Machine Learning Applied to Multi-Electron Events in Scintillator

Harrison LaBollita*[1,3], Nicholas Todoroff[2,3], Jane Kim[2,3], Yani Udiani[2,3], Robert Branson[2,3], and Morten Hjorth-Jensen[2,3]

[1]Department of Mathematics and Physics, Piedmont Collge, Demorest, GA
[2]Department of Physics and Astronomy, Michigan State University, East Lansing, MI
[3]National Superconducting Cyclotron Laboratory, East Lansing, MI

The corresponding GitHub repository for this work can be found here.

Contents

*hlabollita0219@lions.piedmont.edu

1 Introduction

The classical picture of spherical nuclei is far from the reality of the true nuclear structure. In fact, a variety of shapes can be observed in their ground state, such as, prolate, oblate, and triaxial. Shape coexistence is a nuclear phenomenon, where the nucleus exists in two stable shapes at the same excitation energy [2]. Nuclear properties, such as shape coexistence, are expected to vary significantly as a function of proton and neutron number (Z,N). Their properties provide unique information on the impetuses that foster changes to the nuclear structure of rare isotopes. In some neutron-rich nuclei, 0^+ states are predicted to exhibit shape coexistence.

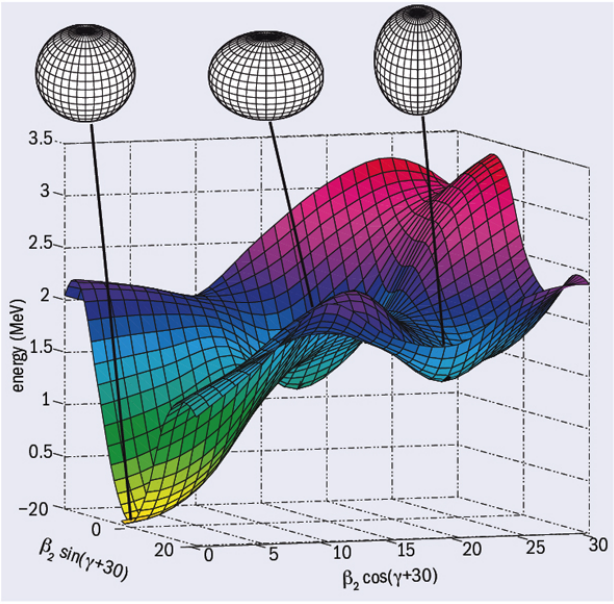

Figure 1: ^{186}Pb exhibits the nuclear property, shape existence, where a nucleus and have the same shape at similar energies

Therefore they are compelling to study, but experimentally challenging. At low energies, where the only energetically allowed decay mode is $0^+ \rightarrow 0^+$, conversion electron spectroscopy is the only viable technique to probe their properties. These shape-coexisting states are paramount in understanding changes to the nuclear structure of exotic nuclei [3].

Radioactive nuclei are produced and isolated at the National Superconducting Cyclotron Laboratory (NSCL) at Michigan State University. Sean Liddick's group focuses on characterizing transition rates of ground and excited states in nuclei as a function of proton and neutron number. The decay transition rates of the excited states can provide quintessential information of the coexisting structures. Sean Liddick's group employs conversion electron spectroscopy to study these transition rates. When a neutron-rich nucleus beta decays, a neutron transforms into a proton and emits an electron (β). The excited nucleus can then interact electromagnetically with the surrounding orbital electrons. This can result in the ejection of an internal conversion electron (e^-) from the atom [1]. Because this process is essentially simultaneous in time, it is pivotal to differentiate between the electron (β) emitted from the nucleus and the internal conversion electron (e^-) emitted from the atom.

Machine learning may offer a means to a solution for this problem. A subfield of artificial intelligence, machine learning is becoming ubiquitous in all fields of science. Due to the current information revolution, there has been an exponential increase in computational power. With this ability to effectively and efficiently apply new techniques to large datasets, machine learning has been blossoming [4]. Many researchers are finding it advantageous to employ these techniques to their own data analysis. Sean Liddick's group records a substantial amount of data making their experiment a potential candidate for machine learning techniques. This project attempts to use supervised machine learning algorithms as a means

to distinguish between one and two electron events and predict the electron's initial positions based on the energy depositions in a scintillator.

