *Cobalt-57* (^{57}Co), *Iron-59* (^{59}Fe), *Cobalt-60* (^{60}Co), *Copper-64* (^{64}Cu), *Gallium-66* (^{66}Ga), *Gallium-67* (^{67}Ga), *Gallium-68* (^{68}Ga), *Selenium-75* (^{75}Se), *Strontium-86* (^{85}Sr), *Ruthenium-103* (^{103}Ru), *Indium-111* (^{111}In), *Iodine-123* (^{123}I), *Iodine-131* (^{131}I), *Barium-133* (^{133}Ba), *Cesium-137* (^{137}Cs), *Cerium-144* (^{144}Ce), *Europium-152* (^{152}Eu), *Samarium-153* (^{153}Sm), *Holmium-166* (^{166}Ho), *Ytterbium-169* (^{169}Yb), *Iridium-192* (^{192}Ir), *Thallium-201* (^{201}Tl), *Bismuth-207* (^{207}Bi), *Thorium-231* (^{231}Th), *Uranium-235* (^{235}U), *Plutonium-239* (^{239}Pu), and *Americium-241* (^{241}Am). The data for these radionuclides was scraped from the NNDC radionuclide gamma-ray tables, which is then used to build the radionuclide templates for the SSLCA dictionary.

2.2.1 Template Tuning

The truth values for the photoelectric peaks present in the NNDC scraped gamma-ray tables are idealized values based on the probabilities of photoelectric interaction by photon energy resulting from a specific radionuclide's emissions. Due to the physics of photon interaction in a gamma-ray detector, the probability of energy deposited within a detector for a given photon is dependent on the energy of the source photon. To account for this phenomenon, calibration spectra are used to approximate the energy dependent efficiency curve for the detector that will be used as input to the SSLCA. The values from the NNDC tables are then scaled according to the efficiency curve to match the probability the detector material will absorb a photon given its incident energy. An example of the raw gamma-ray table for ^{152}Eu and the same table scaled to match the efficiency of a *Sodium Iodide* (NaI) gamma-ray detector can been seen in Fig. 2.4.

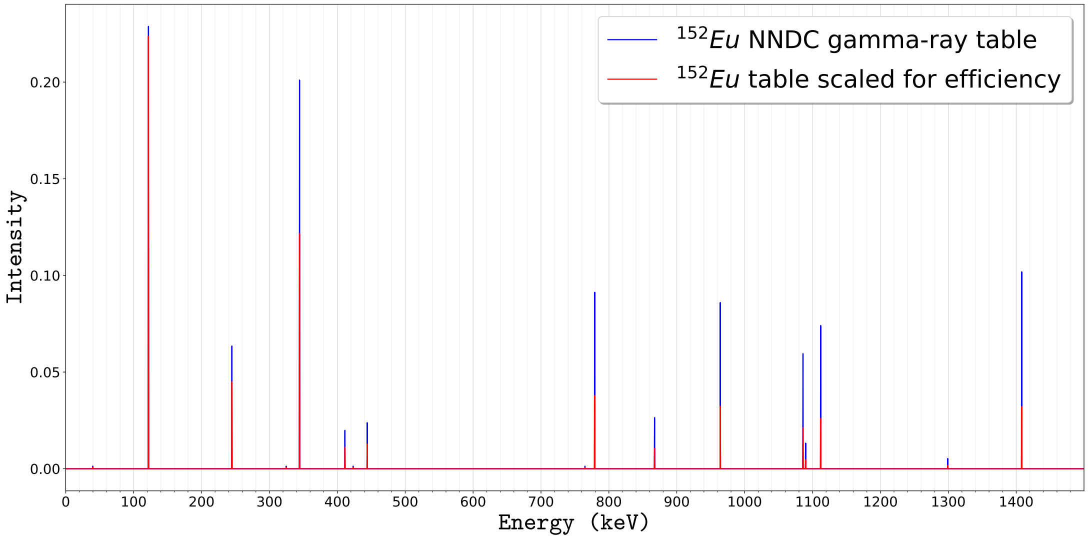

Figure 2.4: Comparison of an ideal ^{152}Eu spectrum and the same spectrum scaled to match a NaI gamma-ray detector's energy efficiency.

Another key characteristic of a gamma-ray detector's response is it's resolution. Detector resolution affects how accurately a gamma-ray at a specific energy level is binned. The NNDC tables list the photons emitted at exact energies, however, the measured photoelectric for most detectors follows closely to a Gaussian distribution with a mean at the specific energy level and a standard deviation dependent on the resolution of the detector. High resolution detectors have small variances and thus thinner, taller photopeaks. Low resolution detectors have much broader photopeaks. Similarly to detector efficiency, resolution is much better at lower keV values and decreases non-linearly at higher energy levels within the spectrum. Therefore, to match radionuclide spectrum templates to the shapes of peaks generated from measurements from a specific detector, a calibration spectrum is used to experimentally determine the *Full Width Half Max* (FWHM), the width of a photopeak at half its maximum intensity, of several photopeaks throughout the energy range of the spectrum. A non-linear fit is approximated using the centroid energy levels and their associated FWHM to determine an equation for standard deviation given an energy level. Using these parameters, energy dependent Gaussian broadening is applied to the detector efficiency scaled photopeaks from the NNDC tables. An example of the broadening effect can be seen applied to the efficiency scaled gamma-ray table for ^{152}Eu in Fig. 2.5.

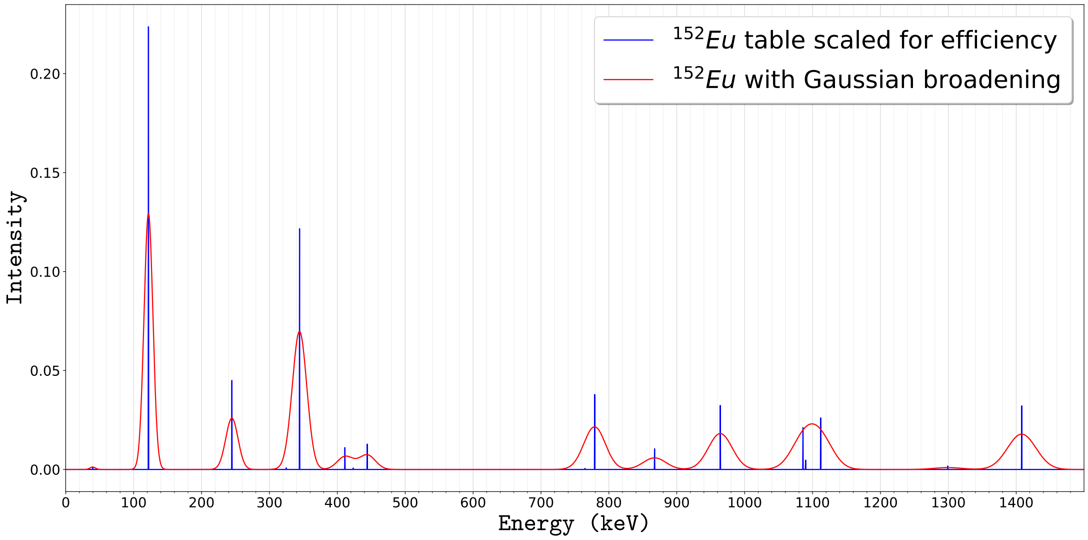

Figure 2.5: Comparison of efficiency scaled ^{152}Eu spectrum and the same spectrum after applying Gaussian broadening to match a NaI gamma-ray detector's energy resolution.

2.2.2 Compton Scatter Predictions

At low photon energies the photoelectric effect dominates. However, at higher energies, above approximately 100 keV, another form of quantum interaction begins to occur known as the Compton effect. As opposed to the photoelectric effect where all energy of an incident gamma-ray is transferred to an electron, the Compton effect is the result of a relativistic, elastic collision. Therefore, only part of the energy of the incident gamma-ray is absorbed by the particle, while the rest of the photon energy ricochets of it. This type of effect, more commonly known as Compton scatter, was discovered in 1922 by Arthur H. Compton while experimenting with scattering X-rays off light elements. A visualization of this effect can be seen in Fig. 2.6.

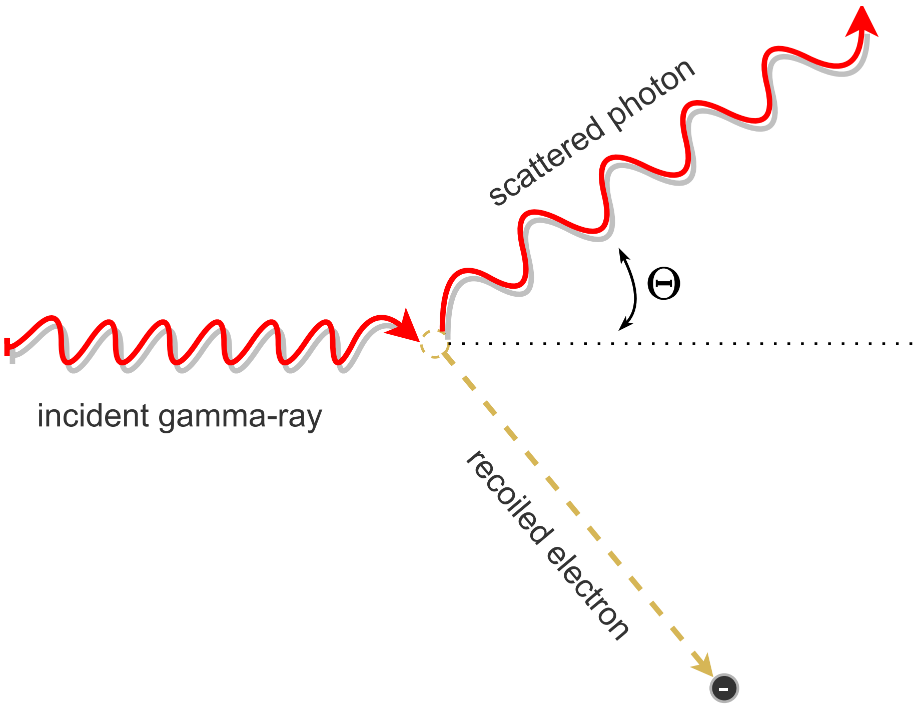

Figure 2.6: Compton effect, where an incident gamma-ray scatters off an electron splitting its energy between a recoiled electron and the scattered photon.

Below 100 keV, virtually all photon energy is deposited in the detector. This interaction probability goes roughly as Z^3/E^3, where Z is the atomic number of the material and E is the energy level of the photon, so it strongly dominates at lower gamma-ray energies but drops sharply with increasing energy. At energies above approximately 100 keV, depending on the particular material, Compton scatter dominates, going roughly as Z/E, and the incident photon scatters, depositing only part of its energy in the detector via a recoil electron. The probabilities of these interactions are modeled by the Klein-Nishina formula, Eq. (2.2), for the differential cross section of Compton scatter off a single electron:

$$\frac{d\sigma}{d\Omega} = \frac{1}{2}r_0^2\left(\frac{E'}{E}\right)^2\left(\frac{E'}{E} + \frac{E}{E'} - \sin^2(\theta)\right), \tag{2.2}$$

where r_0 is the classical radius of the electron, E is the incident photon energy and E' is the scattered photon energy, given through the angular relation in Eq. (2.3):

$$E' = \frac{(E)}{1 + \frac{E}{m_0c^2}(1 - cos\theta)}. \tag{2.3}$$

Integrating over all solid angle gives the total Compton cross section for a given incident photon energy. Multiplying by the atom density gives the interaction probability for a given detector material. This probability is distributed over a range of possible outgoing photon energies, and thus recoil electron energies which give the signal in the detector. To understand the detector energy distribution, the Klein-Nishina formula can be reformulated as a function of electron energy [15] by Eq. (2.4):

$$\frac{d\sigma}{dE_e} = 2\pi\frac{d\sigma}{d\Omega}\frac{m_0c^2}{E'^2}. \tag{2.4}$$

The total amount of Compton scatter is proportional to both the energy of the incident photon and the atomic number of the detector material. Therefore, for each photoelectric peak found in the NNDC tables, the ratio of photoelectric to Compton scatter is determined based on the Z value of the detector material and the energy of the incident photon, which is recorded in the photoelectric peak. Using the magnitude of the photoelectric peak, the total amount of Compton scatter due to the specific peak is estimated. This predicted Compton scatter is than spread out across the distribution created from Eq. (2.4), thus predicting the Compton continuum associated with a given photoelectric peak. An example of a predicted continuum for a single photopeak can be seen in Fig. 2.7.

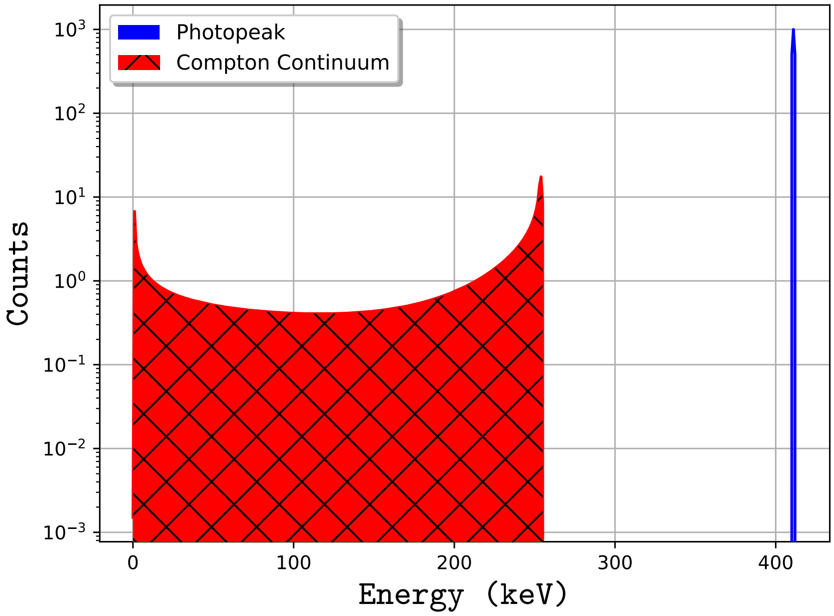

Figure 2.7: Compton continuum created from a photopeak at 411 keV. The Compton continuum is the prediction of detected energy after photons Compton scatter off electrons in the detector. These predictions are added to the template dictionary to assist in identifying radionuclides from gamma-ray spectra that have hit counts at energy levels produced due to Compton scatter.

For all radionuclides selected from the NNDC Nuclear Wallet Cards, the associated Compton continuum is approximated for each possible photoelectric peak within the given detector's efficiency range. These continua are aggregated and added to the detector efficiency-scaled photoelectric peaks prior to the Gaussian broadening. An example of a final ^{152}Eu spectrum template can be seen in Fig. 2.8.

These key features create the radionuclide templates used for the SSLCA's dictionary, transforming the theoretical properties of a radionuclide emission into a more realistic representation of the data based on a detector's properties. Thus, maximizing the likelihood for true positive identifications based on the similarity between a measured gamma-ray spectrum and a radionuclide template.

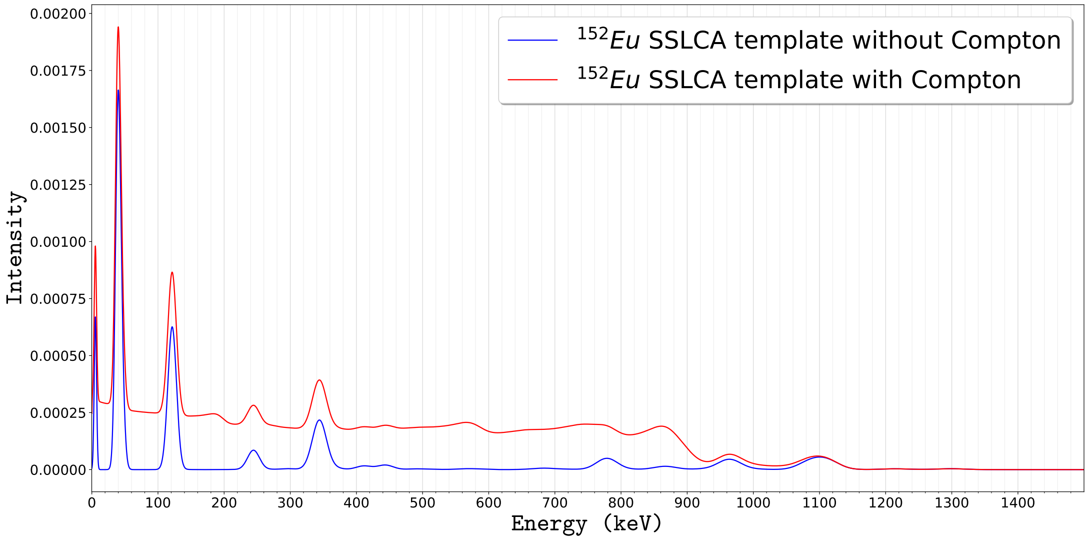

Figure 2.8: Comparison of SSLCA ^{152}Eu spectrum template with and without Compton scatter predictions.

2.3 Gamma-Ray Spectrum Test Set

An initial training set of spectra was created in the *University of New Mexico* (UNM) Nuclear Laboratory to develop and tune the algorithm. This dataset consists of sources of ^{22}Na, ^{137}Cs, ^{152}Eu, ^{57}Co, ^{60}Co, and a tri-nuclear Am-Cm-Pu source, measured with NaI, and *Cadmium Zinc Telluride* (CZT) detectors. These spectra are meant to vary in difficulty of identification, so their *Signal To Noise* (SNR) and counting time intentionally have large variations.

A larger, more balanced test set of spectra was created specifically using ^{137}Cs, ^{57}Co, and ^{152}Eu with *High Purity Germanium* (HPGe), NaI and CZT detectors to determine the effect of detector resolution and energy efficiency on the algorithm's performance. These spectra were collected with varying distances from the detector, ranging from 0 to 24 inches, and with variable counting times ranging from 10 to 300 seconds.

Additionally, uranium and plutonium SNM sources ^{235}U and ^{239}Pu were collected with HPGe, NaI, and CZT detectors along with ^{60}Co which has medical and industrial uses. The ^{235}U is in equilibrium with its decay daughter ^{231}Th. No other decay products are identified in the ^{235}U or ^{239}Pu sources. These spectra were also collected using the same timing and distance variations mentioned above for the test sets. Fig. 2.9 shows two ^{235}U spectrum from the test set. One of the spectrum was measured for 60 seconds with the source 24" from the detector, resulting in a low SNR (-13.18 dB). The other was measured for 300 seconds with the source 2" from the detector, resulting in a higher SNR (10.76 dB).

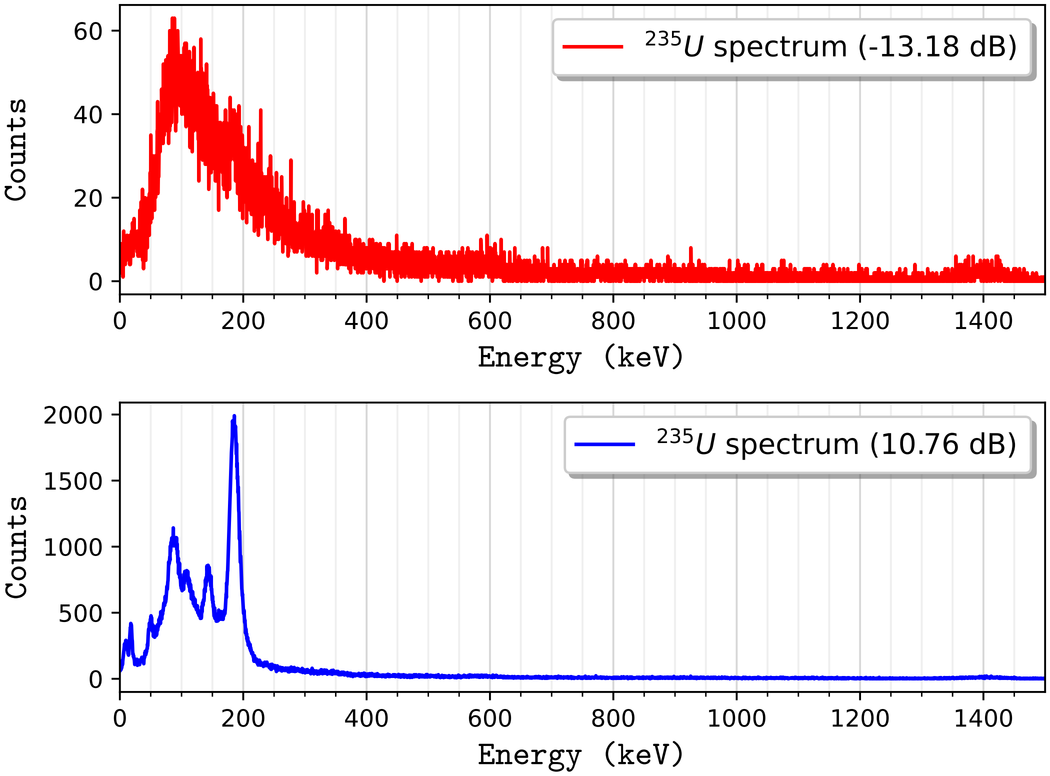

Figure 2.9: Two ^{235}U spectrum from the test set, one with a low SNR (top) and one with a higher SNR (bottom).

2.4 Power Consumption and Efficiency

Since the goal of this research is to create a power-efficient RID, it is important to determine a baseline for the power consumption of the benchmark radionuclide identification algorithm. To quantify the power consumption of an algorithm the algorithm's CPU usage by the processor's thermal design power [16] must be determined. Using Microsoft Visual Studio Performance Profiling Tools [17], it is possible to see the percent of CPU usage during the wall-clock time of all the processes associated with an application, which only accounts for the time a specific process spends in the CPU and not all other concurrent tasks that are moved in and out of the processor during the total elapsed time.

