HHS Public Access

Author manuscript

Phys Med Biol. Author manuscript; available in PMC 2019 January 11.

Published in final edited form as:

Phys Med Biol. ; 63(2): 02LT01. doi:10.1088/1361-6560/aa9dc5.

Using convolutional neural networks to estimate time-of-flight from PET detector waveforms

Eric Berg[1] and **Simon R. Cherry**[1,2]

[1]Department of Biomedical Engineering, University of California-Davis, Davis, CA, USA

[2]Department of Radiology, University of California-Davis, Sacramento, CA, USA

Abstract

Although there have been impressive strides in detector development for time-of-flight positron emission tomography (PET), most detectors still make use of simple signal processing methods to extract the time-of-flight information from the detector signals. In most cases, the timing pick-off for each waveform is computed using leading edge discrimination or constant fraction discrimination, as these were historically easily implemented with analog pulse processing electronics. However, now with the availability of fast waveform digitizers, there is opportunity to make use of more of the timing information contained in the coincident detector waveforms with advanced signal processing techniques. Here we describe the application of deep convolutional neural networks (CNNs), a type of machine learning, to estimate time-of-flight directly from the pair of digitized detector waveforms for a coincident event. One of the key features of this approach is the simplicity in obtaining ground-truth-labeled data needed to train the CNN: the true time-of-flight is determined from the difference in path length between the positron emission and each of the coincident detectors, which can be easily controlled experimentally. The experimental setup used here made use of two photomultiplier tube-based scintillation detectors, and a point source, stepped in 5 mm increments over a 15 cm range between the two detectors. The detector waveforms were digitized at 10 GS/s using a bench-top oscilloscope. The results shown here demonstrate that CNN-based time-of-flight estimation improves timing resolution by 20% compared to leading edge discrimination (231 ps vs. 185 ps), and 23% compared to constant fraction discrimination (242 ps vs 185 ps). By comparing several different CNN architectures, we also showed that CNN depth (number of convolutional and fully connected layers) had the largest impact on timing resolution, while the exact network parameters, such as convolutional filter size and number of feature maps, had only a minor influence.

1. Introduction

Improving timing resolution in scintillation detectors used for time-of-flight positron emission tomography (PET) has attracted research interest due to its potential to increase image quality, quantification, and lesion detection (Karp *et al* 2008, Conti 2011, Vandenberghe *et al* 2016). In scintillation detectors used for time-of-flight PET, the fast timing information is contained in the first 1 – 2 ns of the detector waveforms, the shape of which is influenced by several physical factors including the early optical photon flux (determined by the scintillator emission, deposited energy, and photon transport in the crystal), the photodetector response, and the response of other front-end electronics

(amplifiers). Therefore, most recent improvements in timing resolution have been a result of developing scintillators with higher luminosity and faster emission (both rise and decay times) (Surti *et al* 2002, Spurrier *et al* 2008), optimizing the photon transport and extraction from the crystal (Spanoudaki and Levin 2011, Auffray *et al* 2013, Lecoq *et al* 2013, Berg *et al* 2015), developing photodetectors with higher photon detection efficiency, lower noise, and reduced time jitter (Schaart *et al* 2010, Spanoudaki and Levin 2010, Seifert *et al* 2013, Nemallapudi *et al* 2015), and developing fast and low noise electronics (Anghinolfi *et al* 2004, Rolo *et al* 2013).

However, most PET detectors still make use of rather simple signal processing techniques to estimate the time-of-flight, normally either leading edge discrimination or constant fraction discrimination (Surti and Karp 2016). Now, with readily available hardware for fast waveform digitization (> 1 GS/s), (e.g. domino ring sampling (Ritt 2008, Ashmanskas *et al* 2014)), there is opportunity to develop alternative methods for estimating the time-of-flight from the digitized waveforms using advanced signal processing techniques, and ultimately improve timing resolution. There have been a few developments in signal processing for time-of-flight PET using digitized waveforms, including fitting the digitized samples with cubic spline functions to reconstruct the waveforms (Vinke *et al* 2010), using multiple thresholds along the rising edge to accurately correct for time-walk with leading edge discrimination (Xie *et al* 2009), using a time-over-threshold algorithm to correct for time-walk (Rolo *et al* 2013), or deconvolving the photodetector's response from the rising edge (Berg *et al* 2016). However, these methods still make use of the conventional leading edge or constant fraction discrimination methods on the digitized waveforms. Additionally, some attempts have been made at developing statistical-based timing estimation methods, specifically maximum likelihood estimation, by modeling digitized waveform samples as random Gaussian variables (Barrett *et al* 2009, Ruiz-Gonzalez *et al* 2017) or by modeling the time stamps (obtained with leading edge discrimination) from multiple photodetectors as random variables (Van Dam *et al* 2013). Lastly, Moskal *et al* (2015) have developed a method to estimate time-of-flight from detector waveforms by computing the similarity of a "test" waveform to a library of reference waveforms with known time-of-flights.

We propose to make use of deep convolutional neural networks (CNNs) to estimate PET time-of-flight from the coincident waveforms. Briefly, deep learning with CNNs is a branch of machine learning that uses a stack of convolutional and fully connected layers to learn features that are characteristic of the known input (e.g. image array or data sequence from speech). A detailed description and review of CNNs is not included here, rather we refer to Lecun *et al* (2015). Recently, deep CNNs have achieved state-of-the-art performance in many areas, and are now the dominant method used for image recognition (Krizhevsky *et al* 2012, Simonyan and Zisserman 2014, Szegedy *et al* 2015). Here, we make use of CNNs such that the input to the CNN is the time-varying sequence of waveform samples from the pair of detectors for a coincidence event, and the CNN output is the estimated time-of-flight for the event.