2 Electron Events

The scintillator detector (see Figure 2) has dimensions 4.8 cm × 4.8 cm × 0.3 cm. The detector is subdivided into a 16 × 16 grid of 256 pixels that are each 3 mm × 3 mm. For a given event, an electron(s) deposits energy into a pixel, the amount of energy deposited into the pixel is then recorded in an array.

Sean Liddick has provided us with two simulated datasets. The first is one million (10^6) one-electron events and the second is 218190 two-electron events. In the simulation, the electrons were uniformly generated from a 3 cm × 3 cm region centered on the detector. For each event i, we have the detector grid with all of the energy depositions in each pixel and the initial position(s) of the electron(s).

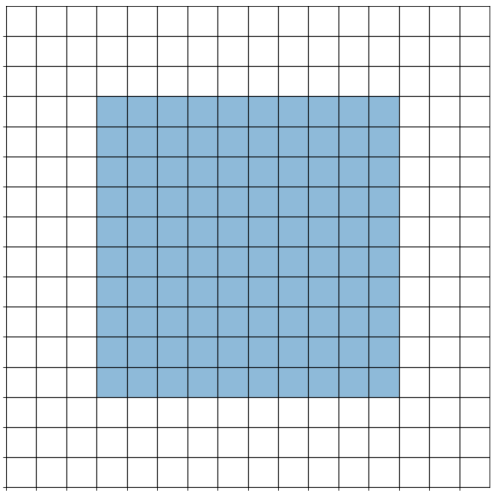

Figure 2: A model of the scintillator in Sean Liddick's experiment showing how it is divided into a grid. The blue region is where the electrons can start from.

In event i, the first 256 characters are the energies deposited in pixel j, where $j \in [0, 255]$. The pixel $j = 0$ is the bottom left corner of the pixel and the index runs across the rows, such that $j = 16$ is directly above $j = 0$. The next 6 characters are the initial energy of the electron(s) (E_0) and the initial position of the electron (x_0, y_0). These datasets will be used to construct training and testing datasets for our machine learning algorithms.

A visual representation of the kinds of events in the dataset are provided in Figure 3. These examples are of the scintillator with the electron's starting locations labelled. Once again, the task is to classify one and two electron events and predict the origin(s) of the electron(s).

To gain further insight into Sean Liddick's problem, we performed exploratory tests on the data. This would allow us to gain a better understanding of the problem and aid in deciding on what machine learning techniques we could implement. Beginning with the one-electron event dataset, we wanted to know where the electrons are starting from inside of each pixel and if there are any patterns associated with this. We found that 70 % of the time the electron starts in the pixel with the highest energy and 100 % of the time if the electron does not start in the pixel with the highest energy it started in one of the neighboring pixels. Therefore, we know that the probable area that the electron started in is the pixel with the highest energy deposited or one of its neighbors. We created a two-dimensional histogram to see if there was a favored location from within each pixel that the electron started from,

see Figure 4. Unfortunately, there was not, but this was still valuable insight necessary for our models. For the two-electron dataset, we noticed that there were essentially obvious two electron events and less obvious ones. So we wrote a script to see how many easy two electron events there were and how many hard ones there were (see Figure 6). We found that only 15% of the entire two-electron event was comprised of hard events.