With the statistics provided by the profiling tool, I calculated the area under the curve for CPU usage by time for all processes specifically related to the radiation identification algorithm. This results in an approximation of the total CPU usage for the complete identification process. Since the tool samples the CPU usage in discrete time slices, calculating the slope of the slice and using the y-intercept at its start, it is possible to determine the approximate CPU usage of the interval by integrating over it with respect to time. The summation of each of these areas, Eq. (2.5), determines the approximate CPU usage of the total process:

$$CPU_{usage} = \sum^{timeslices} \int_0^{\Delta t} \left(\frac{\Delta y}{\Delta t} + y_{int} \right) dt. \tag{2.5}$$

Multiplying the total CPU usage by the processor's thermal design power [16] results in an approximation of the total power used by the process. Determining the total processing time of the identification process in seconds and dividing that out of the total power used determines the power consumption of the algorithm by unit time or watts. Using the results of profiling GADRAS, I determined the average power consumption of its radionuclide identification algorithm to be ~ 1.76 W.

The SSLCA architecture was chosen as a low-power alternative for radionuclide identification. However, since the SSLCA is intended to be a hardware-based algorithm, a different methodology is required to evaluate its power consumption. Once built, the actual power consumption can be directly measured, but for simulation purposes the power consumption can be estimated based on the hardware required for the system. Assuming 8,192 inputs, the maximum number of channels produced by the MCA, and 30 outputs, the size of the radionuclide dictionary, the system produces an average activity of 1% duty cycle per second. That allows for up to 100 decays detected per channel per second. With an RF average of 0.4 W, 53 kΩ resistance memristive devices, and a 2.2 μW CMOS for headers, the total power consumption of the SSLCA is estimated to be approximately 8 mW. Thus, the SSLCA is $\sim 99.5\%$ more power-efficient than the GADRAS radionuclide identification algorithm.

In addition to power efficiency, the SSLCA also has the advantage of time efficiency. It is capable of real-time identification as decays are detected. The spiking signal moves through the circuit at a much faster rate than the processing needed for a traditional software approach that requires large numbers of mathematical operations.

2.5 Radionuclide Identification Experiments

The input to the SSCLA are the histogrammed pulses from a gamma-ray detector, as depicted in Fig. 2.1. The raw counts by channel are converted to keV bins using the detector's calibration values, aligning the data with the templates. Counts below 50 keV are discarded, since they contain a significant amount of detector noise and background. The signal is normalized by acquisition time, resulting in counts per second as seen in Fig. 2.10a for a ^{239}Pu spectrum with a high level of background. The background is then subtracted using an average of time normalized spectra from the location of the gamma-ray detector when no active source is present. Finally, the resulting background subtracted spectrum is normalized by its magnitude, which is the square root of the sum of squares of each channel's count, resulting in a unit vector. The final pre-processed version of a ^{239}Pu spectrum can be seen in Fig. 2.10b.

The cleaned and normalized data from the spectrum is linearly decomposed into its constituent parts. It is then reconstructed using sparse coding optimization based on each of the templates in the dictionary. The SSLCA determines a confidence in its prediction based on the similarity between the reconstructions and their associated template. A threshold is used to determine the top radionuclide candidates, or if there is none present in the signal. A visualization of the process and results can be seen in Fig. 2.10. Fig. 2.11 shows the results of the same process on a ^{235}U spectrum with its decay daughter ^{231}Th.

2.5.1 Special Nuclear Material Masking

For the purpose of threat reduction it is important that any radionuclide identification algorithm can detect SNM when masked with medical, industrial or NORM sources. To determine the SSLCA's ability to identify SNM material in a masking configuration, an additional test set was generated for both the high resolution HPGe detector and the low resolution NaI. Noisy ^{235}U and ^{239}Pu spectra were chosen from the original test set. *Monte Carlo N-Particle* (MCNP) [18] simulations of the industrial radionuclide ^{57}Co and NORM ^{152}Eu was generated for each detector. For each of the SNM spectra, background is subtracted to determine the number of counts due to the SNM radionuclide. The mask radionuclide is then scaled in relation to the background

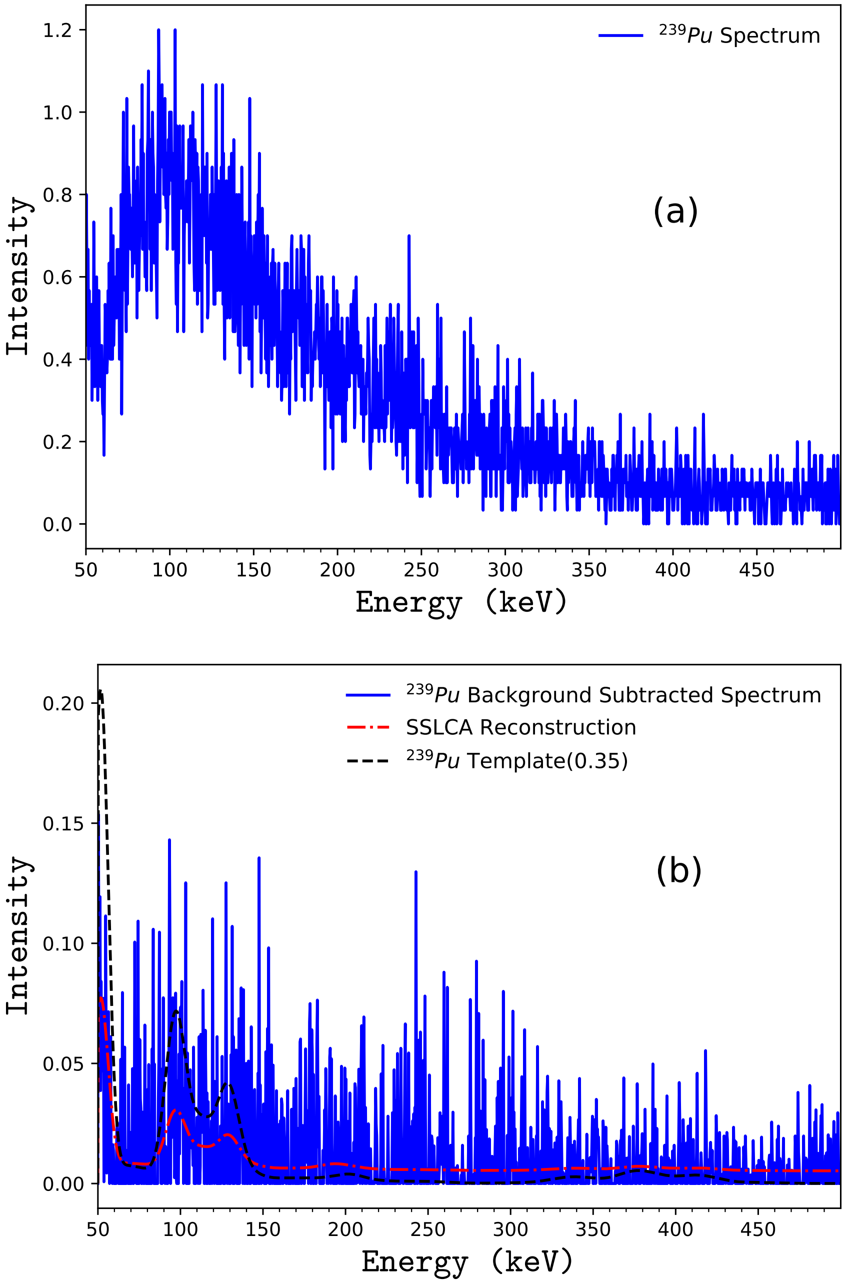

Figure 2.10: Visualization of the SSLCA radionuclide identification on a low SNR ^{239}Pu gamma-ray spectrum, resulting from a 30 second collection time using the NaI detector, with a source activity level of 1.1×10^5 Bq. (a) The time normalized spectrum with 586 counts per second, 498 from background, and 88 from photoelectric. (b) The same spectrum after background subtraction and normalization, shown along with the SSLCA reconstruction using the correctly predicted ^{239}Pu template, resulting in a 35% similarity between the two.

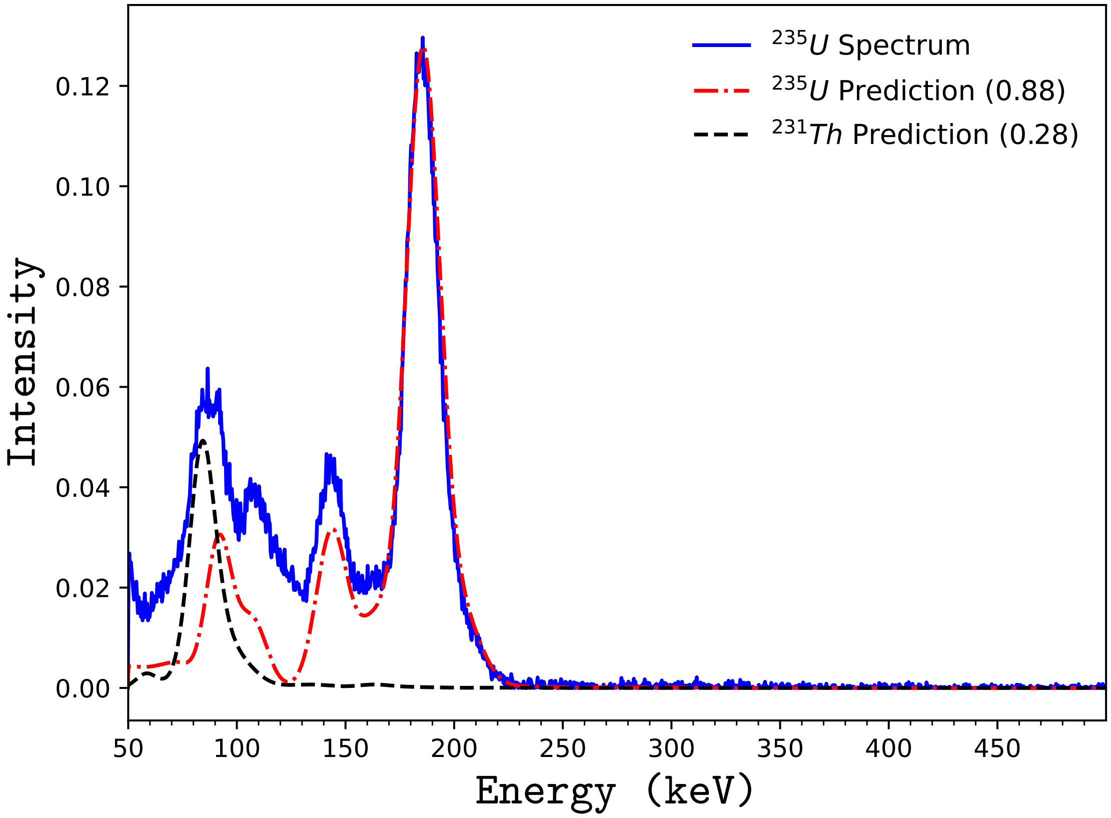

Figure 2.11: Visualization of the SSLCA radionuclide identification algorithm. The solid line is a gamma-ray spectrum collected from a ^{235}U radionuclide with its ^{231}Th daughter using the NaI detector. The dash-dot line is the reconstruction of the spectrum using the ^{235}U template, resulting in an 88% similarity between the two. The dash line is the reconstruction of the spectrum using the ^{231}Th template, resulting in an 28% similarity between the two.

subtracted SNM spectrum and then added to the original, noisy SNM spectrum. For each of the SNM radionuclides and each of the masking radionuclides, spectra were created from 10% mask to SNM up to 10X mask to SNM in 10% increments. This allows for determining the tolerance of mask to SNM ratio at which the SSLCA can no longer identify the SNM in each of the masked configurations tested. An example of a generated SNM masking configuration can be seen in Fig. 2.12 with a ratio of 1 between the mask, ^{152}Eu, and the SNM, ^{239}Pu, signifying an equal number of counts from both radionuclides.

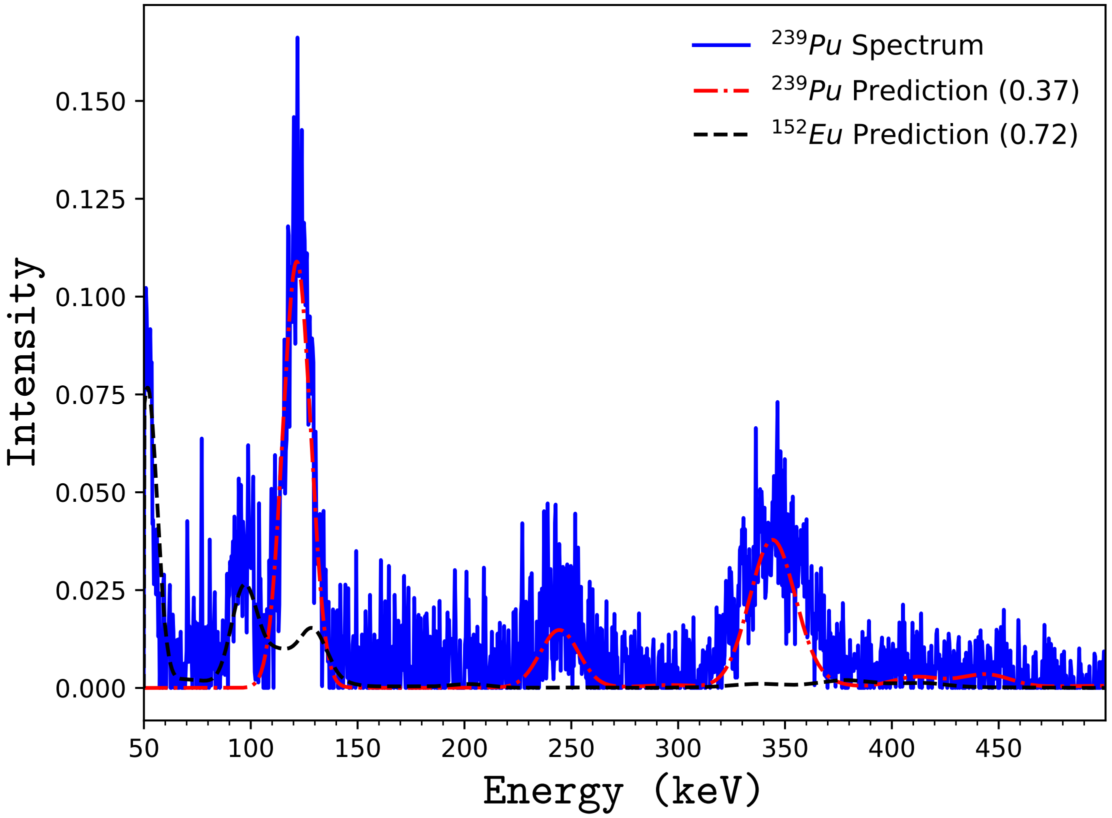

Figure 2.12: Visualization of the SSLCA radionuclide identification algorithm on a spectrum composed of SNM ^{239}Pu masked with ^{152}Eu. This spectrum is the result of the addition of an MCNP simulated ^{152}Eu spectrum and a real ^{239}Pu spectrum collected using the NaI detector with a ratio of 1 between the counts of the two radionuclides.

2.6 Radionuclide Identification Results

Experiments by detector type were performed on the gamma-ray spectra test sets resulting from measurements of ^{57}Co, ^{137}Cs, ^{152}Eu, ^{60}Co, ^{239}Pu, and ^{235}U radionuclides. These spectra were generated in a UNM laboratory representing a large range of signal to noise ratios by varying collection durations, distances, and source activities. The gamma-ray spectra were collected with three different detector types: HPGe, NaI and CZT. Identifications were performed with and without the Compton scatter predictions added to the templates in the SSLCA's dictionary. The accuracy is determined by the fraction of correct, rank-1 identifications of a radionuclide in a test set of spectra.

For comparison, the test sets for all three detectors were run through the GADRAS-DRF single regression analysis tool [19]. To make the comparisons fair, the same ^{152}Eu spectra that were used to calibrate the SSLCA were also used to tune the detector response functions in GADRAS. The weight range lower and upper limits were set to 50 keV and 1450 keV respectively so that the GADRAS analysis tool fits the same region of a spectrum that the SSLCA analyzes. The same 30 radionuclides were used in the library for both GADRAS and SSLCA analysis on all spectra. For each spectrum analyzed, the same calibration values were used for each of the two algorithms, along with the same background spectrum used for background subtraction.

2.6.1 Results by Detector Type

Table 2.1: Numerical comparison of radionuclide identification accuracy results between SSLCA and GADRAS by detector type on measured sources. The SSLCA accuracy for each detector is reported both with and without the predicted Compton scatter.

Detector Type	Accuracy		
	GADRAS	SSLCA	SSLCA Compton
HPGe	92.64%	80.88%	91.18%
NaI	90.57%	69.81%	88.68%
CZT	78.26%	66.67%	76.20 %

Table 2.1 and Fig. 2.13 compare the accuracy results for the radionuclide identification task between GADRAS, the SSLCA without the Compton scatter predictions, and the SSLCA with the Compton scatter predictions. Adding the Compton scatter predictions to the radionuclide templates increased the identification accuracy by an

average of 12.9% across the three detectors. The performance of GADRAS compared to the SSLCA using the Compton scatter predictions is approximately 1.8% better on average, thus showing that the SSLCA is capable of near SOTA results for this task.

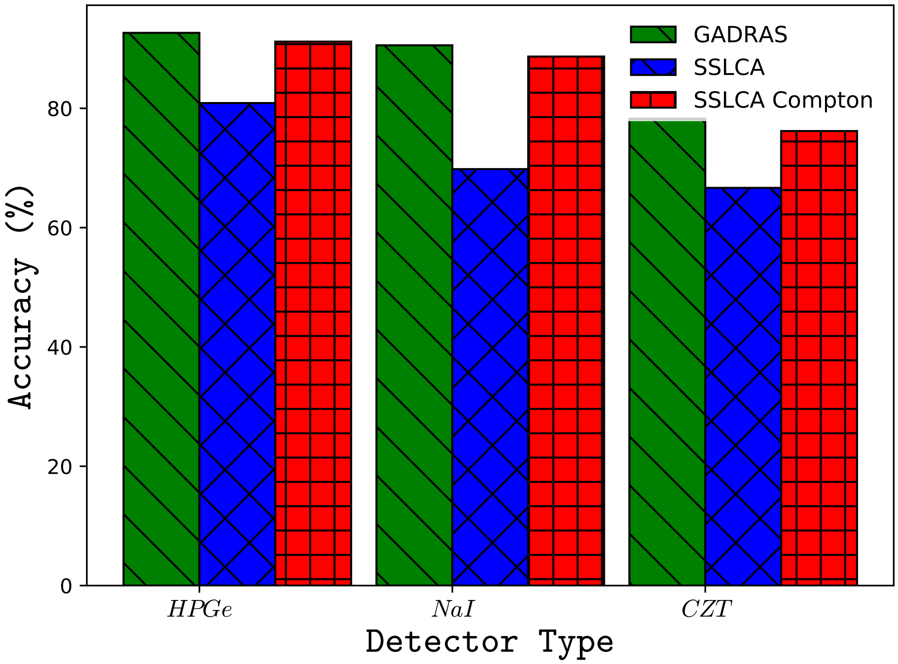

Figure 2.13: Visual comparison of radionuclide identification accuracy results between SSLCA and GADRAS by detector type on measured sources. The SSLCA accuracy for each detector is reported both with and without the predicted Compton scatter.

The CZT detector performed the worst, achieving an accuracy of only 76.2%. Due to its poor efficiency very little energy above approximately 800 keV is absorbed by the detector, making identifications of radionuclides that primarily consist of higher energy photoelectric difficult. The high resolution HPGe detector performed the best, resulting in an accuracy of 91.18%. However, its germanium crystal needs to be maintained at cryogenic temperatures and must be cooled either electrically or with liquid nitrogen, which makes it a poor choice for a low-power, portable device. The NaI and CZT detectors have no such requirement, however, despite the CZT having a higher energy resolution than the NaI, its low efficiency would only make it viable for identifying radionuclides with strong photopeaks at lower energy levels. The NaI detector, which achieved an 88.68% accuracy, has the best energy efficiency of the three detectors, and despite having the worst energy resolution of the three, it only performed 2.5% worse than the HPGe, but performed 12.48% better than the CZT. Due to its relatively high accuracy and no cooling requirements, the NaI detector seems to be the optimal choice for a low-power RID.

2.6.2 Results by Radionuclide

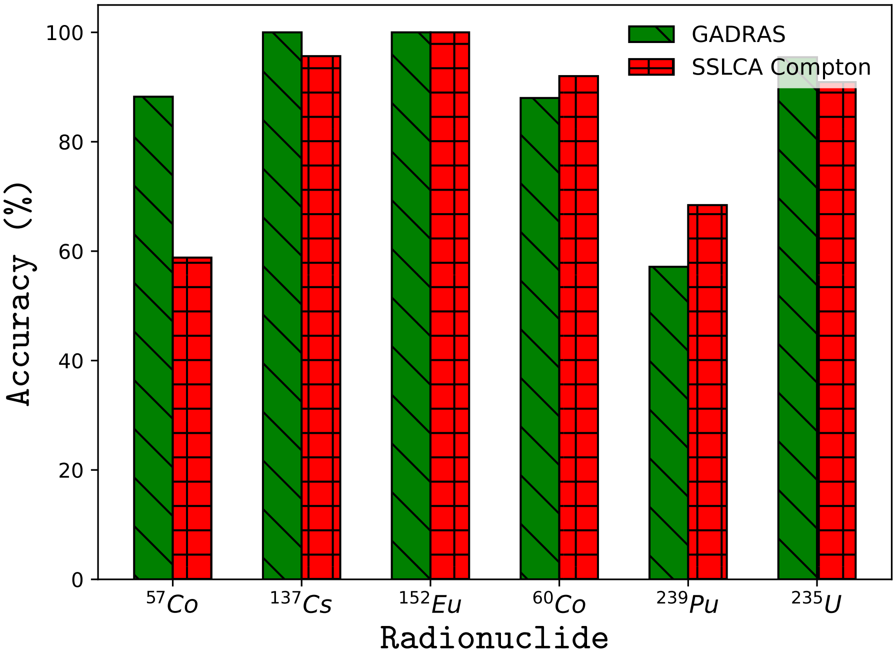

Figure 2.14: Comparison of radionuclide identification accuracy results by radionuclide between GADRAS and the SSLCA with predicted Compton scatter.

Fig. 2.14 compares the radionuclide identification accuracy results by radionuclide across the three detectors between GADRAS and the SSLCA using the Compton scatter predictions. Despite all of the detectors having a slightly lower overall accuracy as compared to GADRAS, for three of the six radionuclides tested the SSLCA performs as well or better. For ^{137}Cs and ^{235}U the SSLCA only performed a few percentage points lower. ^{57}Co is the only radionuclide that the SSLCA performed significantly worse on, with a nearly 30% lower identification accuracy. ^{57}Co shares its one and only significant photopeak around 122 keV with ^{152}Eu, making ^{57}Co difficult to differentiate from ^{152}Eu. However, ^{152}Eu has many strong photoelectric peaks throughout the energy range making it one of the most recognizable radionuclides.

2.6.3 Signal to Noise Ratio Bounds

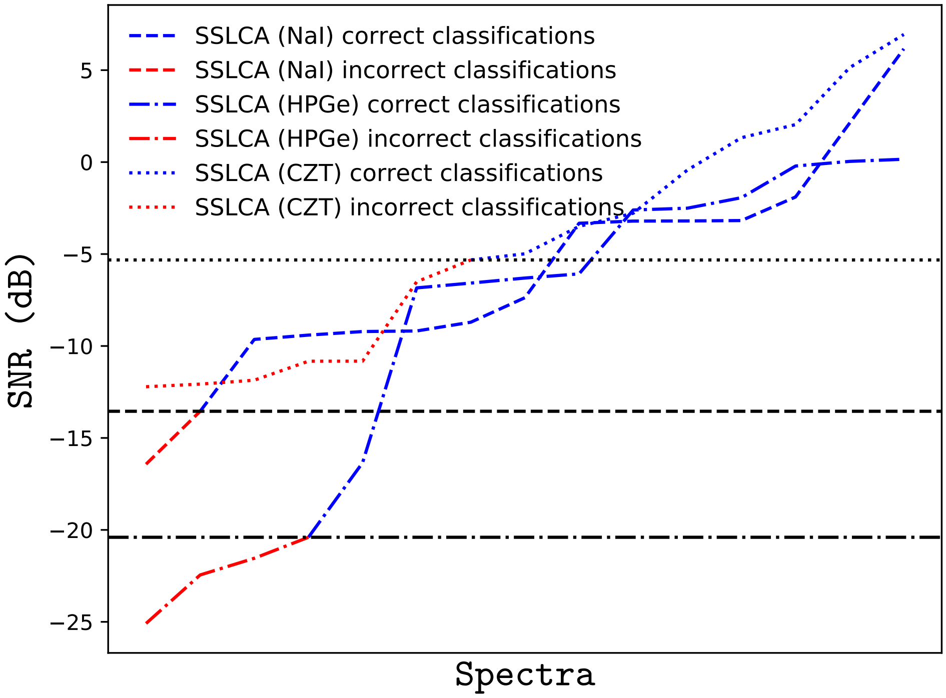

Figure 2.15: Lower bounds in dashed line, dash-dotted line, and dotted line of SNRs for the spectra in the NaI, HPGe, and CZT test sets, respectively, with correct classifications above the corresponding styled horizontal line (blue) and incorrect below (red). This shows how sensitive the SSLCA predictions are in relation to noise for each detector type.

The SNR for each spectrum in each test set was calculated, where the signal is determined to be the sum of counts for the time normalized background subtracted spectrum and the noise is the sum of the time normalized background itself. The signal and noise summations do not include energy levels below 50 keV, since the SSLCA does not use channels containing those energy levels in its analysis, therefore most inherent detector noise is not part of either term. A plot of SNRs for the noisiest spectra in relation to their correct or incorrect classification can be seen in Fig. 2.15, showing the lower bound of signal strength to background that the SSLCA tolerated and still correctly classified radionuclides in the test sets.

The gamma-ray spectra measured with the HPGe detector resulted in the best performance, correctly identifying radionuclides in gamma-ray spectra with SNRs below -20 dB. Spectra measured with the CZT detector performed the worst, with a lower bound of correct identifications around -5 dB. Using the spectra measured with the NaI detector, the SSLCA had an SNR tolerance for correctly identifying radionuclides around -14 dB. Despite slightly worse results than the HPGe, -14 dB SNR results from nearly 25X more background than source in a signal, thus the SSLCA is still capable of performing well on very noisy spectra measured with the NaI detector.

2.6.4 Special Nuclear Material Masking Results

Fig. 2.16 compares identification of HPGe and NaI measured SNM uranium and plutonium spectra masked by ^{152}Eu and ^{57}Co. The masking radionuclide is added in 10% increments to the SNM to determine the threshold at which the SSLCA can correctly identify just the SNM, the SNM and mask, and at what ratio the mask has successfully obscured the identification of the SNM.

The HPGe preformed the best for all masking configurations, identifying the SNM ^{235}U when masked by nearly 10X its intensity with ^{152}Eu, and ^{239}Pu with nearly 9X the same mask. The NaI performed nearly as well on the ^{235}U ^{152}Eu masking

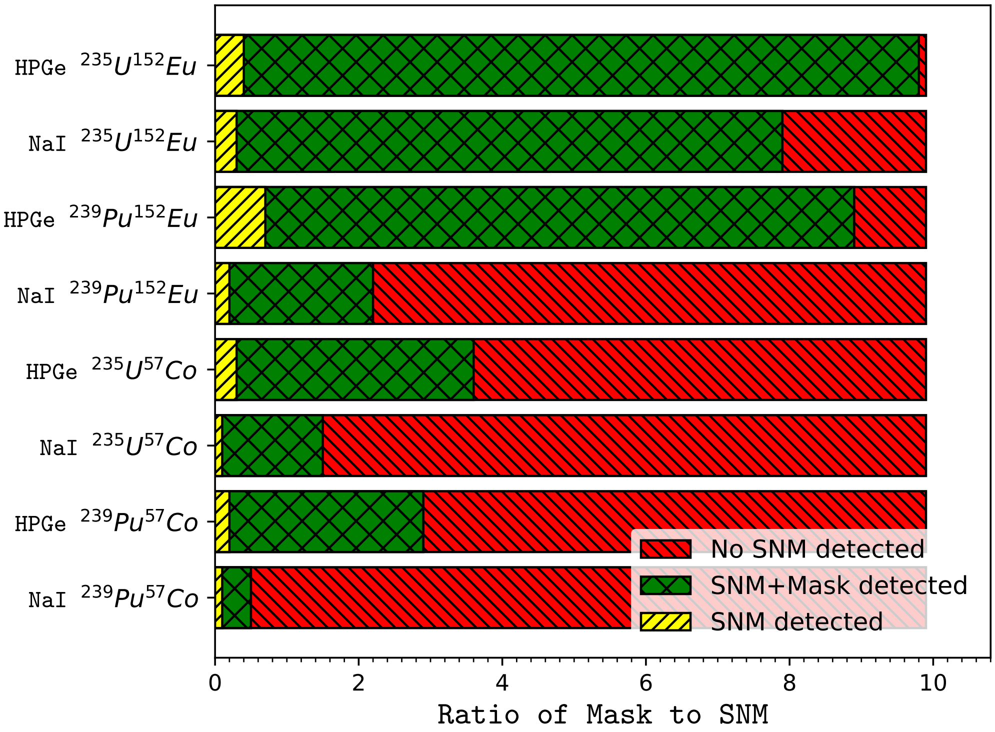

Figure 2.16: Results of the SSLCA identification on masked SNM. For each masking configuration a range of ratio of mask to SNM is shown representing the ratio of mask counts to SNM counts. The 'x' hatched (green) area is the ideal range where both the SNM and mask radionuclides are properly identified. The '/' hatched (yellow) area is where just the SNM material is identified, while not ideal it is still good for threat reduction. The '\' hatched (red) area is where the masking radionuclide has successfully masked the SNM material, thus allowing for a potential threat to go undetected.

configuration, identifying the SNM when masked by nearly 8X its intensity. However, it was only able to identify the SNM ^{239}Pu when masked by slightly over 2X its intensity with ^{152}Eu due to some overlap of photoelectric energy between the two radionuclides caused by the NaI detector's lower energy resolution.

Both detectors performed notably worse on the ^{57}Co masking configurations due to the masking agent having only one prominent photoelectric peak while both SNM have many strong peaks. Because of this, when mixing the SNM and mask, an imblance in magnitude was created between the ^{57}Co photopeak and SNM photopeaks since the SNM photoelectric is spread out across a broader range of energies. Using the HPGe, the SSLCA was still able to correctly identify ^{235}U when masked by nearly 4X the amount of ^{57}Co, and ^{239}Pu when masked by nearly 3X of it. Due to the imbalance in the radionuclides' photopeak magnitudes and the overlap of photoelectric energy due to the NaI's poor energy resolution, it was only able to correctly identify the ^{235}U SNM when masked by approximately 1.7X the amount of ^{57}Co, and ^{239}Pu when masked by 0.5X of the same radionuclide.

Chapter 3

Gamma-ray Spectrum Denoising

3.1 Denoising with Convolutional Neural Networks Overview

One of the most common signal processing techniques is known as the convolution operation. A convolution operation centers a kernel on a datum element and performs a dot product with the kernel coefficients and the overlapping, adjacent data. This is performed as the kernel is slid over the signal at some step known as the stride. The resulting output is the correlation between the kernel and the signal at each step. This operation has applications from signal smoothing to finding features within a signal. By chaining multiple convolution operations, more complex filtering or abstract feature extraction is possible.

In 2008 Jain *et al.* [20] found that simple *Convolutional Neural Networks* (CNNs) provided comparable results to *Markov Random Fields* (MRF) [21], a spatial application of a Markov process which has been successfully used for image processing and denoising. However, MRF is a non-convex optimization algorithm which requires finely tuning a ρ-function to approximate convex optimization and avoid poor local minimum using a process called graduated non-convexity. While *Neural Networks* (NNs) also have non-convex loss landscapes due to stacks of layers with non-linearities and a large parameter space, they use the optimization process stochastic gradient descent that works well for both convex and non-convex optimization problems. Well tuned NNs trained using stochastic gradient descent tend towards falling into similarly performing local minima that are preferable to the theoretical global minimum that would most likely lead to overfitting [22]. CNN, unlike traditional NNs, use shared parameters since the learned coefficients are the real values of a kernel matrix, which not only makes them more time and space efficient, but also helps regularize large models by reducing the number of parameters in deep networks.

In 2016 Zhang *et al.* [23] demonstrated SOTA results for image denoising with CNNs using a technique they referred to as residual learning. Residual learning refers to using a model to estimate the noise in the input signal and then removing it from the input during post-processing. For training and test purposes they used *Additive White Gaussian Noise* (AWGN), a process that simulates many naturally occurring random processes. There are two key elements to their model which helped it perform exceptionally well for this task. First, other than the first and last, each convolution layer is followed by a batch normalization process which contains learnable parameters that shift the output to have a mean of zero and unit variance, increasing the model's capability to learn Gaussian distributed data. Second, their best model consists of nineteen layers, which not only extends model capacity and expressiveness due to a sufficient number of parameters and non-linearties, but also increases the model's receptive field, allowing each element of the network output to be informed by a larger area of the input signal.

CNNs for image processing use 2-D convolution operations since they consist of two or three dimensional data, depending on if the images are color or grayscale. A gamma-spectrum, while visualized in 2-D space, is more appropriately handled as 1-D data, a list of intensity values with the energy levels implicitly defined. Since both the convolution operation and the Gaussian distribution are defined for 1-D space, CNNs are also a natural choice for signal processing 1-D spatial data.

3.2 The Model

For this research I have created a variant of Zhang *et al.*'s CNN model which I will refer to as the *Gamma-Spectrum Denoising Convolutional Neural Network* (GS-DnCNN), with the main differentials being the use of 1-D kernels and dilated convolutions.

3.2.1 Dilated Convolutions

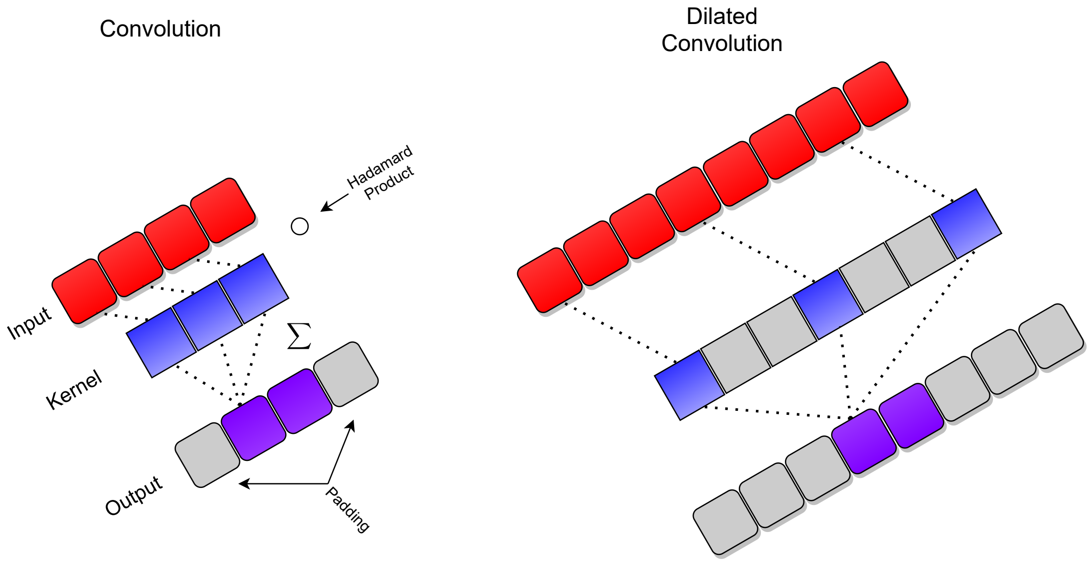

Figure 3.1: Comparison of convolution operation with no dilation and dilated convolution operation with a dilation rate of three, both with a kernel size of three. The output features are padded to keep the dimensionality of the input and output the same.

Traditional convolution operations are considered to have a dilation rate of one, meaning that each adjacent element of the kernel is applied to each adjacent element of the signal. Therefore, the receptive field of each output element is the size of the kernel itself, since that is number of elements of the signal that are required to calculate a single output. Dilated convolutions have what are known as holes in the kernel, zero padded spaces that when applied to the signal do not contribute to the output. For instance, with a dilation rate of three, every third element of the signal contributes to the output of the operation. This increases the receptive field which can be calculated using Eq. (3.1):

$$ReceptiveField = (KernelSize - 1) \times DilationRate + 1, \tag{3.1}$$

thus, with a dilation rate of three and a kernel size of three, the receptive field would be seven. A comparison between a traditional convolution operation and one with a dilation rate of three, both with a kernel size of three at a stride of one, can be seen in Fig. 3.1. The receptive field equation can be generalized for fully CNNs with $\mathcal{L}$ layers using Eq. (3.2):

$$ReceptiveField = \sum_{\ell=1}^{\mathcal{L}} \Bigg[(KernelSize_\ell - 1) \times DilationRate_\ell \Bigg] + 1. \tag{3.2}$$

3.2.2 Gamma-Spectrum Denoising Convolutional Neural Network

The GS-DnCNN is a fully CNN. The first layer consists of a single convolution layer with 16 1×3 kernels applied to the input signal at a stride of one. The layer contains no bias term and the output is passed through the non-linearity activation function *Leaky Rectified Linear Units* (LReLU) [24] defined in Eq. (3.5). This layer is used to encode the input signal into a new feature space. Following the encoder layer are some number of denoising blocks. Each denoising block consists of a dilated convolution layer with 16 1×3 kernels using a dilation rate of three and a stride of one, also with no bias term. The output of the dilated convolution layer is passed through a batch normalization layer with a LReLU activation for its output. The final layer consists of a non-dilated convolution layer with a single 1×3 kernel, no bias, and a linear output, used for decoding the denoised features back to the gamma-ray spectrum feature space. All convolution layers produce what is known as *same-padding*, the minimal number of zero-padded outputs along the edges such that the input dimensionality is the same as the output dimensionality. This is required so that the output reconstruction exists in the same feature space as the input signal.

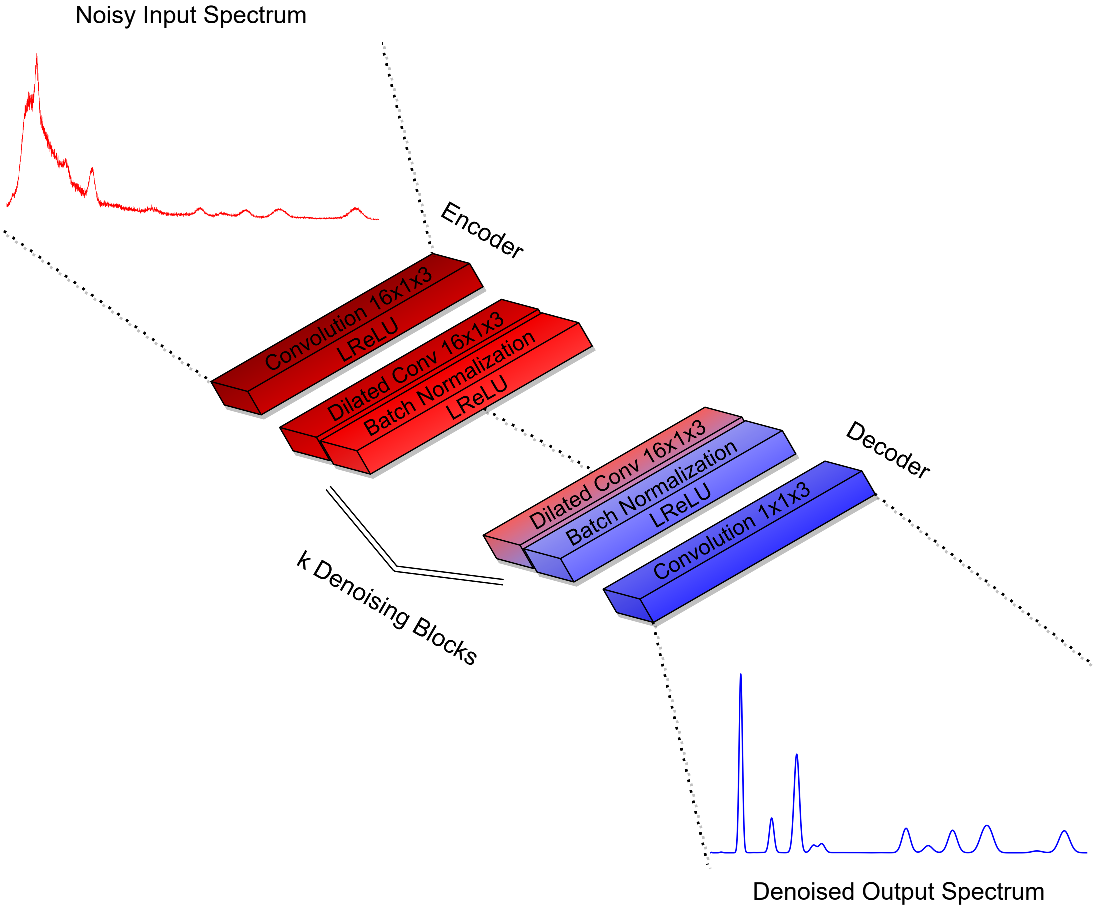

Figure 3.2: Model architecture for the GS-DnCNN. The model's input is a noisy gamma-ray spectrum. After the spectrum is encoded into a new feature space using a single convolution layer, the signal is passed through k denoising blocks consisting of a dilated convolution layer and a batch normalization layer. The denoised signal is converted back to the gamma-ray spectrum feature space with a single convolution layer resulting in a denoised version of the input spectrum. The encoder layer and all denoising blocks use Leaky ReLU activations while the output of the decoder layer is linear.

The input to the model is a noisy gamma-ray spectrum and the output is a denoised version of the input as seen in Fig. 3.2. In total the baseline model has 18 denoising blocks and thus 20 convolution layers, the maximal number before the gradient descent algorithm begins to produce vanishing gradients near the input layer due to the multiplicative property of the chain rule used for propagating small errors backward through the network [25]. Using Eq. (3.2) with the above described parameters, 20 layers, 2 with a dilation rate of 1 and 18 with a dilation rate of 3, all with a filter size of 3, the receptive field for the model is 113 input features for each output element. Therefore, this model uses a neighborhood of 113 channels of a gamma-ray spectrum to determine the amount of noise reduction required for each channel of the spectrum.

3.3 Training Data Preparation

Training a denoising model requires building a dataset of clean and noisy input vectors. Since the function of this model is to denoise gamma-ray spectrum for the purpose of radionuclide identification, a select set of radionuclide gamma-ray spectrum templates is required that is representative of the desired output for the task. Additionally, a set of realistic background noise spectra is required to mix with the templates for the noisy training vectors. Using the noisy mixes as input vectors and the clean templates as the target for an objective loss function, the model can determine an error between a denoised output spectrum and the desired output, a noiseless version of the noisy input. Using the gradients of these errors the model is able to move in the opposite direction of the gradient, thus minimizing the objective function. This therefore results in the model learning to differentiate between photoelectric peaks from the clean spectra and the additive noise.

3.3.1 MCNP Radionuclide Simulations

Since a gamma-ray detector is a measurement device which will always result in some about of noise and error, real gamma-ray spectra are not desirable for creating perfectly clean theoretical spectra. However, since different types of detectors have different characteristics, to train a network to model and learn the distribution of photoelectric generated by the detector that will be used as its input, a close approximation of these spectra is required. Therefore, MCNP is used to simulate the properties of the material for a desired detector type for the purpose of measuring photoelectric energy, generating spectra for a variety radionuclides with a variety of photoelectric peak distributions.

The MCNP is a particle simulator that models the physical interactions of the described material, therefore it not only generates photoelectric but also the Compton effect based on the material type and size for the gamma-ray detector that is being simulated. However, since Compton scatter is not representative of the true energy that is being emitted from a radioactive source, for the purpose of this algorithm it is considered noise. This allows the model to generate spectra that are closer to a radionuclide's theoretical spectrum, thus making identification more feasible. Therefore, the first step in generating clean target spectra is to remove all Compton scatter present in the simulations. Using the energy levels described in the NNDC gamma-ray tables, the photoelectric energy for the specific simulated radionuclide can be identified, thus all remaining energy is considered Compton as seen in Fig. 3.3.