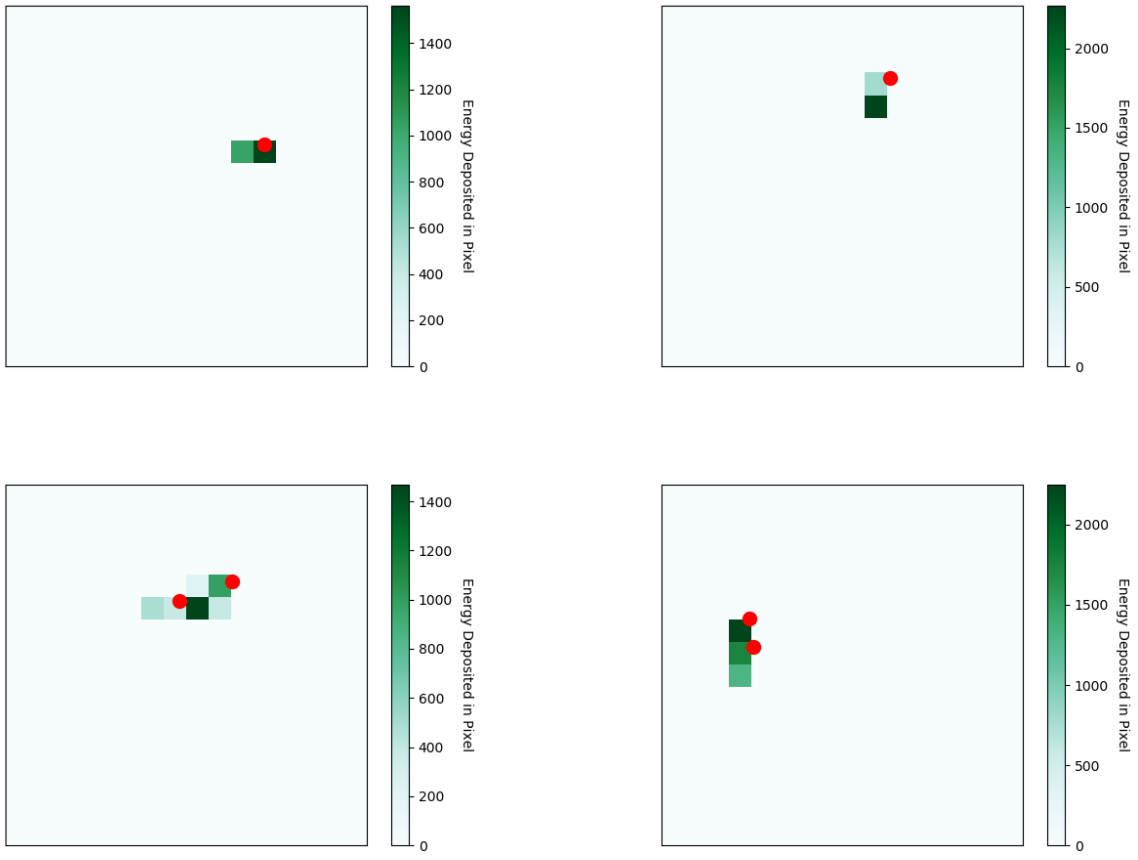

Figure 3: Top: One electron events. Bottom: Two electron events. Red dots indicate the electrons.

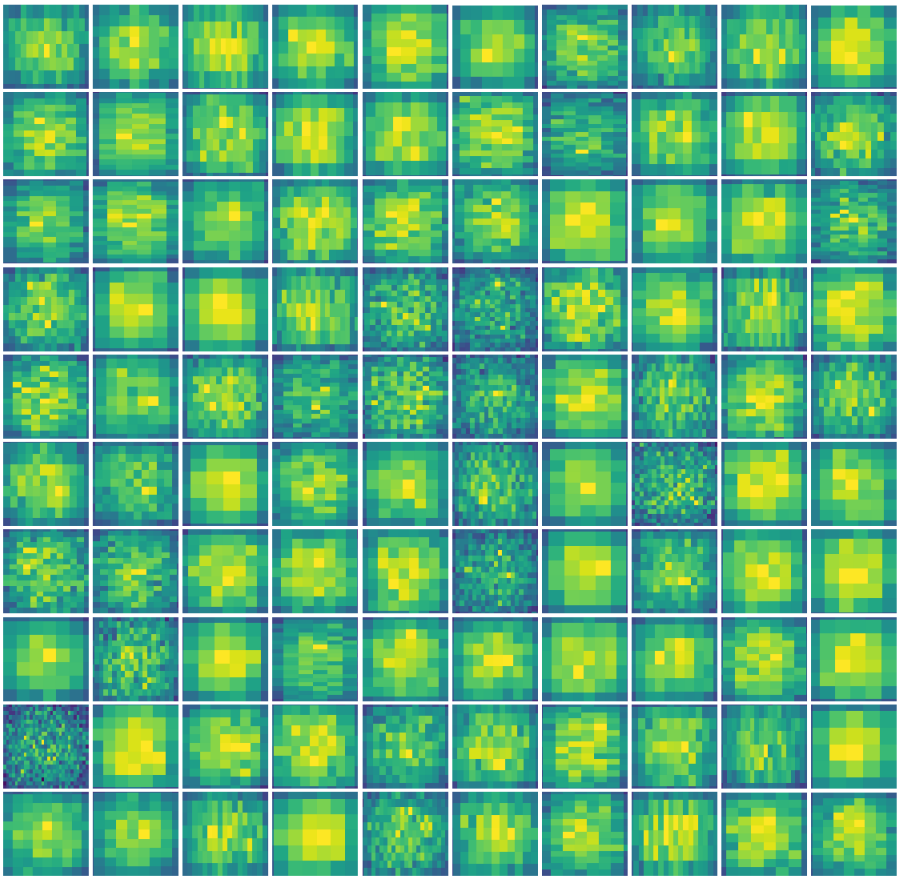

Figure 4: The 100 starting pixels. This 2-D histogram shows how dense the starting locations of the electrons are inside of each pixel.