After the two types of energy are separated, Gaussian broadening is applied to each signal separately, as seen in Fig. 3.4, to model the energy resolution of the type of gamma-ray detector that will be used as input to the algorithm. This allows for using the photoelectric to create clean spectra and the Compton can later be added

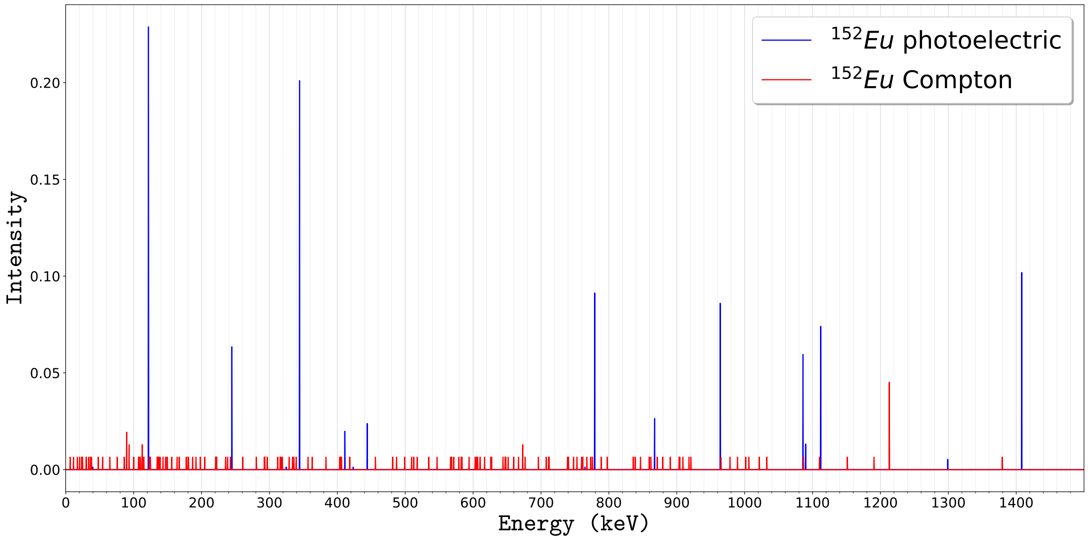

Figure 3.3: Normalized MCNP simulation of a ^{152}Eu spectrum with 100 total counts for a 3” × 3” NaI gamma-ray detector. Since MCNP simulates physical interactions, both photoelectric and Compton scatter are generated. To create clean target spectra that only contain photoelectric the two must be identified and separated.

back as noise when generating the noisy input spectra.

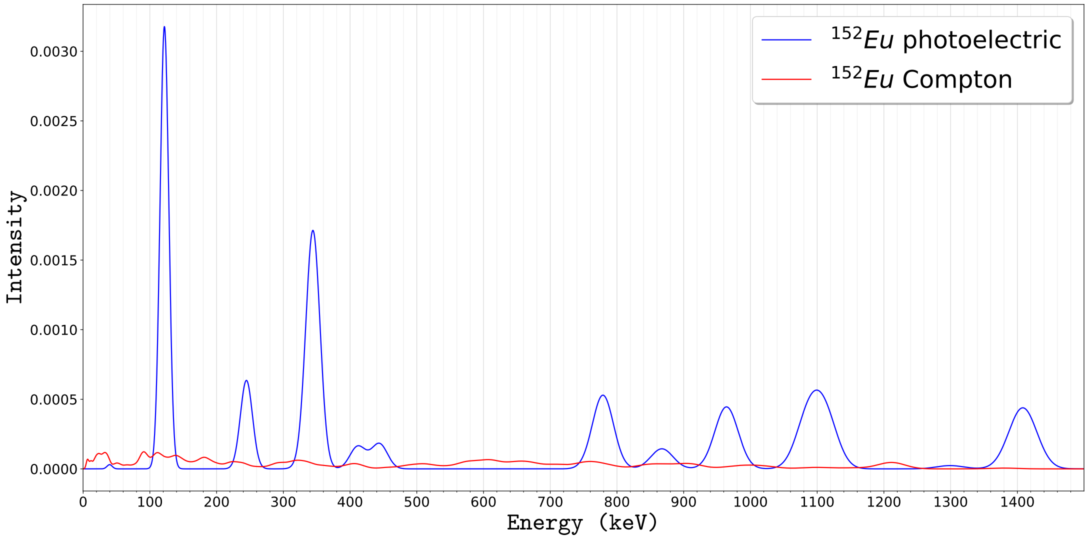

Figure 3.4: Separated Compton and Photoelectric from MCNP simulation after applying Gaussian broadening. The photoelectric forms the clean target spectra while the Compton is added back as noise for the training vectors.

This preprocessing step of seperating and broadening photoelectric and Compton is done for ^{57}Co, ^{60}Co, ^{137}Cs, ^{152}Eu, ^{235}U, and ^{239}Pu radionuclide simulations with 10, 100, 1 thousand, 10 thousand, 100 thousand, and 1 million total counts. These radionuclides consist of a variety of photoelectric distributions from only low, medium, or high, to a broad range. Since photon emissions and Compton scatter are modeled by a Poisson distribution, the larger the intensity of the source, the closer it will model a theoretical spectrum for a radionuclide. However, at lower counts, there will be variations in the ratios between photoelectric peaks and the location of Compton scatter, thus using simulations with varying number of counts is more representative of measured gamma-ray spectrum with different source activity levels.

3.3.2 Noisy Mixtures

For background and noise, a variety of measurements were taken at the UNM labs between different source measurements for different durations. Taking the background measurements between source measurements results in small amounts of different sources being present in the spectra other than just those present in the construction materials of the lab. Measuring for different durations increases the the variance between each measurement due to the probabilistic nature of photoelectric and Compton scatter interactions. Using measurements from the detector type that will be used not only models the expected distribution of real background but also includes any inherit noise and error that exists for the detector.

To generate the noisy input spectra for the training and validation set, a cleaned simulation has its Compton scatter and a background added to it at some SNR, as seen for a ^{152}Eu simulation at -15 dB in Fig. 3.5. This is done for each of the simulations and each of the backgrounds at SNRS from -25 dB, where the signal is mostly obfuscated by the noise, to 50 DB, where almost no noise is present, in 5 dB

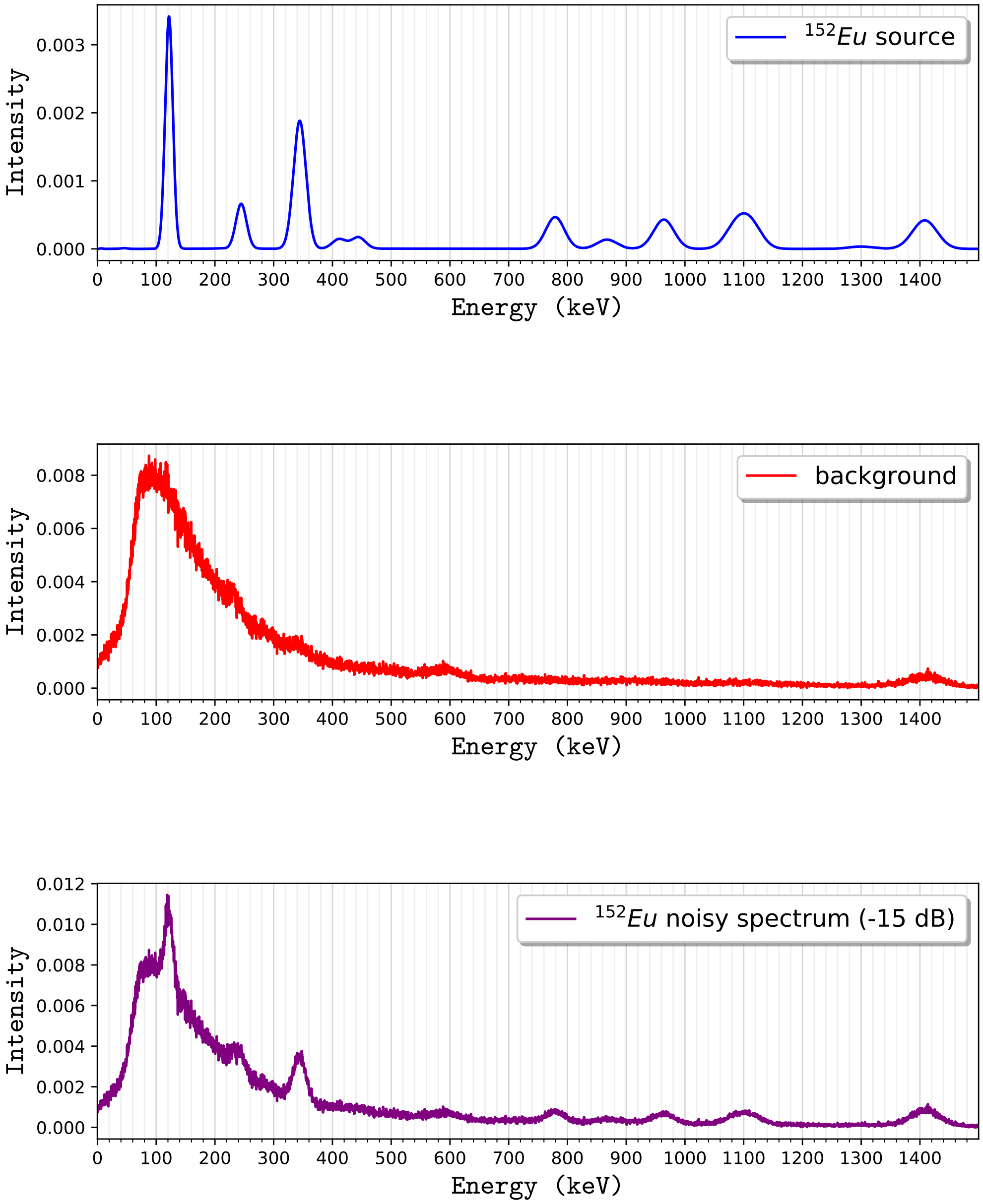

Figure 3.5: Top graph is the photoelectric for ^{152}Eu simulated by MCNP. The middle graph is a measured background spectrum using a NaI gamma-ray detector. The bottom graph is a mixture of the two signals at an SNR of -15 dB used as a training vector.

increments. This creates training data with a variety of noise and intensity levels.

3.3.3 Data Augmentation

Gamma-ray detectors are prone to calibration drift over time, ultimately resulting in measured energy levels shifting to neighboring channels. To model this behavior, for each of the generated training examples, with a $\frac{2}{3}$ probability an additional version of the same spectra is shifted left or right 1 to 10 channels chosen uniformly with a 50% probability. An example of this data augmentation can be seen in Fig. 3.6.

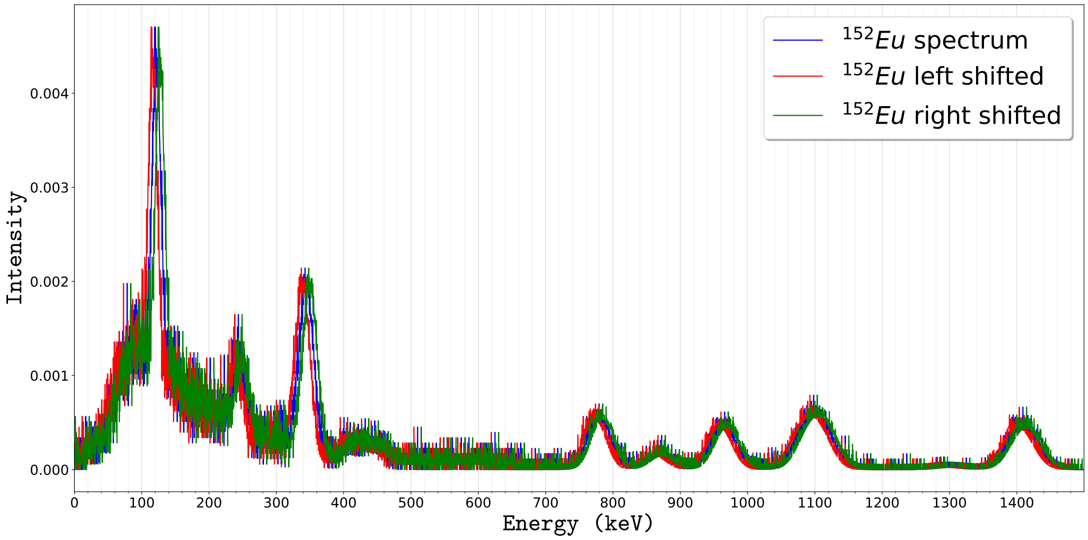

Figure 3.6: Mixture of a ^{152}Eu MCNP simulated spectrum and a measured background at 0 dB SNR with left and right shift data augmentations.

These steps used for generating training and validation data are intended to capture a variety of potential conditions that can occur and affect the measurement of radionuclides using gamma-ray detectors, thus forming a well-disciplined data set for training a gamma-ray spectrum denoising model.

3.4 Model Training

NNs are known as universal function approximators, meaning that given appropriate weights they can model complex transformations, mapping an input vector to an output vector within some upper error bound ϵ. They are especially adept at learning weights for well-behaved, constrained distributions, which can result in a small ϵ. This process is made easier with normalization techniques such as data standardization, which shifts and scales the data to have a mean of zero and unit variance. However, this primarily benefits the first layer of a model, since weight updates during training result in a phenomenon known as internal covariate shift, where the output of one layer shifts the distribution of data that becomes the input to the subsequent layer. This is especially problematic for deep networks, since small shifts between several layers can result in a large shift at the output layer.

In 2015, Ioffe *et al.* [26] introduced a technique called *Batch Normalization* (BN). BN consists of 2 operations and only requires 2 learnable parameters, γ and β. The first operation normalizes the output of the previous layer which is the input x_i to the BN layer, where i is the i^{th} input. This is performed by subtracting the mean of the batch μ_B, and dividing by the square root of the sum of the variance of the batch σ_B^2 and ϵ, a small constant for numerical stability, as seen in Eq. (3.3). The next operation is to scale the normalized input by γ and shift it by β, as seen in Eq. (3.4). This process is performed on the linear output of a layer prior to the non-linear activation, thus reducing the chance of saturating the function which can result in erratic outputs and poor gradients. Ioffe *et al.* found that not only does this increase the accuracy of CNNs, but also results in quicker model convergence by allowing for the use of higher learning rates and being more tolerant to the variance from one weight initialization to the next:

$$\hat{x}_i = \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}}, \tag{3.3}$$

$$y_i = \gamma \hat{x}_i + \beta. \tag{3.4}$$

The shifting and scaling operations performed by the BN process helps maintain the Gaussian distribution of the standardized data from one layer to the next, enabling models to be more adept at learning and recognizing Gaussian distributed data. This was the motivation for Zhang *et al.* to choose to have their image denoising model learn to predict the AWGN and remove it in postprocessing, instead of directly predicting a denoised output. However, since the distribution of photoelectric energy measurments collected by gamma-ray detectors is Gaussian in shape, and the background and detector noise is more erratic, for this research I have chosen the opposite approach. Thus, the input to the GS-DnCNN is a noisy gamma-ray spectrum and the target for the output is a denoised version of the noisy input.

3.4.1 Optimization and Regularization

The objective function used to calculate the error for each batch, often referred to as the loss, is the traditional *Mean Squared Error* (MSE). This is the average of the squared difference between the denoised output of the model and the target clean spectra that was used to create the noisy input spectra for the training and validation sets. This objective function is minimized using the *Stochastic Gradient Descent* (SGD) variant *Adaptive Moment Estimation* (ADAM) presented by Kingma *et al.* [27] in 2014. This optimization technique produces an adaptive learning rate for each parameter based on a running estimation of the first and second order moments of their gradient updates. It has been found to be suitable for both noisy and sparse gradients by naturally annealing each parameter's weight updates based on a global learning rate. This results in much quicker training than traditional SGD and pairs well with BN and LReLU to produce robust, fast convergence.

The original ADAM optimizer contains a term known as weight decay. Weight decay for NNs is analogous to L2-regularization, also known as Ridge Regression, a penalty term added to the loss that uses the 2-Norm of the weights to reduce overfitting in models with a large number of parameters. Weight decay in NNs encourages weights to be small, which produces more stable and proportional output changes with respect to the input. However, the weight decay term for the ADAM optimizer is scaled by the adaptive estimate, therefore it is not true L2-regularization. In 2019 Loshchilov *et al.* [28] presented ADAMW, a modified version of ADAM which decoupled the weight decay from its adaptive elements, thus correctly calculating true L2-regularization which results in more effective weight decay. Therefore, for this research I have chosen to use the ADAMW optimizer.

In addition to the L2-regularization provided by the ADAMW optimizer I have also implemented L1-regularization for the GS-DnCNN to use during training. L1-regularization, also known as Lasso Regression, uses the L1-norm of the model's weights as a penalty to avoid overfitting for overly complex models. The L1-norm encourages the weights for the less impactful features to be zero. This creates a more sparse representation throughout the network and leads to lower activity levels, reducing power consumption in neuromorphic models. Since the GS-DnCNN is a denoising model for gamma-ray spectrum, which theoretically should have a sparse representation with signal only present where photoelectric exists for the measured radionuclide, L1-regularization helps squash the noise floor to be near or at zero for all areas where it does not identify the Gaussian shape of photoelectric measurements.

3.4.2 Activations

LReLU is a variant of the max function ReLU [29] which is defined as $f(x) = max(0, x)$. ReLU is a non-linear activation function originally applied to image recognition CNN models. It now has become the most commonly used activation for deep NNs due to its low complexity computation and a strong gradient of 1.0 for all positive values. Traditional activations like the logistic function have a maximum gradient of 0.25, which results in slow learning. Tanh has a maximum gradient of 1.0, but like the logistic function the gradients quickly decline on either side of this global maxima. Both activations quickly lead to vanishing gradients if either of them are saturated. The ReLU activation is linear and unbounded for all positive values and therefore does not lead to vanishing gradients or saturation issues.

Models trained with ReLU have shown a significant decrease in convergence time, and good local minima can still be found despite the strong gradient when paired with an adaptive learning rate optimizer. However, the gradient of the activation is undefined for non-positive values and thus leads to a condition known as dead neurons. Since SGD optimization algorithms use the derivative of the activation with respect to the error to determine a gradient, when the signal to the activation is not positive it produces zero gradient and therefore the backpropagated weight update for the neuron is zero and no learning occurs. Due to this, models that use ReLU activations have a tendency to contain a significant number of neurons that do not contribute to the output signal for the network. For classification networks this is not a major issue since other paths through the network learn to compensate, however, it is a waste of computational resources.

LReLU, defined in Eq. (3.5), has the similarity to ReLU in the fact that it is linear for all non-negative numbers and thus has the same strong gradient for these values. However, for all other values instead of being 0 it has a very small gradient of 0.01. This produces a very similar activation shape as ReLU, and allows for weight updates when the signal to it is not positive. Therefore, models using LReLU not only have the quick learning property of the ReLU activation, but also avoid the problem of dead neurons.

$$f(x) = \begin{cases} x & \text{if } x > 0 \\ 0.01x & \text{otherwise.} \end{cases} \tag{3.5}$$

3.4.3 Weight Initialization

BN helps keep the flow of data through the model Gaussian and L2-regularization encourages the weights to be small, but both of these processes are directly affected by the weight space. If weights in a layer are skewed then BN must learn parameters to overcome this, and the optimal range of weight values must be proportional to the number of inputs to the layer or else the output of the layer can end up being exponentially amplified. Therefore, to address these issues, proper weight initialization is key to avoiding poor local minima in the weight space and for fast convergence. In 2010 Glorot *et al.* [30] introduced Xavier initialization, which samples weights from a uniform distribution proportional to the number of inputs to the layer, n_l, and the number of outputs, n_{l+1}:

$$W_l \sim U\left(-\frac{\sqrt{6}}{\sqrt{n_l + n_{l+1}}}, \frac{\sqrt{6}}{\sqrt{n_l + n_{l+1}}}\right). \tag{3.6}$$

This weight initialization scheme has become default across most deep learning

APIs. The derivations and conclusion in the paper assume that the activations have a zero mean for the outputs of each layer to maintain a near Gaussian distribution, since $\mathbb{E}[x^2] \neq Var(x)$ unless $\mathbb{E}[x] = 0$. This works well for linear outputs and Tanh activations which are symmetric across the origin, and the logistic function which has a small mean of 0.5. However, the ReLU and LReLU activations are unbounded functions with a significantly different slope for positive and negative values, thus are not symmetric. Therefore, in 2015 He *et al.* [31] introduced Kaiming Normal initialization, which samples weights from a normal distribution:

$$W_l \sim \mathcal{N}\left(0, \frac{2}{n_l}\right). \tag{3.7}$$

They showed that using ReLU resulted in $\mathbb{E}[x^2] = \frac{1}{2}Var(y)$, and thus the variance of each layer is defined as Eq. (3.8):

$$Var(y_l) = \frac{1}{2}n_l W_l Var(y_{l-1}) \implies \frac{n_l}{2}Var(W_l) = 1, \forall l. \tag{3.8}$$

By sampling from a normal distribution with a mean of 0 and standard deviation of $\frac{\sqrt{2}}{\sqrt{n_l}}$, near unit variance is achieved. This not only results in sufficiently small weights being generated that do not exponentially amplify the signal from layer to layer and produce large L2 penalties, but also assists in keeping the signal closer to a Gaussian distribution so BN layers does not have to overcompensate for a skewed distribution. Since the GS-DnCNN uses BN, L2-regularization, and the ReLU variant LReLU, which the He *et al.* paper also showed was ameliorated using Kaiming Normal initialization, for this research I have switched all convolution layers from the PyTorch default of the uniformly distributed Xaviar Initialization Eq. (3.6) to the normal distribution described in Eq. (3.7).

3.4.4 Model Validation

Deep NNs are prone to overfitting, where the model learns the distribution of the training set too precisely and therefore cannot properly approximate the desired function on out-of-sample data, inputs that were not used to train the model. Therefore, during training a subset of the training set is used as a validation set to test the model between epochs, one cycle through the entire training set. The validation set is not used for weight updates and therefore is well suited for evaluating how well the model generalizes the learned function on data that it is not used during the learning phase of training.