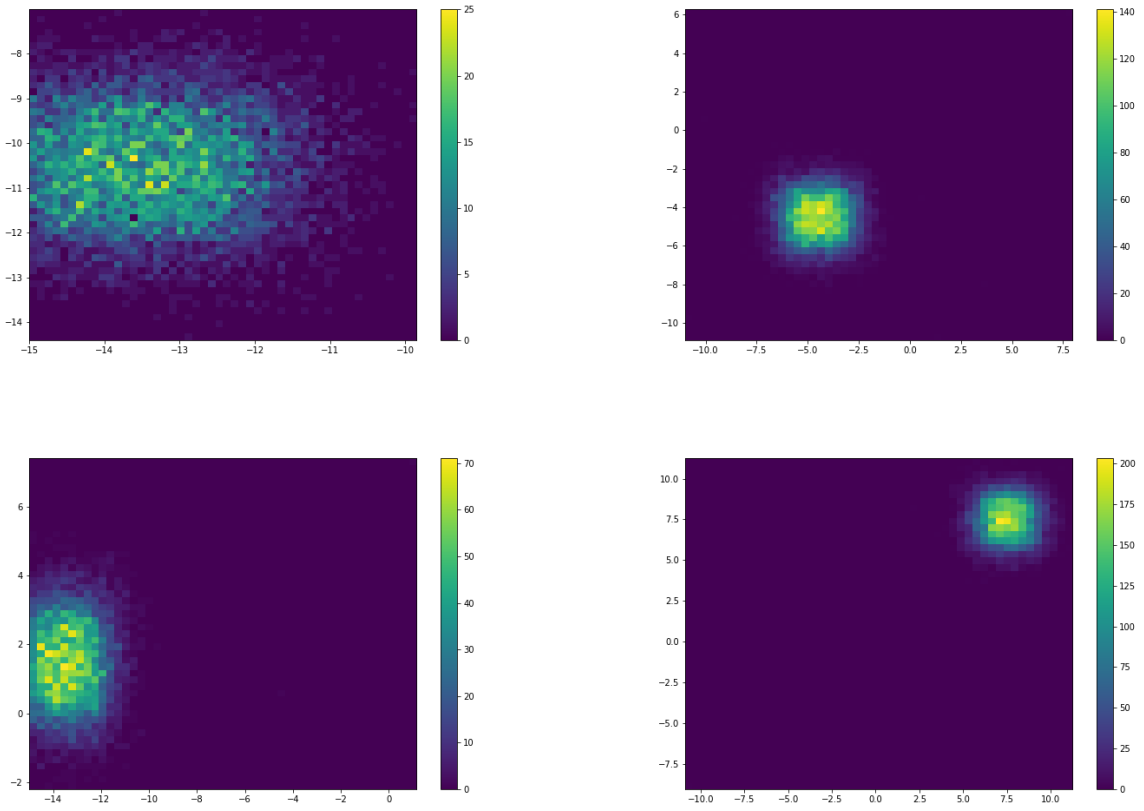

Figure 5: These are four of the individual histograms pulled from the figure above.

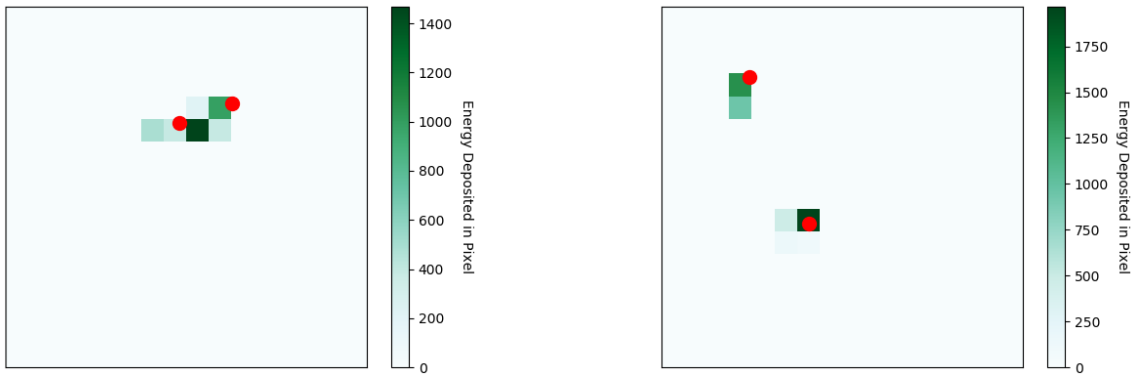

Figure 6: Left: "Hard" two electron event. Right: "Easy" two electron event.

3 Convolutional Neural Network

We chose to use convolutional neural networks to combat our problem. Convolutional neural networks (CNN) are a class of deep neural networks optimized for analyzing images. CNNs provide the computer with the ability to see. This will allow us to treat each scintillator event as a visual image, so the computer can see where the electron has been. For more information on convoluational neural networks, see [4].

3.1 Single-Electron Model

We created a CNN architecture in Keras[1] to predict the origin of the electron in single electron events. Figure 7 is a visual representation of all the layers in our model. Effectively, the model selects the pixel the electron started in, we assume it trains to select the pixel with the highest energy, then the model predicts where the electron started inside of that pixel. Our model was trained on one-third of the single-electron dataset and tested on a separate dataset of size 20000. We achieved a model accuracy of 96%.

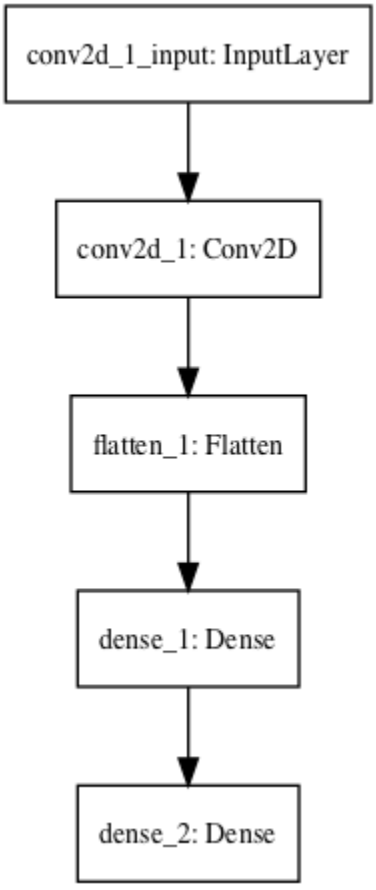

Figure 7: The schematic of all of the layers in the Single-Electron Model

[1] https://keras.io

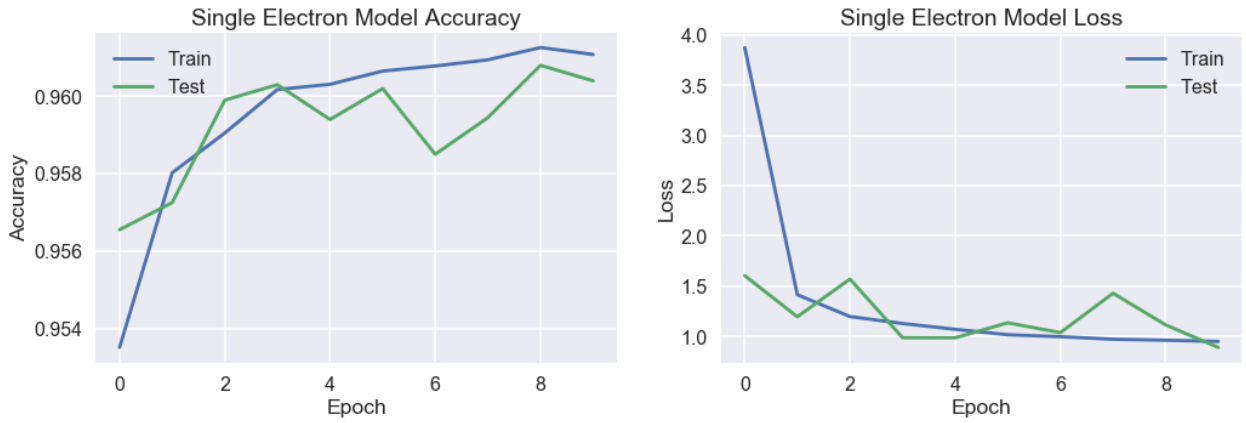

Figure 8: Performance history of the Single-Electron Model after ten epochs of training.

3.2 Multi-Event Model

The second CNN architecture was designed to correctly categorize a one and two electron event. Figure 10 is a representation of the layers in our architecture. We first needed to mix our two datasets together and then correctly label each event as a 0 for one electron event and 1 for two electron event. We wrote a simple script to perform this task. We created a new dataset of size 200000. We then trained the network on 150000 events and tested on the remaining 50000 events.

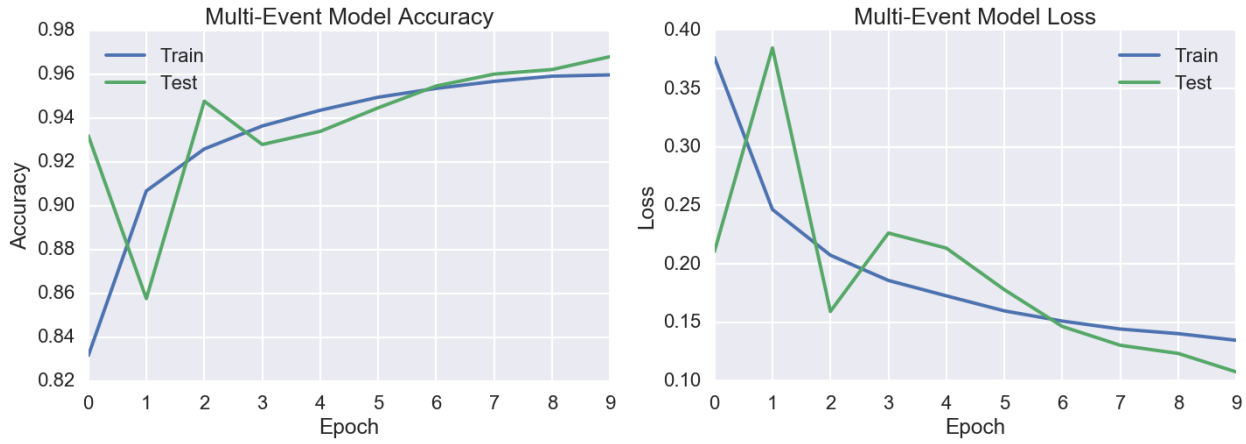

Figure 9: Performance history of the Multi-Event Model after ten epochs of training.