In addition to monitoring the validation loss, the MSE of the validation set, it is also helpful to determine the perceptual quality of the denoising that is occurring on the validation set. For the generation of training data, traditional SNR is used to build examples, which is the ratio of signal power to noise power. Due to the high dynamic range of many signals, the ratio is converted to a log space known as *Decibels* (dB) to approximate human perception. However, this requires knowledge of the clean signal and the noise, so once a noisy signal has been enhanced by a model it is difficult to approximate. Therefore, for image denoising it is common to use the metric *Peak Signal to Noise Ratio* (PSNR) to quantifying the effect of the enhancement. PSNR determines the quality of a signal based on the maximum intensity, MAX_I, and the MSE between a target signal and a predicted signal, as seen in Eq. (3.9). Thus it is a good indicator of how well high-intensity regions of a signal present over noise:

$$PSNR = 10 \log_{10} \frac{MAX_I^2}{MSE(\text{target, prediction})}. \tag{3.9}$$

Low values represent reconstructions where noise obfuscates high-intensity regions, while high values represent higher quality enhancements with lower noise floors. Since the purpose of the GS-DnCNN is to reconstruct strong photoelectric peaks while suppressing as much noisy signal as possible, it is a good metric to determine the quality of a gamma-ray spectrum which ideally should be mostly sparse. Therefore, during training the MSE and the PSNR is calculated for the validation set between each epoch to show how well the model is learning the denoising task.

During training, the desired results from epoch to epoch is to see both the training loss and the validation loss decreasing. However, at some point overfitting will most likely occur, signaled by the training loss continuing to decrease while the validation loss plateaus or begins to increase. Two other forms of regularization are used to minimize the validation loss as much as possible before this begins to occur. First, any time the validation loss begins to plateau the learning rate is decayed by a factor of 10. This forces the model to take smaller descent steps after each of these occurrences, allowing for a more thorough exploration of the weight space in search of a narrow escape path out of a potential saddle point. Second, early stopping is used to identify and save the weights of the epoch with the lowest validation loss. Thus, after signs of overfitting begin to show and decaying the learning rate no longer results in better performance, training can be halted. The weights of the model after the epoch that produced the lowest validation loss is considered convergence. This is the point at which the model can approximate the learned function at the lowest ϵ given a set of hyperparameters, parameters that determine the complexity of the model and the learning process.

3.4.5 Hyperparameter Tuning

Training a deep NN is a computationally complex task. Traditional deep models consisted of a few layers and had anywhere from a few hundred to a few thousand learnable parameters. Some of the most recent SOTA models have as many as a trillion parameters and have been successfully trained with over a thousand layers. The number of training examples required for a deep NN is generally proportional to the number of their parameters, therefore requiring anywhere from megabytes to terabytes of data. Training these models is assisted by *Graphical Processing Units* (GPU), which are able to perform large amounts of floating point operations per-second. Significant research and development of these over the past several years has resulted in GPUs capable of performing trillions of operations per-second, without which would have made the most recent and complex models' training computationally prohibitive. Deep NNs have many hyperparameters, some of which must be tuned independently or in conjunction with others during subsequent trainings, thus compounding the time required to achieve ideal results.

For the first set of training experiments I conducted with the GS-DnCNN I needed to determine the ideal size of the model. If the model has too few parameters it will not have the expressiveness to approximate the denoising function well. If the model has too many parameters it will be overly complex and thus not generalize well to the variations present in the out-of-sample data. Therefore, I began by tuning the number of convolution kernels there should be in each layer, how wide the model is, and the number of convolution layers, how deep the model is. The combination of which should determine the ideal size required for the model to efficiently perform the denoising task. I setup up a grid search of the two parameters with the number of kernels in a typical range found in other CNNs, {16, 32, 48, 64}, and the number of layers in increments of 5 up to the maximum of 20 that can properly be trained given the fairly simple architecture, {5, 10, 15, 20}. Results of these training experiments can be seen in Fig. 3.7 which shows an ideal size for the model to contain 20 convolution layers with 32 kernel filters per-layer, resulting in a validation loss of approximately 8.372×10^{-3} and a PSNR of 77.05 dB.

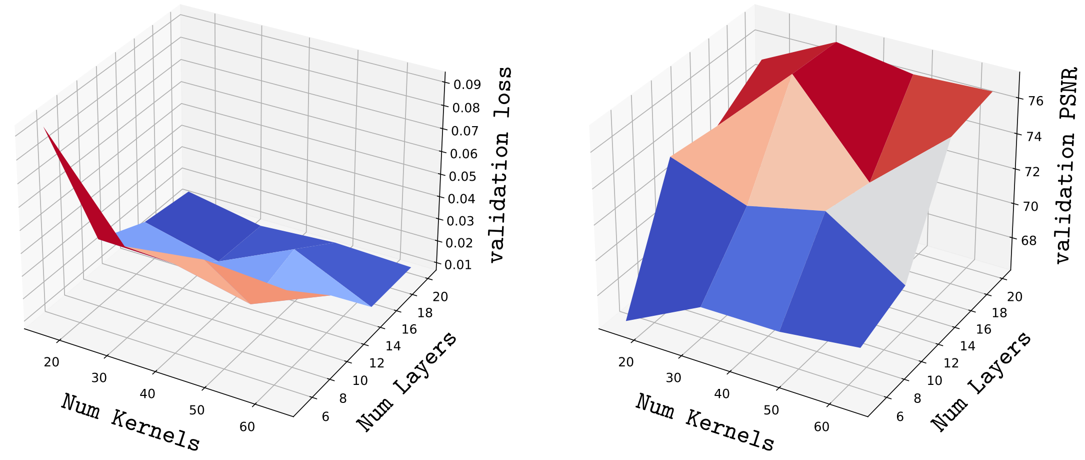

Figure 3.7: Hyperparameter tuning for network size: number of kernels per-convolution layer and number convolution layers. The grid search of these two hyperparameters resulted in finding the ideal depth for the GS-DnCNN to be 20 convolution layers each with 32 kernels.

Once I had determined the ideal size for the GS-DnCNN to approximate the gamma-ray spectrum denoising function, I needed to know how much I can penalize the model's weights during training, allowing it to learn a more generalized function, not just a close fit to the training data. Therefore, I experimented with different scalar values for the L1 and L2-regularization. Regularization creates less precises gradients with respect to the training data, but still in the correct direction, significantly slowing down training from approximately 30 mins ($\sim$ 50 epochs) to 1.5 hrs ($\sim$ 130 epochs) using 4 Nvidia RTX 2080Ti GPUs. Due to the sensitivity of the model to regularization, several grid searches were performed starting with increments in powers of 10 from 1.0×10^{-8} up to 0.1, and with increments of 0.1 from 0.1 up to 2.0, {1e-8, 1e-7, ..., 0.1, 0.2, ..., 1.9, 2.0}. These scalar values are used to determine the amount of each regularization to apply. Eventually my grid searches led to determine a value in [1e-7, 1e-6] for the L1-regularization and in [0.5, 1.5] for L2-regularization, as seen in Fig. 3.8. This resulted in a model that achieved a validation loss of 9.168×10^{-4} and a PSNR of 90.62 dB, using an L1 scalar of 7.75×10^{-7} and L2 scalar of 1.0.

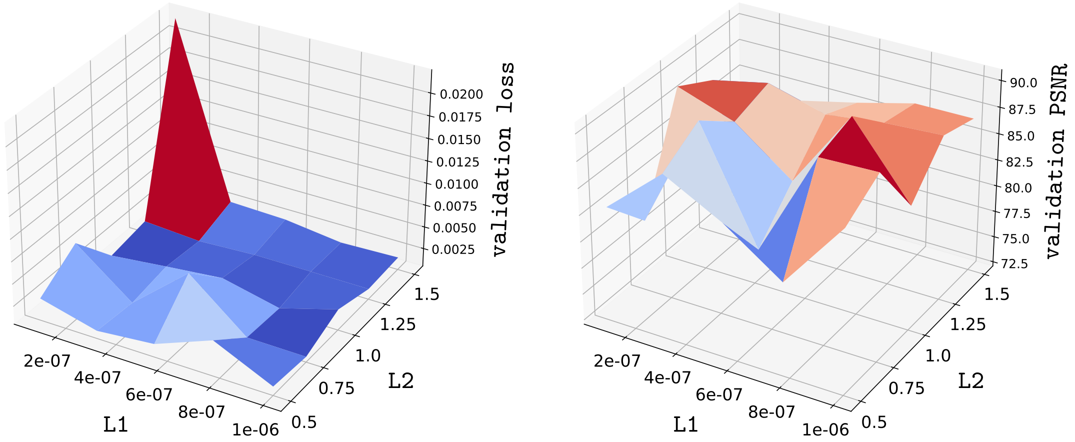

Figure 3.8: Hyperparameter tuning for weight regularization: L1 and L2. The grid search of these two hyperparameters resulted in finding ideal regularization scaling coefficients for the GS-DnCNN, 7.75×10^{-7} for L1 and 1.0 for L2.

Finally, I needed to know the ideal speed at which training can occur given the size of the network and the amount of regularization. The two main hyperparameters that judge this are learning rate, how large weight updates are, and batch size, how many training examples the gradient is averaged over before a weight update occurs. For this I started with a wide sweep of commonly used values for training deep learning models, {1e-5, 1e-4, 1e-3, 1e-2, 1e-1} for learning rates, and {16, 32, 64, 128, 256} for batch sizes (due to memory architectures in parallel systems, powers of 2 are more efficient for batch processing in GPUs). This led to fine tuning my search for learning rate from the range of [5e-4, 5e-3] and batch sizes of {16, 32, 48, 64}, as seen in Fig. 3.9. Ultimately I found a learning rate of 5.0×10^{-4} and batch size of 32 to produce the best results of 3.34×10^{-4} for the validation loss with a 92.54 dB PSNR.

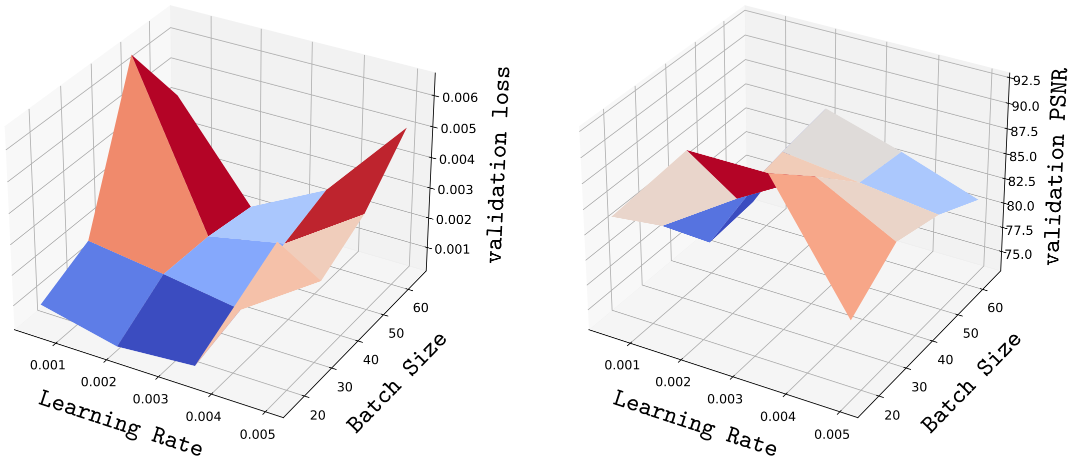

Figure 3.9: Hyperparameter tuning for learning parameters: learning rate and training batch size. The grid search for these two hyperparameters resulted in determining an optimal starting learning rate of 5.0×10^{-4} and averaging the gradient across a batch size of 32 examples per-weight update when training the GS-DnCNN.

In addition to finding good hyperparameters for the GS-DnCNN model, the tuning experiments also showed the strong inverse correlation between MSE and PSNR, validating that minimizing MSE between a clean and noisy gamma-ray spectrum optimizes the model to perform well on a perceptual metric used for evaluating noise reduction in a variety of domains.

3.4.6 Model Training Results

The training statistic for the version of the GS-DnCNN with the best determined set of hyperparameters can be seen in Fig. 3.10. The top graph shows the training and validation loss curves over the course of training, while the bottom graph shows the PSNR on the validation set during the same process. In total the model trained for over 180 epochs. The vertical lines at epochs 1, 23, 84, 153, and 164 show the learning rate adjustments from 5.0×10^{-4} to 5.0×10^{-8} due to plateaus of 5 epochs in the validation loss.

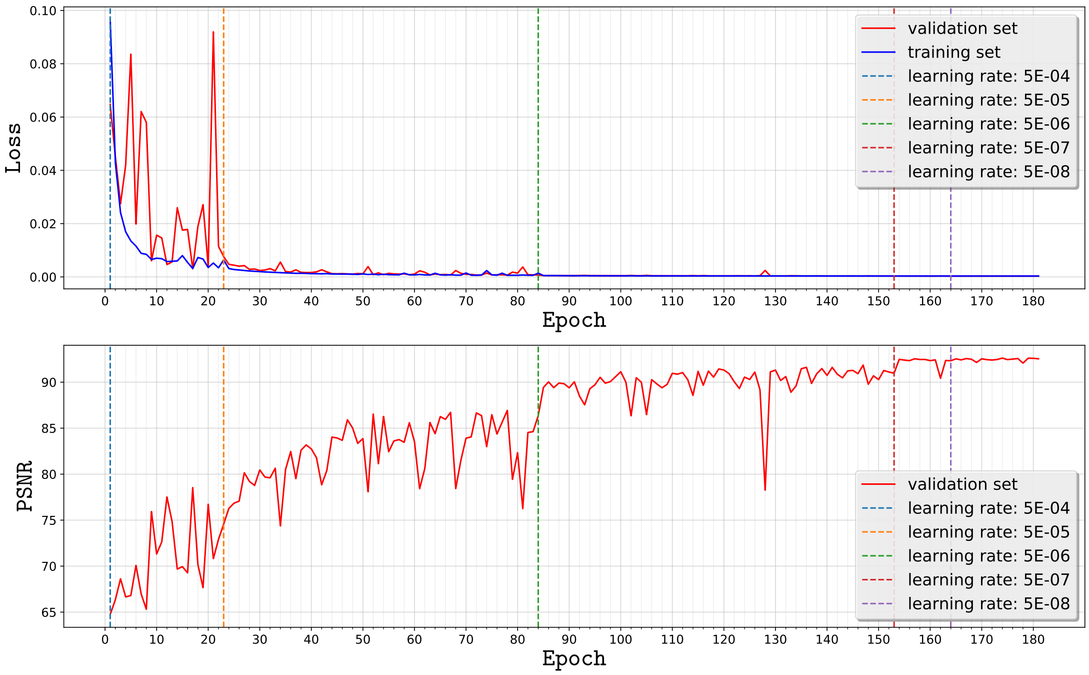

Figure 3.10: Top figure shows training and validation loss curves for best the GS-DnCNN model. Bottom figure shows the validation PSNR for the best GS-DnCNN model. Both figures show the learning rate decay on validation loss plateaus.

In early epochs the training loss drops quickly, showing the model is learning the denoising function well for the training set. However, the validation loss and PSNR are erratic due to the relatively high learning rate and running average of gradient change, resulting in large jumps in and out of areas of the weight space where there is a significant effect on generalization. Still, the validation PSNR trended towards good improvements. After a local minimum in validation loss around epoch 17, resulting in a PSNR around 78, the learning rate is annealed by a magnitude starting at epoch 23. With this smaller learning rate finer steps are taken in the weight space leading to a more stable descent of the loss function and therefore a less erratic, upward trend in PSNR. After epoch 84 the loss is annealed again and the increase in metrics becomes much more stable yet slower due to the very small learning rate and running average of the gradient. At epoch 181 the model has reached the point known as convergence. At epoch 186 the learning rate is annealed again and at epoch 192 training is stopped (not shown in figure) due to the fact that the validation loss has not decreased for 10 epochs. Due to early stopping, where the weights from the epoch that resulted in the lowest validation loss are saved, the final model produces an average PSNR enhancement of 22.01 dB on the validation set of examples with a uniformly distributed SNR from -25 dB to 50 dB.

Fig. 3.11 shows an application of the GS-DnCNN on a noisy gamma-ray spectrum from the validation set. The '-.-' line (green) shows the noisy input spectrum, the solid line (blue) shows the clean target spectrum that was used to build the noisy spectrum, and the '..' line (red) shows the resulting denoised spectrum produced by the model. As can be seen from the figure, despite an extremely low SNR, the model suppresses virtually all noise and recreates the clean target with high fidelity, resulting in a PSNR enhancment of 53.26 dB.

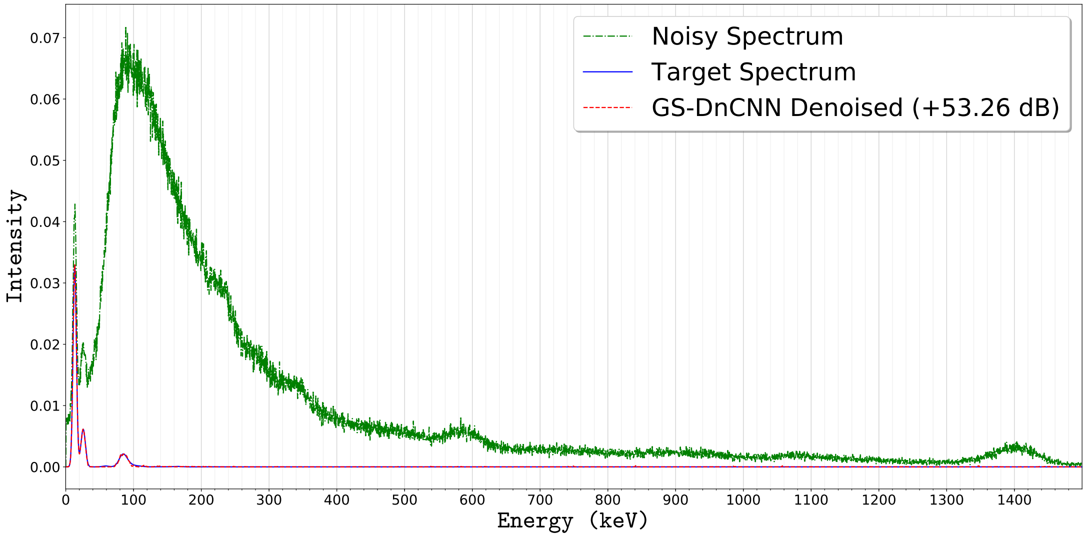

Figure 3.11: GS-DnCNN denoised, low SNR spectrum from validation set with 53.26 dB enhancement.

Fig. 3.12 shows the GS-DnCNN's performance on a high SNR example from the validation set. Due to the extremely low level of noise, the small reconstruction errors in the photoelectric peaks outweight the enhancement due to the surpressed noise, resulting in a 10.08 dB corruption in the signal. However, the reconstructed photoelectric is still a very close approximation to the desired clean target spectrum.

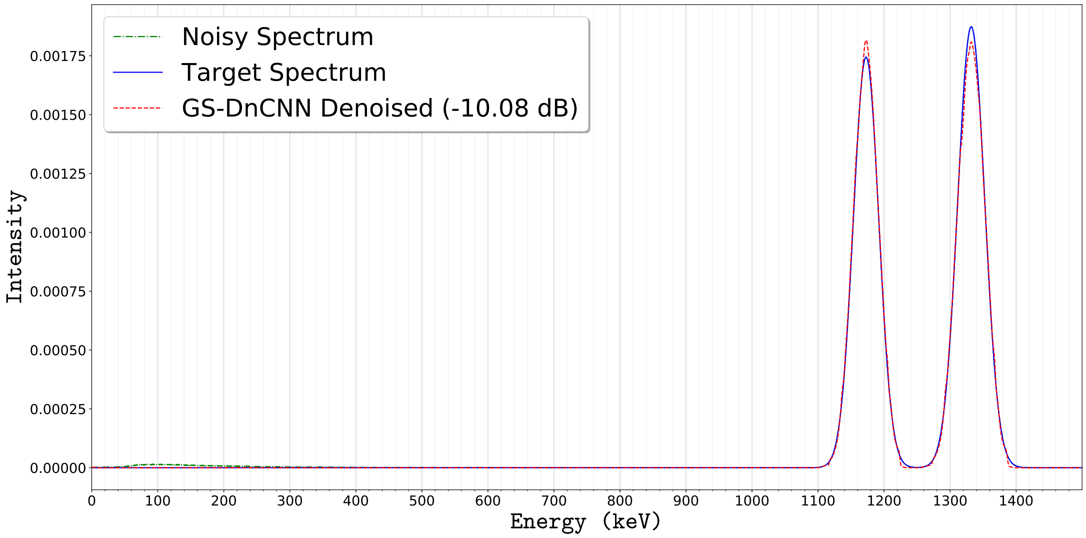

Figure 3.12: GS-DnCNN denoised, high SNR spectrum from validation set with 10.08 dB of corruption.

The quality of reconstruction for the validation set is a good proof of concept for the model, however, due to the validation examples being from the same distribution as the training examples, it is not a good evaluation of the efficacy for using the model to denoise non-synthetic gamma-ray spectrum. Real, measured gamma-ray spectrum contain background radiation, detector noise, and measurement error, so there is no clean target that can be compared to a denoised measured spectrum to use as an objective measurement. Therefore, subjective measurement is required to get a rough idea for the quality of the learned denoising function. For testing the model, gamma-ray spectrum from the measured platform, in this case a NaI gamma-ray detector, are used as a test set. Fig. 3.13 shows a 300 second measurement of ^{235}U at a distance of 2" from the detector and the reconstruction of the spectrum's photoelectric after processing it through the GS-DnCNN for denoising. As can be seen from the figure, the model reproduces well proportioned, near Gaussian shaped photoelectric peaks. The areas between these peaks and for energy levels above approximately 250 keV, where no photoelectric is generated by the radionuclide, is mostly or entirely suppressed.

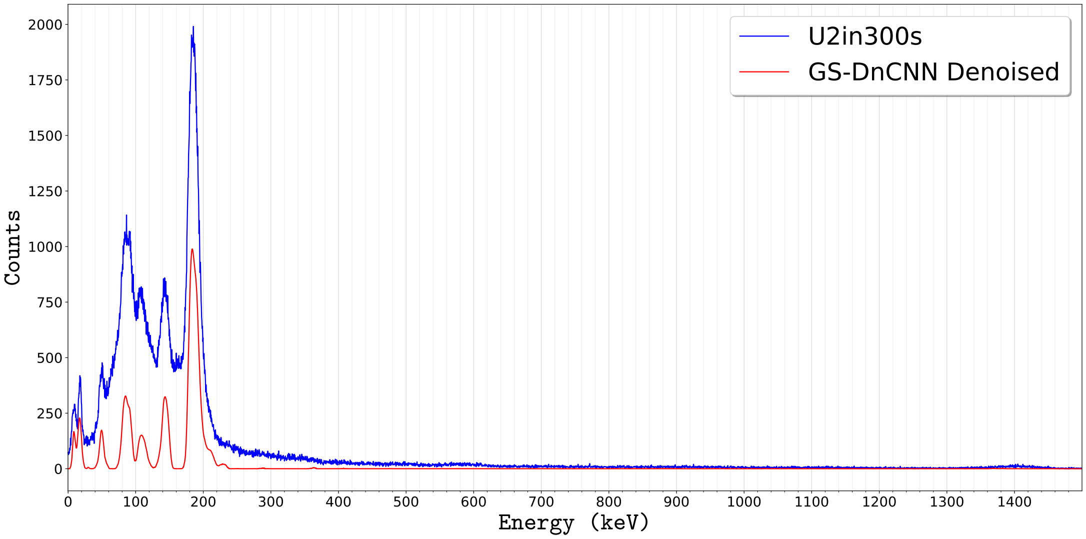

Figure 3.13: GS-DnCNN denoised, 300 second measurement of ^{235}U at a distance of 2" from the NaI gamma-ray detector. The reconstructed photopeaks due to the denoising are near Gaussian in shape and most of the background and noise has been suppressed.

Fig. 3.14 shows a comparison between the GS-DnCNN denoising algorithm and the traditional noise reduction approch of background subtraction for a measured ^{239}Pu spectrum. The top-left figure shows the raw spectrum along with the resulting spectrum after subtracting its corresponding background spectrum. The top-right figure shows the same raw spectrum and the resulting spectrum after being processed by the GS-DnCNN. The bottom-left figure shows a comparison between the two types of noise reduction, background subtraction and GS-DnCNN denoising, showing that the GS-DnCNN successfully recreates a large mass of the key photoelectric energy required to identify the radionuclide without the knowledge of the background statistics that are contained in the background spectrum that is used for background subtraction. Finally, the bottom-right corner shows both spectra normalized by their magnitude, showing that the GS-DnCNN denoised spectrum produced very similar results for the strong photoelectric peaks with a significantly larger reduction in noise between them and at energy levels above the maximum photoelectric that is generated by the radionuclide.

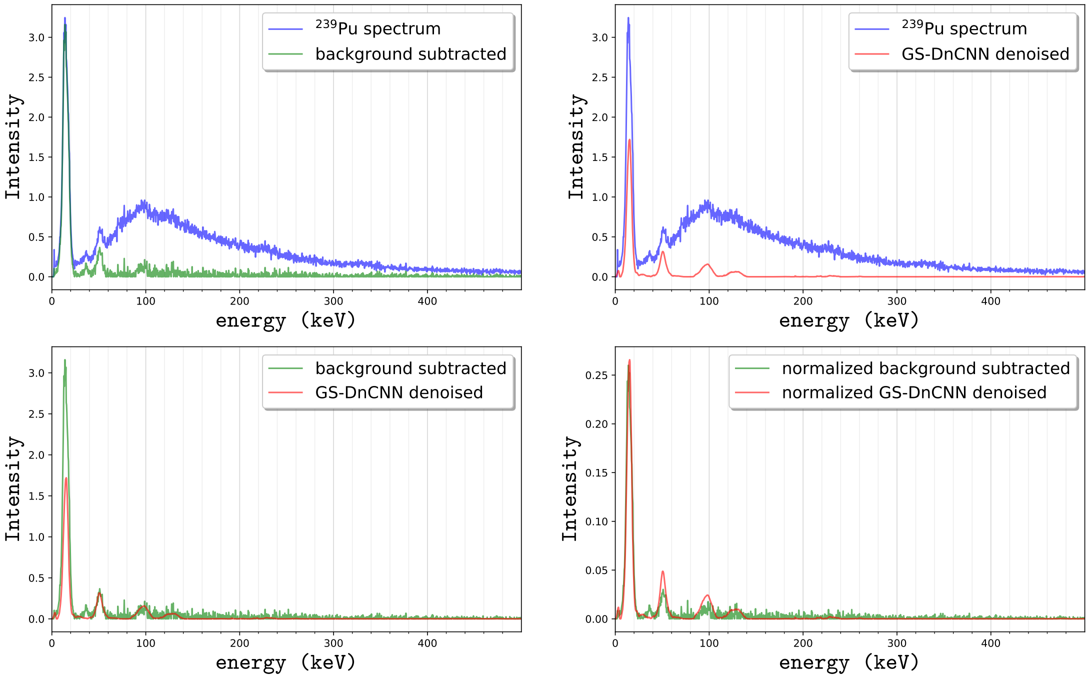

Figure 3.14: Comparison of background subtraction vs. GS-DnCNN denoising on a ^{239}Pu measured spectrum from the test set. The top-left figure shows the spectrum both before and after background subtraction. The top-right figure shows the spectrum both before and after GS-DnCNN denoising. The bottom-left figure shows a comparison of the two versions of denoised spectrum. The bottom-right figure shows normalized versions of both denoised spectrum. The final version of the GS-DnCNN denoised spectrum results in very similar photoelectric distributions as the background subtraction produced but with significantly less noise.

The above shown plutonium and uranium examples demonstrate that the GS-DnCNN is capable of filtering out nearly all background and detector noise while recreating Gaussian shaped photoelectric peaks with good fidelity. While it is possible some amount of photoelectric will be suppressed or removed during this process, it is most likely near or below the noise floor and thus is indistinguishable. Additionally, for radionuclide identification only a subset of the photoelectric energy produced by a radionuclide is required to properly identify it, so visual inspection by domain experts or automated identification algorithms should still be capable of correct classification despite some amount of information loss.

3.5 Neuromorphic Architecture

Traditional neuromorphic implementations of NNs have been prohibitive for complex models due to high synaptic density. Recent advances in the manufacturing of memristors and memristive crossbars, memristors architectured in a 2D-grid layout, have led to a significant amount of research for their use in designing and building more complex neuromorphic models. This stems from a crossbar's nano-size, low-power consumption, and ability to approximate the dot-product of the input and a column of memristors, and thus perform a vector-matrix product in parallel using the entire crossbar.

The mathematical operation for a column in a memristive crossbar is defined by Eq. (3.10):

$$V_{out_j} = \frac{\sum_i V_{in_i} C_{i,j}}{\sum_i C_{i,j}}, \tag{3.10}$$

where V_{in_i} is the i^{th} row input voltage, V_{out_j} is the output voltage of the j^{th} column, and $C_{i,j}$ is the capacitance of the memristor for the i^{th} row and the j^{th} column of the crossbar. However, due to the fact that capacitance cannot take on negative values, each column of the weight matrix must be duplicated and negative values must be represented as their absolute value. Therefore, to perform this operation there must be one column for the positive weights, with the negative entries set to zero, and one column for the absolute values of the negative weights, with the positive weight entries set to zero. The dot-product of the positive side V^+ is summed with the inversion of the dot-product of the negative side V^- using an op-amp, thus resulting in an output voltage V that is the difference between the two dot-products.

3.5.1 Memristive Convolution

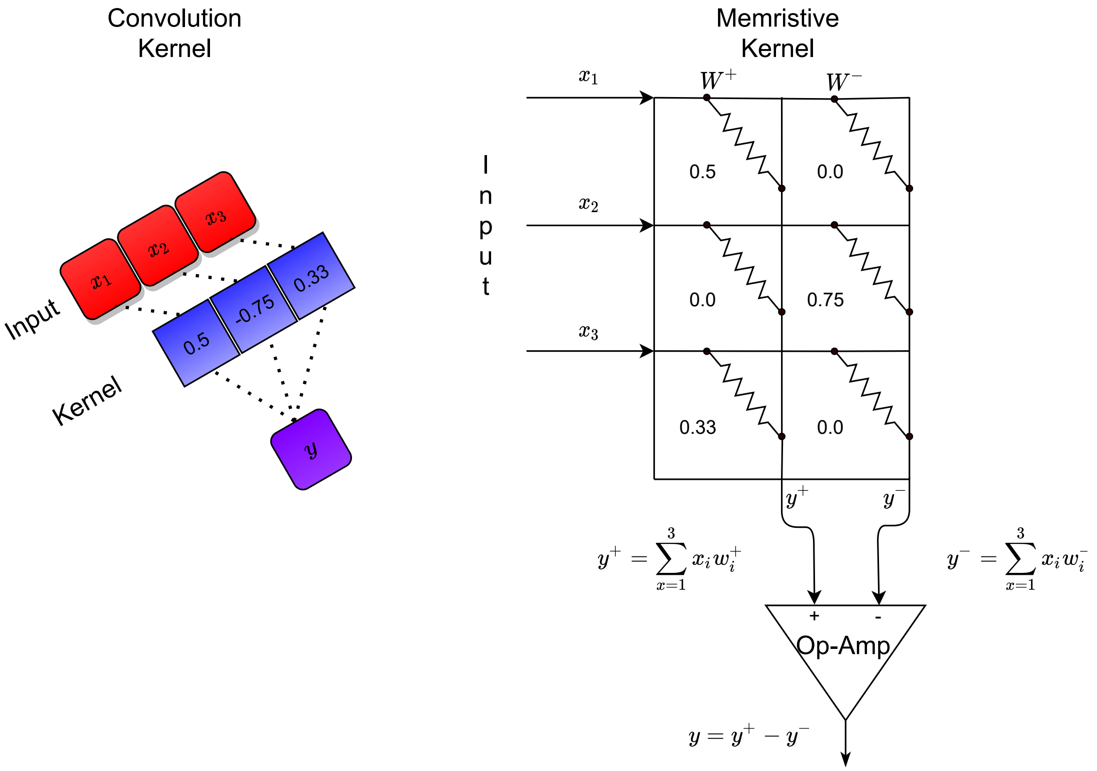

Figure 3.15: Neuromorphic implementation of a convolution kernel using memristors. Two dot products are performed using a memristive crossbar, one between the input and the postive weights, and one between the input and the absolute value of the negative weights. The output is the sum of the first with the inversion of the second.

An example of a using memristive crossbar as described above for convolution operations was presented by Gao *et al.* [32] in 2016. Their work revolved around image classification and used 2D inputs and 2D kernels, therefore requiring dimensionality reduction for efficient processing using crossbars. For a single 1D-convolution operation the output is the dot product between an input vector x and the weights of the convolution kernel W, $\langle x, W \rangle$, therefore no dimensionality reduction is required. A comparison between a traditional 1D-convolution kernel used by the GS-DnCNN and a memristive implementation where the positive and negative weights are split between two columns can be seen in Fig. 3.15, where the input voltage V_{in} is represented by x, the capacitance of the memristors is represented by the weights of the kernel W, and the output voltage V_{out} is represented by y.

3.5.2 Memristive Batch Normalization and Leaky ReLU

In 2021 Ran *et al.* [33] presented a memristor-based implementation of GoogleLeNet, a 22-layer deep CNN that won the 2014 ImageNet Large-Scale Visual Recognition Challenge. GoogleLeNet utilized the newly introduced NN feature batch normalization. Once training is complete, this operation only requires a static scaling and shifting parameter for each output kernel in a convolution layer, operations that are often fused into the activation function for efficiency when optimizing NNs prior to deployment. Ran *et al.* shows a derivation of the final static BN operation and how it has the same form as the convolution operation where the scaler value is represented by a diagonal weight matrix and the shift value is a bias term. Therefore, it can also be performed on a memristive crossbar with the output of the previous convolution as its input.

In 2018 Bala *et al.* [34] introduced a memristive circuit that uses a comparator and a memristive MIN function, presented by Yang *et al.* [35], to perform a ReLU activation. With the slight modification of removing the memristors and using the comparator to trigger a single pole double throw style switch, a LReLU activation is approximated. When the negative input to the comparator is larger than the positive input, 0 V is passed through and the switch stays in the up position attenuating the summed output of the BN stage. When the positive signal at the comparator is larger the switch moves to the lower position and the BN output is passed through.

A diagram of the two circuits can be seen in Fig. 3.16, where the input is the positive and negative voltage outputs for a single convolution operation. The BN additionally requires a static voltage for the bias term. Considering the scalar value γ and offset β, and with the splitting of positive and negative weights, the BN process requires four memristors for each filter in a layer.

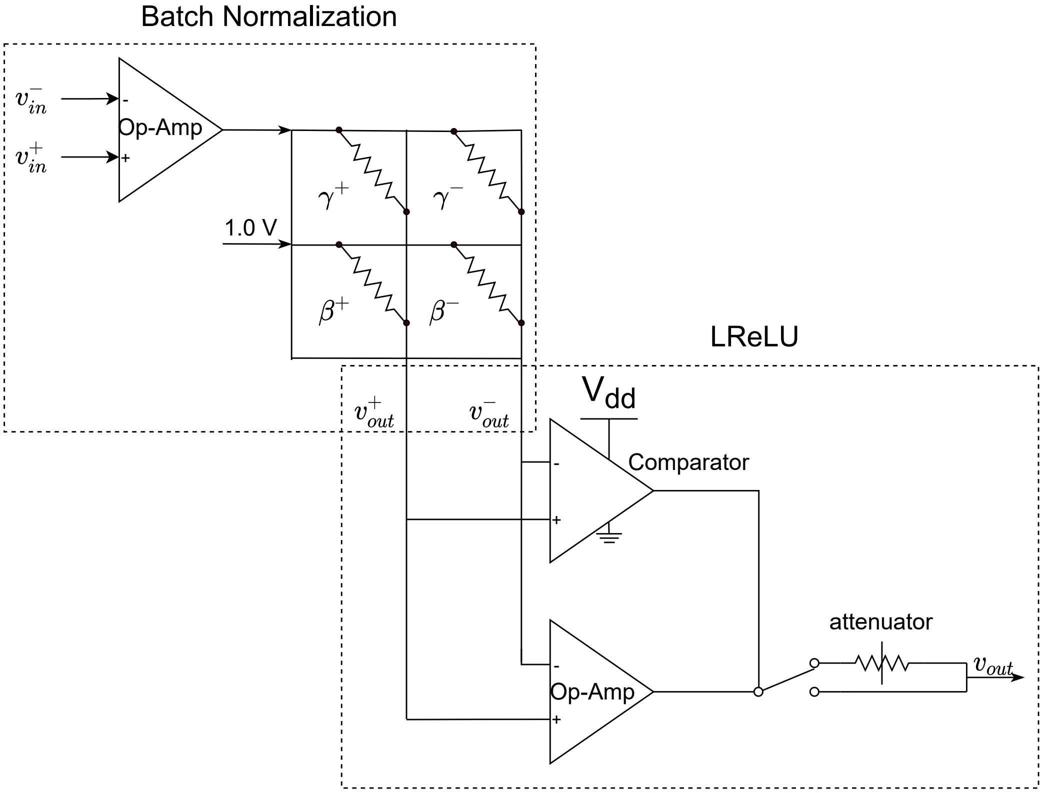

Figure 3.16: Neuromorphic implementation of batch normalization and leaky ReLU using memristors. The batch normalization stage scales (γ) and shifts (β) the signal. A comparator is used to determine if the output should be linear (if positive) or attenuated (if negative) for the non-linearity.

3.5.3 Memristive Convolutional Neural Network

In 2019 Sun *et al.* [36] proposed building a GoogleLeNet style CNN using memristive crossbars, their architecture reduced the number of parameters from the original model's 6.8 M to 4.15 M, thus requiring 8.3M million memristors to handle both positive and negative valued weights. The Ran *et al.* model used a more sophisticated weight pruning technique reducing to 0.64 M parameters, 1.28 M memristors, increasing the accuracy by $\sim 3\%$ from the Sun model on the CIFAR-10 dataset to 89.83%. The maximum power consumption of the Ran *et al.* model was estimated to be 11.7959 μW, calculated using Eq. (3.11) for a single memristor:

$$P_{mem} = \frac{\int_{t_i}^{t_r} UG_t dt}{t_r - t_i}, \tag{3.11}$$

where U is the voltage across the memristor, G_t is the conductance of the memristor at time t, and $t_r - t_i$ is the off on switching time of the memristor required for a single computation.

By applying the input to a memristive crossbar a single element for all kernels in a layer can be computed in a single computation period. Sun *et al.* [37] proposed stacking memristive crossbars with duplicate weights so as to perform all computations for a single layer in parallel. Yao *et al.* [6] built and experimented with a 5-layer deep memristive CNN and found that the parallelization technique was problematic due to slight inconsistencies between memristor characteristics from the manufacturing process. They proposed a method of processing three of the computations in parallel instead of parallelizing all computation, thus reducing the amount of error. This also assisted with the amount of latency that is produced between the calculations of the convolution layers and the fully-connected classification layer at the end of the model that can be performed in a single computation period.

The GS-DnCNN model is a fully CNN, no full-connected layers, and all layers are the same size. Therefore, all operations can be done in series which adds to the overal latency of the system, but does not create problems within the model that could present between different sized or types of layers. A diagram of an application of the GS-DnCNN layer operations to a memristive crossbar is shown in Fig. 3.17. The figure shows how the memristive crossbar can accommodate both dilated and non-dilated convolution layers with no architecture differences, just a change in input routing.

The tuned GS-DnCNN with 20 layers and 32 kernels per-layer contains a total of 56,640 parameters, thus would require 113,280 memristors to build using the proposed architecture. Assuming a maximum of 5.0 V for input, a switching time of $\sim$ 30 ns for HfO_x metal oxide memristors as empirically determined from those manufactured for the SSLCA by UNM, and a total capacitance for all memristors equal to the absolute value of all the weights in the model, the estimated maximum power consumption for the GS-DnCNN is $\sim$2.03 μW.

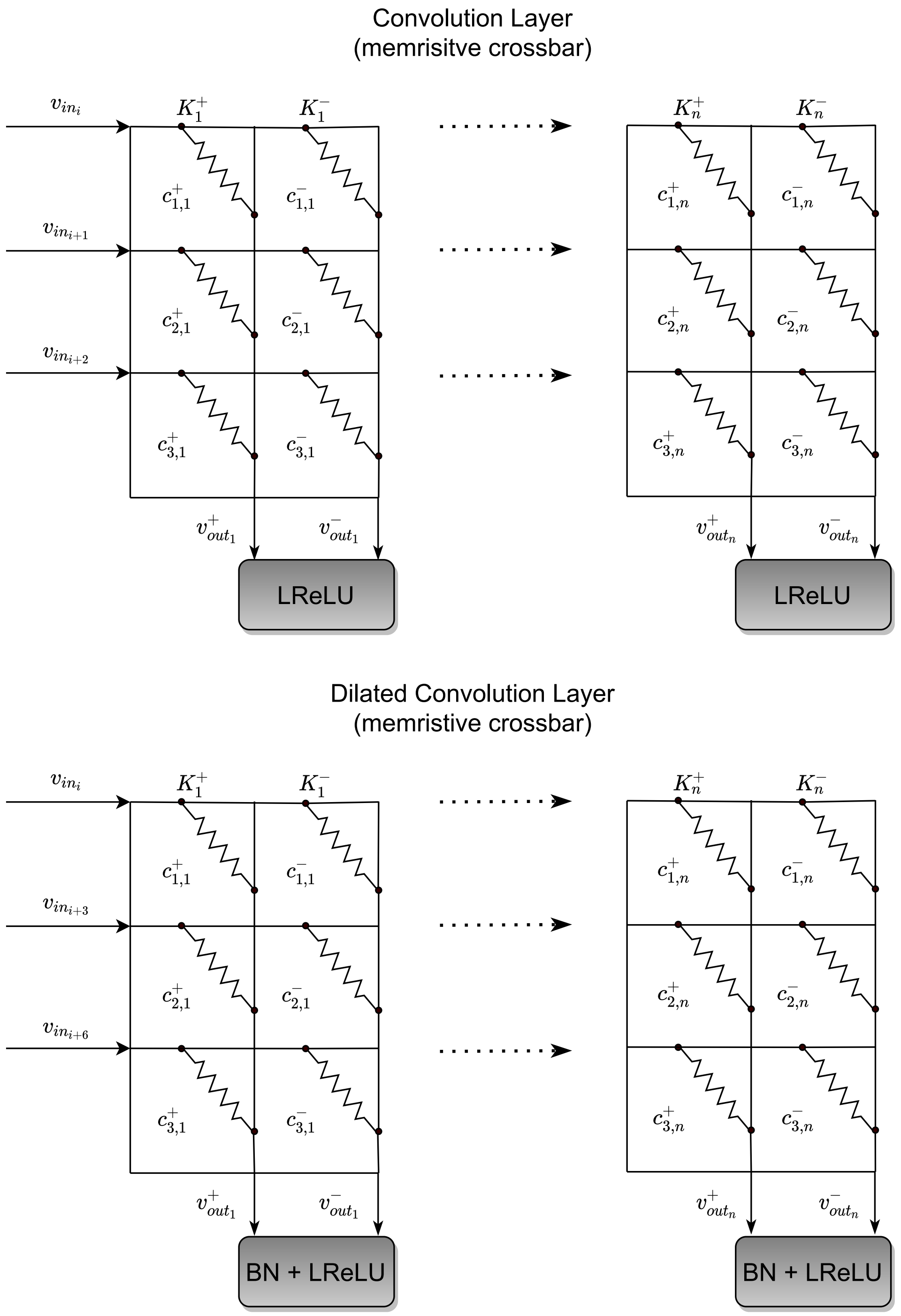

Figure 3.17: Neuromorphic implementation of convolution and dilated convolution layers using memristive crossbars. For each convolution layer, the output is passed through a leaky ReLU activation. For each dilated convolution layer, the output is passed through a batch normalization and leaky ReLU activation. The circuitry for the batch normalization and leaky ReLU functions are described in Fig. 3.16.