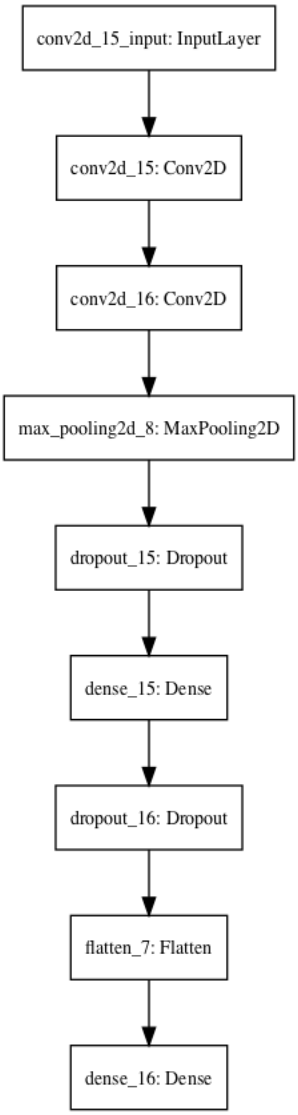

Figure 10: A schematic of all of the layers in our CNN for the Multi-Event Model.

4 Results and Discussion

The Single-Electron Model was designed to predict the origins of the electron in single electron events. After training, our model was 96% accurate on the testing set. When evaluating the distances between the predicted value $(\hat{x}, \hat{y})$ and the actual value (x_0, y_0), we found that on average our model predicted points about 1.1 mm away from the actual origin. This is promising considering the width of a pixel totals in 3 mm, so the majority of the time we are predicting points that are within the same pixel. However, we needed to benchmark the true performance of our model, so we compared it with a random guessing algorithm. Using the assumption that the electron starts in the pixel with the highest energy, (which is correct 70% of the time) we then randomly pick a point inside of that pixel. See Figures 13 and 12 to compare the CNN predictions versus our random guessing algorithm.

On average, the random guessing algorithm predicted points that were 1.5 mm away from the actual point. Our model's error was less than the random guessing algorithm's average 77% of the time, see Figure 12. However, this may not be an appreciable result, because randomly guessing is almost as advantageous as our model's performance, especially when considering that randomly guessing does no worse than guessing a point 3 mm away. Therefore it is imperative to define a proper error scheme and desirable resolution to better compare our neural network's performance. See Table 1 for more details.

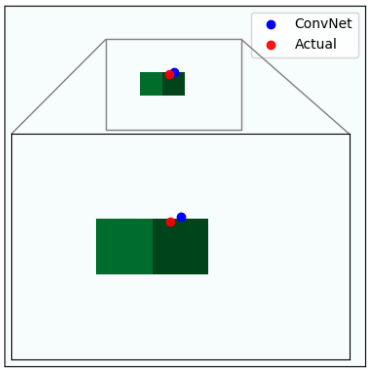

Figure 11: An example of testing our model's performance. The blue dot is our predicted origin of the electron, while the red dot is the actual origin of the electron.

The Multi-Event Model was designed to categorize one and two electron events based on the energy deposited in the scintillator. After training our model, we achieved a 96.79% accuracy. Our model was easily able to learn the difference between simulated one and two electron events.