Chapter 4

Radionuclide Identification Results Using Gamma-Ray Spectrum Denoising

4.1 Test Set Identification and Denoising Models

The purpose of this research is to develop an efficient, neuromorphic RID, therefore the final stage of testing consisted of seeing how well the GS-DnCNN denoising and SSLCA worked in concert. To minimize power consumption and to maximize the quality of match between a radionuclide template and a measured gamma-ray spectrum, the Compton scatter predictions were not used in the SSLCA templates for these experiments. The GS-DnCNN model was trained to remove energy due to this phenomena and thus their addition would not assist in identification and could actually match with photopeaks from incorrect templates where leakage occurs in the denoising process. The removal of these predictions also results in sparse SSLCA dictionary elements for the radionuclide templates, which drastically lowers the overall required capacitance of the model's memristors and therefore power consumption.

I tested several of the top performing GS-DnCNN models trained during the tuning procedure using the SSLCA to classify radionuclides in the test set spectra after denoising. I discovered that the previously discussed best model, with 32 kernels per-convolution layer, was slightly overfitting to the validation set since it came from the same distribution as the training set, and therefore did not generalize as well to measured spectra as a narrower, 16 kernels per-convolution layer model. Additionally, I found a slightly higher L2-regularization of 1.3 along with no L1-regularization also resulted in slightly better generalization. The reduction in kernels per-layer significantly reduced the number of parameters in the model from 56,640 to 14,496, requiring only 28,992 memristors to build. However, despite a slight increase in L2-regularization, the removal of L1-regularization resulted in a higher overall magnitude for the model's weight values, which in turn would require a larger capacitance for many of the memristors. So despite a nearly $\frac{3}{4}$ reduction in the number of memristors, there is only a slight drop in the estimated power consumption from $\sim$2.03 μW to $\sim$1.9 μW. All results shown and discussed in this chapter use this smaller, more efficient GS-DnCNN model which is tested using the NaI measured gamma-ray spectra test set.

4.2 Test Set Identification Results

Fig. 4.1 shows the overall SSLCA accuracy for the test set spectra using no noise-reduction, background subtraction, and GS-DnCNN denoising. The accuracy is calculated by the total number of correct, rank-1 predictions, the radionuclide template that matches the spectrum the closest as determined by the SSLCA after applying its sparse coding optimization. With no noise-reduction, the SSLCA's identification accuracy was approximately 39.62%. Using GS-DnCNN denoising the accuracy increased by nearly 20% to 58.49%. Due to the knowledge of a close approximation of the background statistics, since the background is sampled from the same location prior to introducing the source radionuclide, the background subtraction performed approximately 11% better, 69.81%, than the GS-DnCNN which has no inherent knowledge of the background.

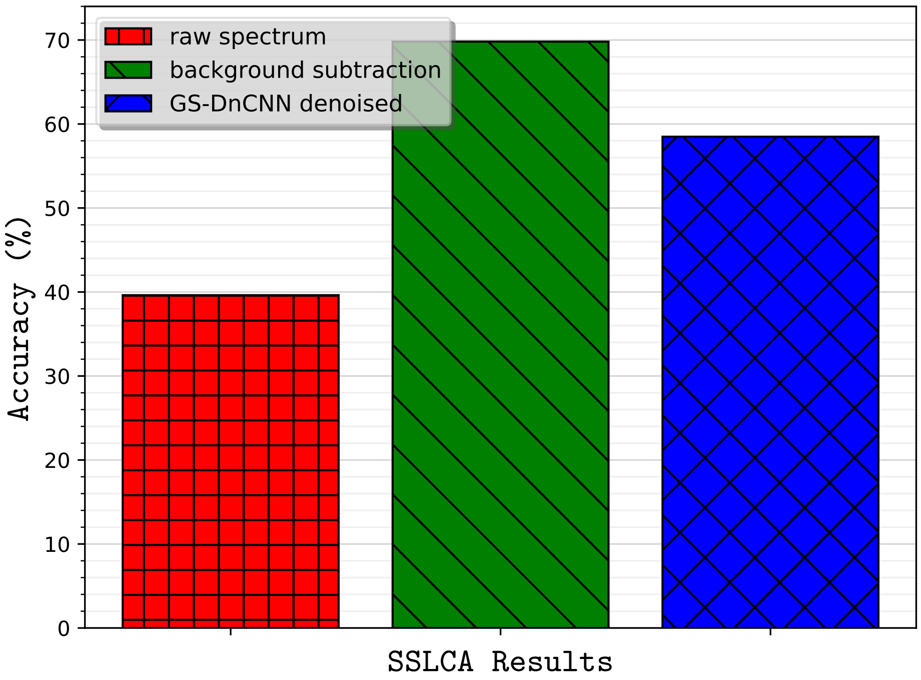

Figure 4.1: Comparison between SSLCA radionuclide identification results for spectra in the test set with no noise-reduction, background subtraction, and GS-DnCNN denoising. Identification accuracy increases by nearly 20% when the spectra are denoised using the GS-DnCNN. When the background statistics are known and subtracted from the spectra, identification accuracy is 11% better than when denoised with GS-DnCNN which does not require knowledge of the background.

4.3 Test Set Identification Results by Radionuclide

Fig. 4.2 shows a comparison of the SSLCA accuracy by radionuclide for the test set spectra using no noise-reduction, background subtraction, and GS-DnCNN denoising. For radionuclides ^{152}Eu, ^{235}U, and ^{57}Co, the performance using GS-DnCNN denoising is as good or better than when using background subtraction. For the other three its performance was worse, however, it was still as good or better than when no noise-reduction was used. This shows that despite any potential corruption or loss of information in the reconstruction of a spectrum output from the GS-DnCNN, there is no reduction in overall performance compared to the baseline of the raw, measured spectrum.

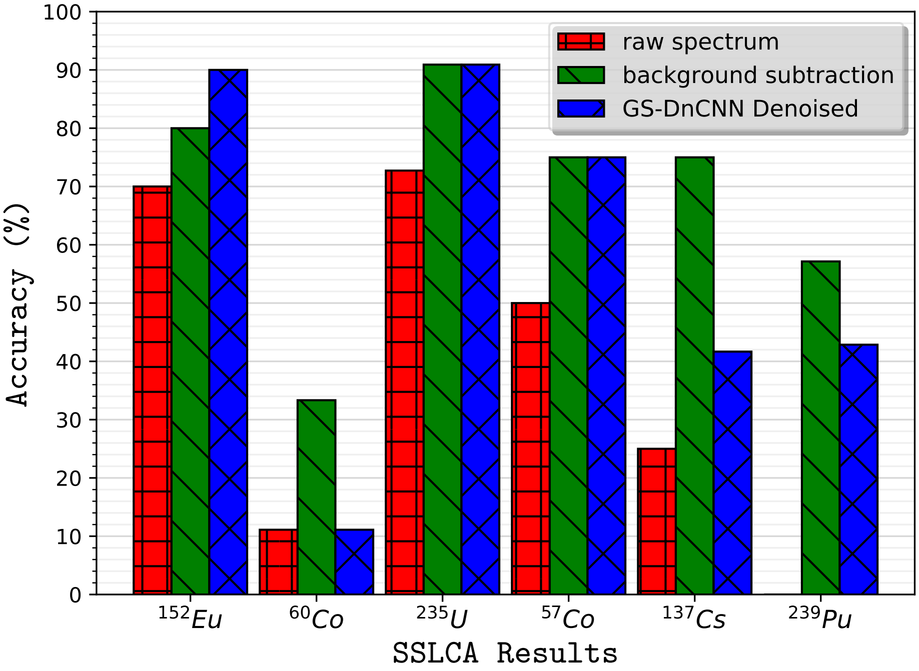

Figure 4.2: Comparison between SSLCA radionuclide identification results by radionuclide for spectra in the test set with no noise-reduction, background subtraction, and GS-DnCNN denoising. For half of the six radionuclides tested, SSLCA identifcation results are the same or better using GS-DnCNN denoising compared to background subtraction. For radionuclides that the GS-DnCNN performed worse than background subtraction, results are as good or better than no denoising.

4.4 Test Set Identification Results by Signal to Noise Ratio

Fig. 4.3 shows for each of the spectrum in the test set its SNR and whether or not the SSLCA resulted in a correct prediction. These results are also reported for no noise-reduction, background subtraction, and GS-DnCNN denoising. The SNR values are approximated by using the time-normalized background subtracted spectrum as the source and the time-normalized background as the noise. As can be seen from the figure there is a fairly uniform distribution of SNRs from 0 dB to -25 dB along with several between 0 dB and 25 dB. Unfortunately the SNRs are highly dependent on the nuclear material source intensity and decay rate and therefore the radionuclides are not uniformly distributed across the SNRs.

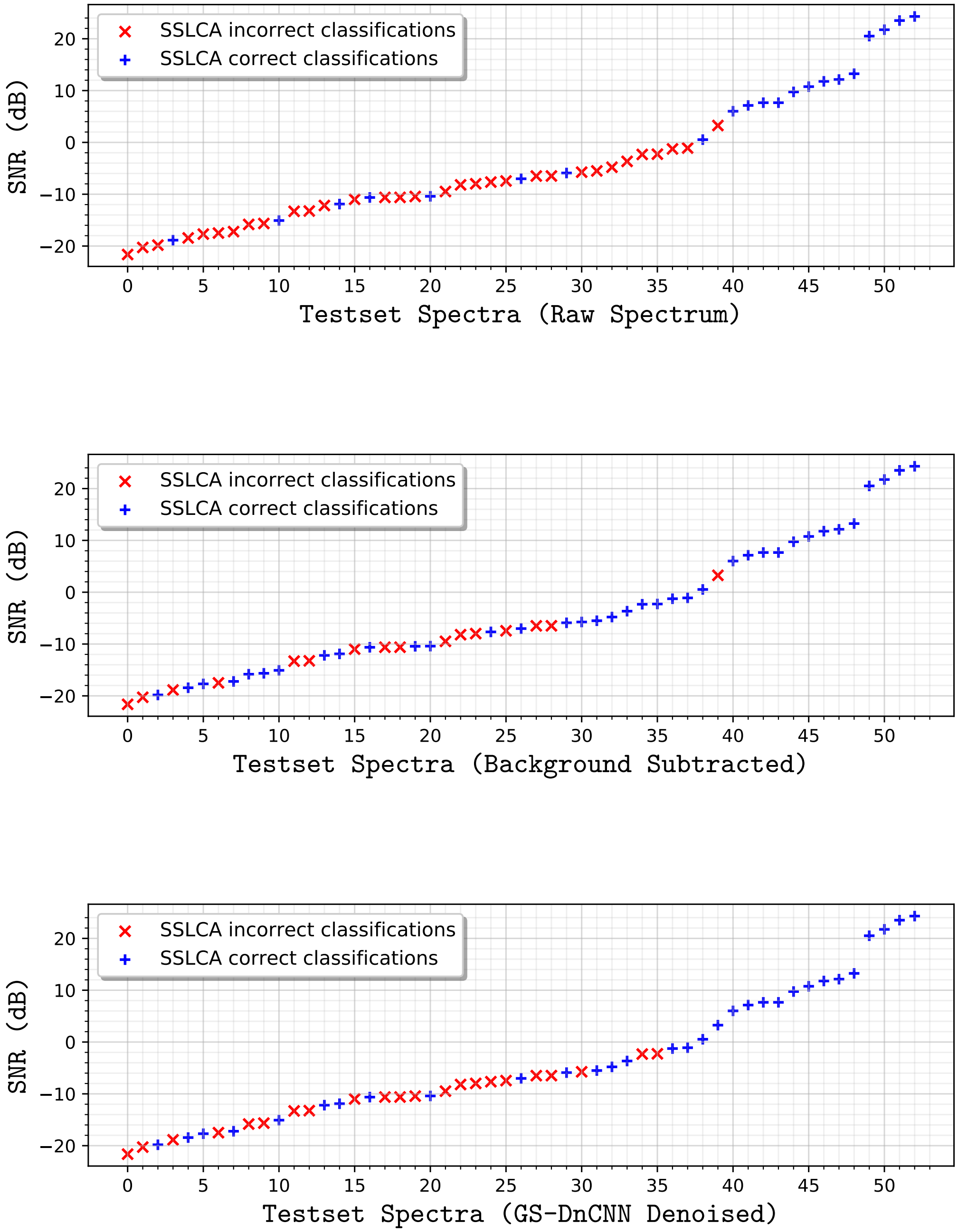

Figure 4.3: Correct and incorrect identifications by SNR for spectra in the test set with no noise-reduction, background subtraction, and GS-DnCNN denoising. The GS-DnCNN has a lower SNR tolerance for correctly identifying radionuclides compared to no-noise reduction, -2 dB vs. 3 dB respectively. Background subtraction performs better throughout the range of SNRs, however, the GS-DnCNN has a 24% increase in accuracy for classifications below 0 dB than prior to denoising.

Comparing the top graph of the figure (no noise-reduction) with the bottom graph of the figure (GS-DnCNN denoising), it can be seen that the first incorrectly classified spectrum has a significantly lower SNR when applying GS-DnCNN denoising compared with no noise-reduction. The SSLCA begins incorrectly classifying spectra without noise-reduction around 3 dB SNR, while with GS-DnCNN denosing the upper bound for incorrect identifications is slightly below -2 dB. Despite a single outlier, with background subtraction the upper bound for incorrect classifications is around -7 dB. Below these upper bounds the increase in incorrect classifications increases as the SNR levels decrease. This is most notable for the raw, no noise-reduced spectra which are almost entirely incorrect below the stated upper bound. The GS-DnCNN denoised spectra produce over twice the number of correct predictions below 0 dB than with no noise-reduction, increasing the accuracy in this range by 24%. However, it still incorrectly classifies more than it correctly classifies in the same region. The background subtraction on the other hand while still incorrectly classifying several below its upper bound, correctly identifies more. Both the background subtraction and GS-DnCNN denoising both result in the same lowest correctly classified spectrum with an SNR of -19.83 dB.

It is important to note that for all but one spectrum that was correctly classified with no noise-reduction, the GS-DnCNN denoised versions were also correctly classified, once again showing that despite any reconstruction error that results due to this denoising process the identification results are almost always superior to no noise-reduction.

4.5 Analysis of Test Set Results

Fig. 4.4 shows a visualization of the SSLCA's analysis of the lowest SNR spectrum (-19.83 dB) that resulted in a correct identification. While the SSLCA did make a rank-2 match with the correct ^{235}U template using the raw spectrum, it is not the top match and therefore is not counted as a correct identification for the accuracy results.

Using background subtraction the match increased by 17% compared to the raw spectrum, resulting in a top match and thus a correct identification. The background subtraction (middle graph), notably reduced the background compared to the raw spectrum (top graph), but a significant amount of noise is still present which can be seen comparing it with the ^{235}U template in the figure, the '-.-' hatched line (red). Due to this, much of the photoelectric is obscured by the relatively high noise floor, therefore a significant amount of the matching with the templates comes from background not source.

Using GS-DnCNN denoising the match was 33% higher than with the raw spectrum, also producing a correct identification. The GS-DnCNN denoised spectrum (bottom graph) resulted in a significantly larger suppression of background and noise, as noted by the majority of the spectrum having an intensity near or equal to zero. The reconstruction of the major ^{235}U photopeak at 185 keV fits nicely with the corresponding template. The smaller 89, 93, and 142 keV photopeaks are not reconstructed as well, but there is still enough area to assist in increasing the match by 18% compared to the background subtraction. Large clustering of background above the local average in the model's receptive field around 80 and 110 keV generated false peaks that result in small but notable rank-2 and rank-3 matches. However, the rank-1 match is 43% higher than the rank-2 match and therefore is clearly a much more confident prediction.

Fig. 4.5 shows a visualization of the SSLCA's analysis of the highest SNR spectrum

(-2.27 dB) that resulted in an incorrect identification using GS-DnCNN denoising. With the raw spectrum the SSLCA did not successfully identify the ^{137}Cs in any of the top 3 ranked matches. While the one and only notable photopeak for the radionuclide at 661 keV is clearly visible to the eye, the noise floor at lower energy levels is much larger and completely matches several photopeaks from other radionuclide templates that produce energy in that range.

Using background subtraction the lower energy background is reduced enough to make the 661 keV photopeak the dominant energy in the spectrum. Therefore, the SSLCA correctly identifies the ^{137}Cs as a rank-1 match. However, the noise floor at the lower energies is still relatively high resulting in rank-2 and rank-3 matches that are only 19% and 26% lower respectively.

The GS-DnCNN once again does a very good job squashing the noise floor to at or near zero through most of the spectrum. However, it does a poor job of reconstructing the 661 keV photopeak. Additionally, large false photopeaks are generated at 60 and 185 keV which partial match photopeaks in other templates. Therefore, the correct ^{137}Cs radionuclide is not present in the top 3 ranked matches.

Fig. 4.6 shows a visualization of the SSLCA's analysis of another ^{137}Cs spectrum that resulted in an incorrect identification for both the raw spectrum and the GS-DnCNN denoising, but correct using background subtraction. With GS-DnCNN denoising the 661 keV photopeak is reconstructed with a sufficiently large magnitude, however the width and height of the photopeak does not match well with the expected shape from the template and therefore only results in a rank-2 match of 39%. Due to several large spikes in background at lower energies, false peaks are generated at 70 and 185 keV that match well to photopeaks from other radionuclide templates. The top ranked ^{201}Ti has a decent but not high match of 53% while the correct ^{137}Cs match is only 14% lower. In this case the rank-1 match could represent a low confidence prediction and rank-2 and or rank-3 matches might be considered.

For the purpose of threat reduction it is more tolerable to allow false alarms, incorrectly predicting the presence of a dangerous material, vs. false rejections, not predicting a present dangerous material. Therefore, confidence levels could be determined from the magnitude of a match and the relative differences between subsequent matches, allowing for the use of rank-2 and rank-3 matches as alternate predictions when the top match is not large enough or is close to the rank-2 or 3 match. This type of prediction scheme could be used to correctly identify a spectrum like this where the rank-1 match is not sufficiently high and the correct match is not significantly below it, leading to a much lower false rejection rate and thus a higher probability of detecting and identifying clandestine material.

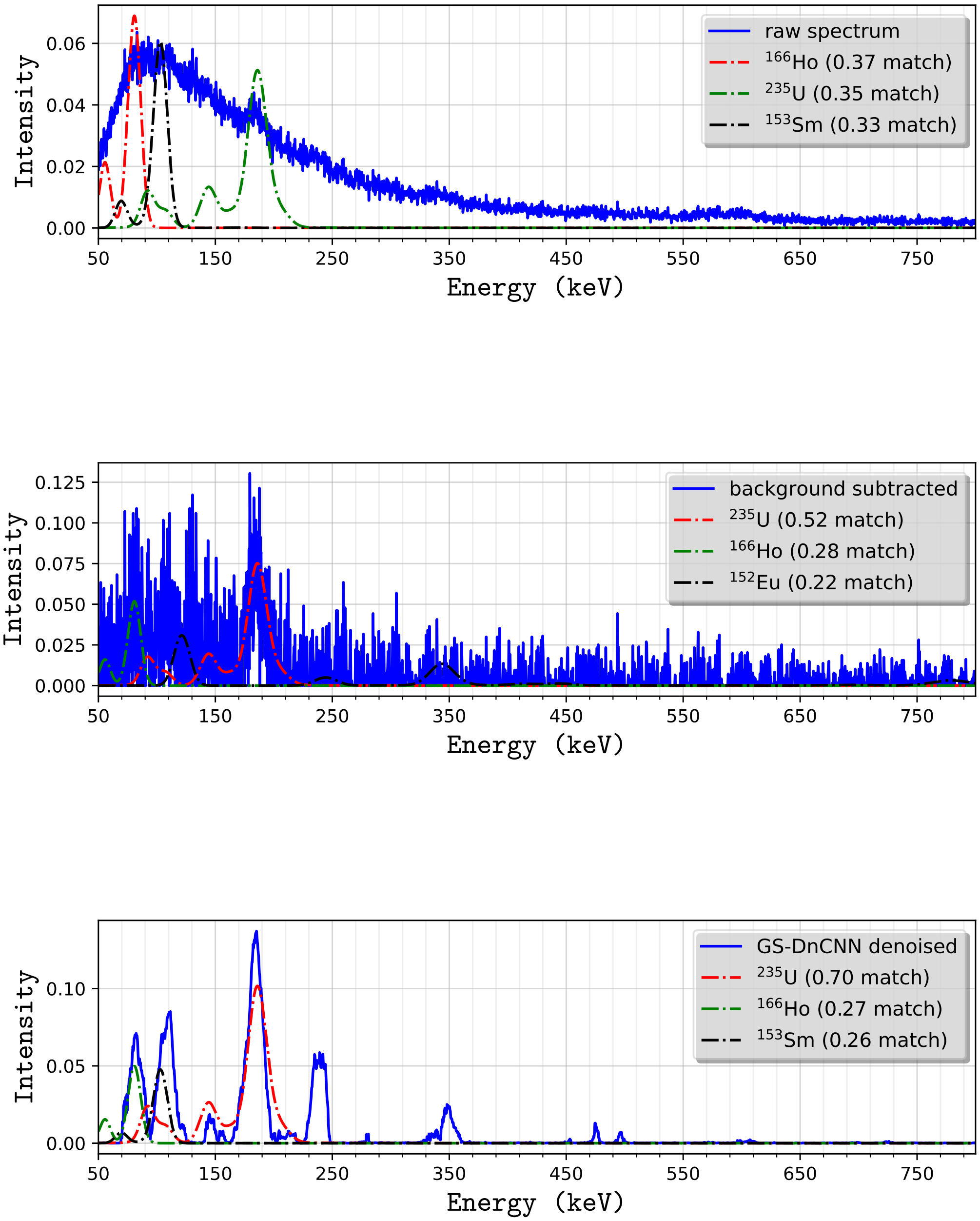

Figure 4.4: Visualization of the lowest SNR (-19.83 dB) spectrum which the SSLCA corectly identified using both background subtraction and GS-DnCNN denoising. After background subtraction the SSLCA determined a 52% match with the correct ^{235}U template. Using GS-DnCNN denoising the match was 18% higher compared with the background subtraction resulting from a good fit with the reconstruction of the prominent 185 keV photopeak and since most of the noise floor has been suppressed.

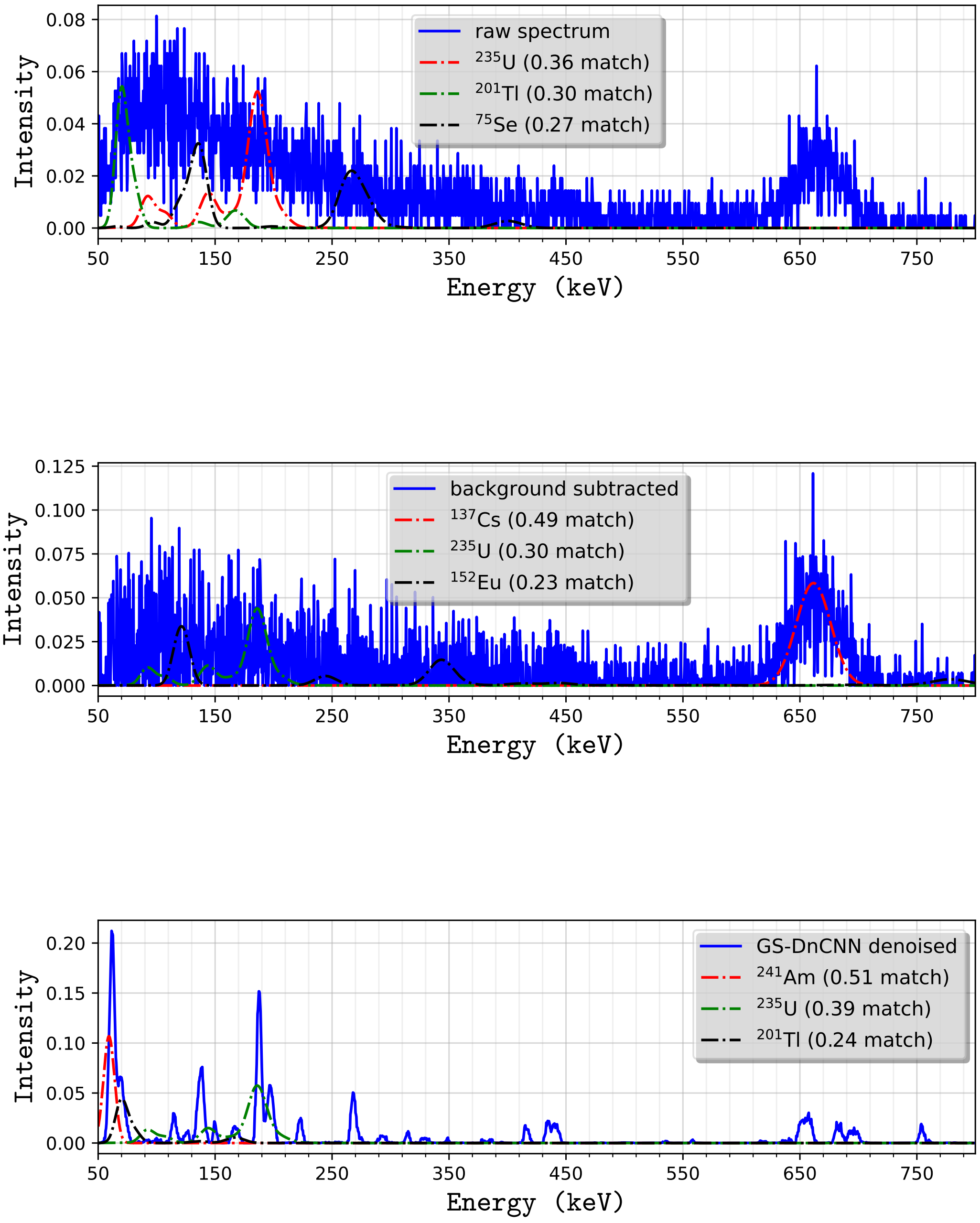

Figure 4.5: Visualization of the highest SNR (-2.27 dB) spectrum which the SSLCA correctly identified using background subtraction but incorrectly identified using GS-DnCNN denoising. With background subtraction lower energy level noise is reduced while the correct radionuclide's (^{137}Cs) main photopeak at 661 keV is not notably affected. After GS-DnCNN denoising the reconstruction of 661 keV photopeak is significantly smaller, resulting in an incorrect identification.

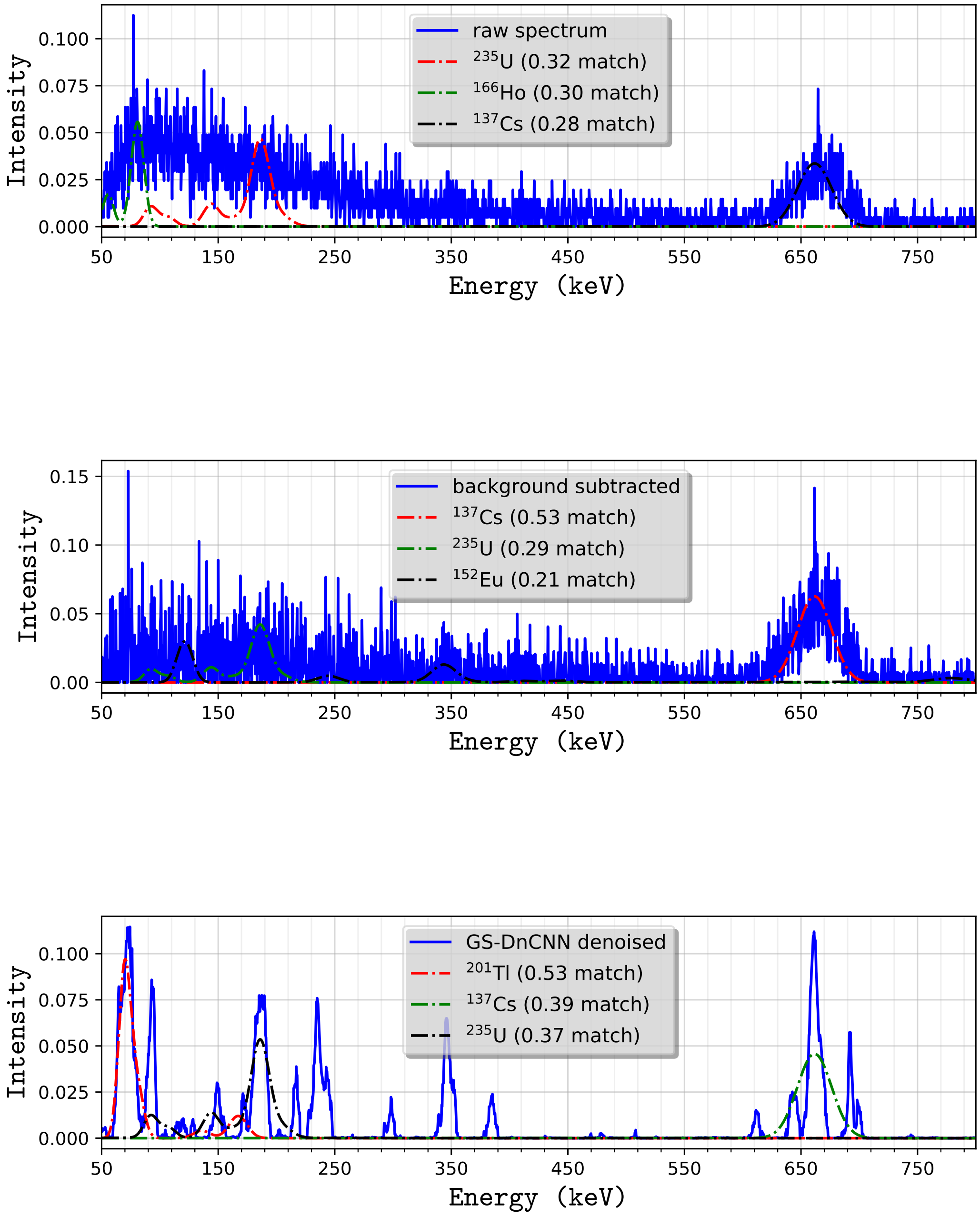

Figure 4.6: Visualization of a ^{137}Cs spectrum which the SSLCA correctly identified using background subtraction, but is not the top match using GS-DnCNN denoising. After GS-DnCNN denoising the reconstruction of the 661 keV photopeak has approximately the correct magnitude, but is about 10 keV narrower at FWHM compared to the template, resulting in a rank-2 identification.

Chapter 5

Conclusion & Future Work

5.1 Conclusions

First, this research has shown that the SSLCA achieves a near 89% accuracy on the radionuclide identification task using a test set of gamma-ray spectra measured with a low-resolution detector, within 2% of the performance achieved on the same test set using an industry standard, SOTA algorithm. Next, that the GS-DnCNN produces desirably sparse gamma-ray spectra due to the suppression of nearly all background radiation and detector noise, along with producing well defined photoelectric peaks. Finally, that the SSLCA's radionuclide identification accuracy increases by nearly 20% when using the two neuromorphic algorithms together compared with no noise-reduction.

It is estimated that the two neuromorphic algorithms together consume approximately 8 mW, over 99% less than the benchmark radionuclide identification algorithm GADRAS. Due to the GS-DnCNN's ability to suppress nearly all background and noise, not only does the input to the SSLCA become sparse, but the SSLCA's radionuclide dictionary elements can be sparsely represented since they no longer have to compensate for physical interactions due to non-photoelectric effect. This sparseness reduces power consumption due to less voltage being passed through the input of the SSLCA along with many of the dictionary elements' memristors not requiring a capacitance.

Clearly, in an ideal scenario, knowing the ground truth of the background statistics and performing background subtraction produces notably higher identification accuracies. However, background subtraction is not always a reliable or efficient process. As seen and discussed in Section 4.5, background subtraction does not suppress all noise since even if the background is static, collection is Poisson distributed and thus is different between each measurement. Due to this, gamma-ray spectra will not be sparse and therefore the use of background subtraction requires more power consumption than GS-DnCNN denoising for the SSLCA or any other radionuclide identification algorithm. Background radiation can also change over time, so an RID would have to continuously sample and average background and an anomaly detection algorithm would be required to determine when to stop sampling due to the presence of new, strong sources. Both of these are computationally expensive and power consuming tasks.

The GS-DnCNN allows for the denoising of static, measured spectrum from any location, without the need for a current background measurement of the area which may not be available. Additionally, since the purpose of these algorithms is to be deployed in an automated or handheld device, if entering a new location like a battle field or near a nuclear reactor where there is the potential for already existing dangerous levels of radioactivity, there is no way to sample the background and perform background subtraction.

The GS-DnCNN and SSLCA together have been shown to perform radionuclide identification well, but are far from the required SOTA needed to perform such a critical task as preventing a nuclear incident due to domestic or foreign terrorism. However, the algorithms have important properties, such as low power consumption, needed for long-term, independent monitoring, and the ability to increase the probability of identifications when background levels are unknown. Due to these high value properties, more research and further development of the GS-DnCNN and SSLCA could lead to improvements that would make them indispensable tools for radiation threat reduction.

5.2 Future Work

In the last few years there has been a significant amount of research devoted toward using unsupervised learning for generative and denoising tasks. Unsupervised learning uncovers high level structure in data and learns latent representations of it without an explicit objective. In 1985 Rumelhart *et al.* [38] introduced a novel NN architecture that performs this task exceptionally well which has now become known as an *Autoencoder* (AE).

The model consisted of N inputs units, $\log_2 N$ hidden units, and N output units. It was trained to learn bit encodings by objectively learning to reproduce the inputs at the output units. Due to the bottleneck created by the smaller number of hidden units, the dimensionality of the data is decreased, forcing the model to learn a code that can be encoded and decoded similar to a type of compression. When no non-liner activations are used for the units, the model performs a generalization of *Principal Component Analysis* (PCA), where a signal is decomposed into orthognal vectors ordered by direction of greatest variance. In 2006 Hinton *et al.* [39] showed that adding non-linearities to the model resulted in a generalization of non-linear PCA, and that using the equivalent number of hidden units as PCA components resulted in a significantly lower reconstruction error.

PCA is a traditional approach to signal denoising, but since gamma-ray spectra are implicitly one-dimensional, dimensionality reduction is not possible. However, AEs learn latent codes from the high level structure of data, discarding low level information like noise, and can be used to reconstruct sparse, yet informative sections of an input signal. Therefore, they are a natural choice for denoising signals of any dimensionality. Additionally, unsupervised learning does not require labeled data, and therefore synthetic data would not have to be used to train a model since clean targets are not required.

For future work I would like to investigate the use of unsupervised learning and AEs for the task of gamma-spectrum denoising. With the use NNs, a non-linear encoding of a gamma-ray spectrum's high level features (photopeaks) could be learned. Using a deep model, where the number of hidden units is gradually decreased and then increased back to the input size over several layers, a multiple step encoding and decoding can be learned. This deep representation would allow a model to have the capacity to learn a complex transformation that reconstructs the most informative sections of a noisy gamma-ray spectrum. By training an unsupervised model using measured gamma-ray spectra, latent features could be learned that are not reproducible through simulations, and therefore would have the potential to generate more accurate denoising results.

Bibliography

[1] W. Woods and C. Teuscher, “Fast and accurate sparse coding of visual stimuli with a simple, ultralow-energy spiking architecture,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 7, pp. 2173–2187, Jul. 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8541104

[2] L. Chua, “Memristor-the missing circuit element,” *IEEE Transactions on Circuit Theory*, vol. 18, no. 5, pp. 507–519, Sep 1971. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/1083337

[3] R. Marani, G. Gelao, and A. G. Perri, “A review on memristor applications,” *International Journal of Advances in Engineering & Technology*, vol. 8, no. 3, pp. 294–305, Jun 2015. [Online]. Available: https://arxiv.org/abs/1506.06899

[4] F. Cai, J. M. Correll, S. H. Lee, Y. Lim, V. Bothra, Z. Zhang, M. P. Flynn, and W. D. Lu, “A fully integrated reprogrammable memristor–cmos system for efficient multiply–accumulate operations,” *Nature Electronics*, vol. 2, no. 7, pp. 290–299, Jul 2019. [Online]. Available: https://www.nature.com/articles/s41928-019-0270-x

[5] A. Thomas, “Memristor-based neural networks,” *Journal of Physics D: Applied Physics*, vol. 46, no. 9, Feb 2013. [Online]. Available: https://iopscience.iop.org/article/10.1088/0022-3727/46/9/093001

[6] P. Yao, H. Wu, B. Gao, J. Tang, Q. Zhang, W. Zhang, J. Yang, and H. Qian, “Fully hardware-implemented memristor convolutional neural network,” *Nature*, vol. 577, no. 7, pp. 641–646, Jan 2020. [Online]. Available: https://www.nature.com/articles/s41586-020-1942-4

[7] L. Gerward. (2016, Mar) The discovery of gamma rays. [Online]. Available: https://web.archive.org/web/20050316204556/http://www.blis.canberra.edu.au/irps/Archives/vol14no1/Gerward.html

[8] M. Carson, W. Woods, S. Reynolds, M. Wetzel, A. J. Morton, A. A. Hecht, M. Osiński, and C. Teuscher, “Application of a simple, spiking, locally competitive algorithm to radionuclide identification,” *IEEE Transactions on*

Nuclear Science, vol. 68, no. 3, pp. 292–304, Mar 2021. [Online]. Available: https://ieeexplore.ieee.org/document/9335987

[9] C. J. Rozell, D. Johnsonn, R. Baraniuk, and B. A. Olshausen, "Sparse coding via thresholding and local competition in neural circuits," *Neural computation*, vol. 20, pp. 2526–2563, May 2008. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/18439138/

[10] S. H. Jo, T. Chang, I. Ebong, B. B. Bhadviya, P. Mazumder, and W. Lu, "Nanoscale memristor device as synapse in neuromorphic systems," *Nano Letters*, vol. 10, no. 4, pp. 1297–1301, Mar 2010. [Online]. Available: https://pubs.acs.org/doi/10.1021/nl904092h

[11] D. B. Strukov, G. S. Snider, D. R. Stewart, and R. S. Williams, "The missing memristor found," *Nature*, vol. 453, pp. 80–83, May 2008. [Online]. Available: https://www.nature.com/articles/nature06932

[12] National Nuclear Data Center. (2021) US Nuclear Data Program. [Online]. Available: https://www.nndc.bnl.gov/usndp/

[13] *American National Standard Performance Criteria for Handheld Instruments for the Detection and Identification of Radionuclides*, ANSI standard N42.34 Std., 2004.

[14] J. K. Tulli. (2019) Nuclear Wallet Cards. [Online]. Available: https://www.nndc.bnl.gov/wallet/

[15] J. E. Parks, "The Compton effect – Compton scattering and gamma ray spectroscopy," University of Tennessee, Tech. Rep., Jan 2015. [Online]. Available: http://www.phys.utk.edu/labs/modphys/Compton%20Scattering%20Experiment.pdf

[16] Intel. (2018) Intel Product Specifications. [Online]. Available: https://ark.intel.com/content/www/us/en/ark.html

[17] Microsoft. (2017) Visual Studio Performance Profiler. [Online]. Available: https://docs.microsoft.com/en-us/visualstudio/profiling/cpu-usage?view=vs-2017

[18] L. A. N. Laboratory. (2019) A General Monte Carlo N-Particle (MCNP) Transport Code. [Online]. Available: https://mcnp.lanl.gov/

[19] S. M. Horne, G. G. Thoreson, L. A. Theisen, D. J. Mitchell, L. Harding, and W. A. Amai. (2016, May) GADRAS-DRF 18.6 user's manual. [Online]. Available: https://prod-ng.sandia.gov/techlib-noauth/access-control.cgi/2016/164345.pdf

[20] V. Jain and S. Seung, "Natural image denoising with convolutional networks," in *Advances in Neural Information Processing Systems*, vol. 21, Dec 2009, pp. 769–776. [Online]. Available: https://proceedings.neurips.cc/paper/2008/file/c16a5320fa475530d9583c34fd356ef5-Paper.pdf

[21] D. Fleet and A. Jepson. (2011, Sep.) Markov random fields. [Online]. Available: http://www.cs.toronto.edu/~fleet/courses/2503/fall11/Handouts/mrf.pdf

[22] A. Choromanska, M. Henaff, M. Mathieu, G. B. Arous, and Y. LeCun, "The loss surfaces of multilayer networks," in *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics*, vol. 38, May 2015, pp. 192–204. [Online]. Available: http://proceedings.mlr.press/v38/choromanska15.html

[23] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising," *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3142–3155, July 2017. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/7839189

[24] A. Y. H. Andrew L. Maas and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Proceedings of the 30th International Conference on Machine Learning*, vol. 30, 2013. [Online]. Available: http://ai.stanford.edu/~amaas/papers/relu_hybrid_icml2013_final.pdf

[25] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, June 2016, pp. 770–778. [Online]. Available: https://www.computer.org/csdl/proceedings-article/cvpr/2016/8851a770/12OmNxvwoXv

[26] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proceedings of the 32nd International Conference on Machine Learning*, vol. 37, Jul 2015, pp. 448–456. [Online]. Available: http://proceedings.mlr.press/v37/ioffe15.html

[27] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proceedings of the 32nd International Conference on Machine Learning*, May 2015. [Online]. Available: https://arxiv.org/abs/1412.6980v5

[28] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *International Conference on Learning Representations*, May 2019. [Online]. Available: https://openreview.net/forum?id=Bkg6RiCqY7

[29] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proceedings of the Fourteenth International Conference on Artificial*

Intelligence and Statistics, vol. 15, Apr 2011, pp. 315–323. [Online]. Available: http://proceedings.mlr.press/v15/glorot11a.html

[30] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, vol. 9, May 2010, pp. 249–256. [Online]. Available: http://proceedings.mlr.press/v9/glorot10a.html

[31] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Dec 2015, pp. 1026–1034. [Online]. Available: https://openaccess.thecvf.com/content_iccv_2015/html/He_Delving_Deep_into_ICCV_2015_paper.html

[32] L. Gao, P.-Y. Chen, and S. Yu, "Demonstration of convolution kernel operation on resistive cross-point array," *IEEE Electron Device Letters*, vol. 37, no. 7, pp. 870–873, May 2016. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/7479502

[33] H. Ran, S. Wen, K. Shi, and T. Huang, "Stable and compact design of memristive GoogLeNet neural network," *Neurocomputing*, vol. 441, pp. 52–63, Jun 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231221002290

[34] A. Bala, X. Yang, A. Adeyemo, and A. Jabir, "A memristive activation circuit for deep learning neural networks," in *2018 8th International Symposium on Embedded Computing and System Design (ISED)*, Dec 2018, pp. 1–5. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8704116

[35] X. Yang, A. Adeyemo, A. Bala, and A. Jabir, "Novel techniques for memristive multifunction logic design," *Integration*, vol. 65, pp. 219–230, Mar 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167926016301766

[36] S.-Y. Sun, H. Xu, J. Li, H. Liu, and Q. Li, "Cascaded neural network for memristor based neuromorphic computing," in *2019 International Joint Conference on Neural Networks (IJCNN)*, Jul 2019, pp. 1–6. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8851836

[37] S.-Y. Sun, Z. Li, J. Li, H. Liu, H. Liu, and Q. Li, "A memristor-based convolutional neural network with full parallelization architecture," *IEICE Electronics Express*, vol. 16, no. 3, 2019. [Online]. Available: https://www.jstage.jst.go.jp/article/elex/16/3/16_16.20181034/_article

[38] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," in *Parallel Distributed Processing: Explorations*

in the Microstructure of Cognition, Vol. 1: Foundations. MIT Press, 1986, p. 318–362.

[39] G. E. Hinton and R. R. Salakhutdinov, “Reducing the dimensionality of data with neural networks,” *Science*, vol. 313, no. 5786, pp. 504–507, Jul 2006. [Online]. Available: https://science.sciencemag.org/content/313/5786/504.abstract

Appendix

A.1 Gamma-Spectrum Denoising Code Repository

All code relating to the gamma-ray spectrum denoising algorithm, the GS-DnCNN, and the GS-DnCNN AE is publicly available through Github under the MIT open source license at: https://github.com/mpc6/gamma-spectra_denoising. The simulated radionuclides and measured background are included which are required to build the training and validation sets. The NaI measured radionuclides are also included for testing purposes. The GS-DnCNN was developed using the PyTorch machine learning framework. The dataset generation, training, and testing scripts are fully parameterized with default values set to reproduce the results presented in this document. Script descriptions and usage are presented in the readme file associated with the repository.