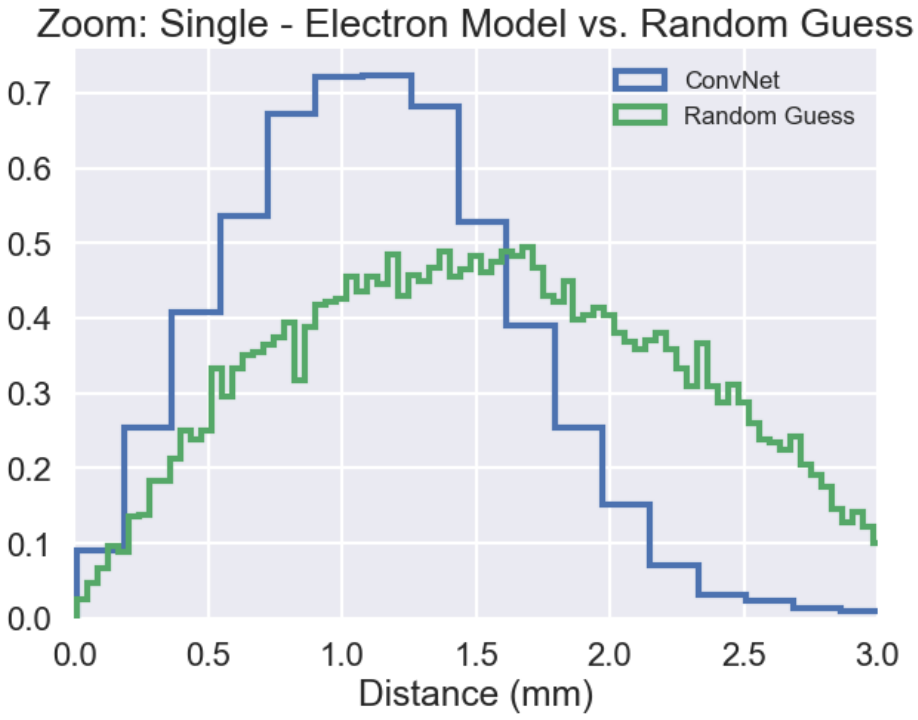

Figure 12: Restricting the x-axis from 0 to 3 mm gives a better visualization of our model's accuracy. Note that 77% of the area underneath the blue curve is to the left of the peak of the green curve.

Figure 13: Top: Normalized histogram of the distance error between our model's predicted values and the actual value. Bottom: Log-scale y-axis of the above plot showing that sometimes our model guesses wrong.

Metrics

Method	Average	Min	Max	90 %	95 %	99 %
Single Electron Model	1.2117	0.011	21.253	1.900	2.165	3.293
Random Guessing	1.555	0.008	3.944	2.574	2.805	3.218

Table 1: Metrics for Single-Electron Model and Random Guessing Algorthim. All metrics are units of mm.

5 Conclusion

With the implementation of machine learning techniques, we were able to successfully train a convolutional neural network (CNN) to distinguish between a one and two electron event. Furthermore, we successfully trained a CNN to predict the origin of the electron for one electron events. However, our model's performance was only marginally better than the random guessing. Therefore, a proper uncertainty quantification needs to be explored. This technique will be generalized to predict the origins of the electrons in the multi-event case and their respective initial energies. These models were trained and tested on simulated data provided by Sean Liddick, so they will need to be tested with a noisy dataset. Once these models are completely generalized, they can then be applied to real experimental data. If they perform well on the experimental data, then machine learning will be a viable data analysis technique for the Sean Liddick group and conversion electron spectroscopy in general.

References

[1] O. Dragoun. "Internal Conversion-Electron Spectroscopy". In: ed. by Peter W. Hawkes. Vol. 60. Advances in Electronics and Electron Physics. Academic Press, 1983, pp. 1–94. DOI: https://doi.org/10.1016/S0065-2539(08)60888-4. URL: http://www.sciencedirect.com/science/article/pii/S0065253908608884.

[2] R. Lucas. "Nuclear Shapes". In: (2001).

[3] A. Gade and S.N. Liddick. "Shape coexistence in neutron-rich nuclei". In: *Jounral of Physics G: Nuclear and Particle Physics* 43.2 (Jan. 2016).

[4] P. Mehta et al. "A high-bias, low-variance introduction to Machine Learning for physicists". In: *ArXiv e-prints* (Mar. 2018). arXiv: 1803.08823 [physics.comp-ph].