One of the main attractions of deep convolutional neural networks is their ability to learn complex representations of the input data with little human engineering (Lecun *et al* 2015). In particular, deep CNNs can be sensitive to small changes or patterns in the input data,

while being relatively insensitive to irrelevant changes. Therefore, CNNs appear as a promising candidate to make use of all the information contained in the rising edge of the waveform and finding patterns in complex relationships between event timing and the various physical factors, without the need to explicitly model these physical processes in the neural network.

One potential pitfall associated with machine learning techniques, specifically for supervised learning methods, is the requirement for ground-truth-labeled training data. However, one fortunate aspect of using machine learning techniques for time-of-flight PET is that it is theoretically very easy to obtain ground-truth-labeled training data: since the two 511 keV annihilation photons are produced simultaneously, the ground-truth time-of-flight difference of the photons to the entrance face of the detectors is determined only by the distance between the positron emitting source and the coincident detectors, and these distances can be easily controlled in both benchtop and scanner settings. For instance, in this proof-of-concept study, we obtain ground-truth-labeled time-of-flight training data by stepping a point source between a pair of detectors.

This proof-of-concept study focuses mainly on evaluating the use of convolutional neural networks to improve timing resolution for time-of-flight PET, and is not specifically aimed at improving upon the current state-of-the-art timing resolution. Therefore, in this paper we use a rather simple detector and focus on assessing the impact of differences in the design of the convolutional neural networks, such as network depth. We first describe the experiment, describe the configurations of the CNNs and their training, then compare timing resolution obtained with CNNs vs. conventional timing discrimination methods (leading edge and constant fraction discrimination).

2. Materials and Methods

2.1. Experiments

Coincidence waveforms were acquired using two identical detectors (Figure 1), each consisting of a 5 × 5 × 10 mm lutetium fine silicate (LFS) crystal wrapped with 5 layers of Teflon tape and coupled to a single channel photomultiplier tube (PMT, Hamamatsu R12844) using pressure sensitive optical adhesive (3M 8194). The face-to-face distance between detectors was ~ 40 cm. A 100 kBq ^{68}Ge point source was stepped between the detectors at 5 mm increments over a 15 cm range (−7.5 cm to +7.5 cm from the mid-point between detectors), so that a total of 29 datasets were acquired. The PMT waveforms were digitized at 10 GS/s using a bench-top oscilloscope (Tektronix DPO7254), and approximately 15,000 coincidence waveforms were acquired for each source position.

2.2. Signal Processing

The waveforms acquired from the oscilloscope (Figure 2a) were first baseline corrected and integrated over 175 ns to estimate the energy deposited by each event. A 430 – 590 keV energy window (extracted from energy histograms) and 5 ns coincidence timing window was applied to all waveform pairs. Next, each waveform pair was cropped to capture only the first 3.5 ns of the waveforms (Figure 2b). The start of the cropped waveform pairs was

determined as 1.5 ns prior to the first sample at which one of the waveforms crosses a 150 mV threshold, whichever is earliest. This cropping ensures that all waveform pairs contain the full rising edge of both waveforms, as well as ~ 1 ns of baseline before the samples containing the early rising edge. Lastly, for use in the CNNs, the waveform pairs were stored as two-dimensional arrays, and labeled with the ground-truth time-of-flight difference based on the distance of the point source relative to each detector (Figure 2c). Each array dimension was 2 × 35 × 1, where the rows correspond to the waveforms from each detector, the columns correspond to the digitized waveform samples (0.1 ns sampling), and stored in grayscale with pixel values in volts (approximately 0 to 0.5 V range).

Time-of-flight estimation using digital leading edge discrimination and constant fraction discrimination were performed using the cropped waveform pairs. Leading edge discrimination was optimized by varying the leading edge threshold over a range of 10 – 100 mV, and time-walk correction of the leading edge timing pick-off values was performed using a linear correlation between the estimated energy values and the measured time-of-flight. For constant fraction discrimination, the delay, attenuation factor, and threshold were also simultaneously optimized. Linear interpolation was used to estimate waveform values between samples.

2.3. CNN Architectures

The CNNs used here consist of several layers, including the input layer, convolutional layers, fully connected layers, and a regression layer. Several configurations for the convolutional layers were compared; these are summarized in Table 1. All convolutional layers used a stride of 1 pixel. The fully connected layers were fixed for all networks: the first fully connected layer had 256 outputs, while the second fully connected layer had one output. The filter weights were initialized with Gaussian randomness (standard deviation of 0.01) and a bias of 0.7. For the 7-layer CNNs, the first four convolutional layers were initialized using the trained 6-layer CNNs to enable practical training time (Simonyan and Zisserman 2014). A dropout layer with 50% dropout probability is included after the first fully connected layer to reduce over-fitting of the training waveforms. The convolutional layers were followed by rectifying linear units (ReLU, not shown in Table 1).

The set of CNN architectures compared here is clearly not an exhaustive analysis of the infinite parameter space (number of layers, filter size, number of feature maps, etc), instead we focus mainly on investigating the impact of the CNN depth. Aside from the network depth, two configurations for the convolutional filters were compared, referred to as “fixed” and “tapered”. The fixed configuration uses small convolutional filters (5 pixel width) for all layers. The tapered CNNs use large filter width (11 pixels) for the first layer, intermediate filter width (7 pixels) for the second layer, and small filters (5 pixels) for subsequent layers. The fixed configuration is loosely adapted from the architecture of one of the top performing deep CNNs (VGG16 (Simonyan and Zisserman 2014)), while the tapered architecture follows the more traditional CNN design (Krizhevsky *et al* 2012).

2.4. Training the CNNs

Each CNN was trained in three realizations using 145,000 randomly chosen waveform pairs for each realization (5000 events from each of the 29 source positions). The order of the training waveform pairs was randomized before training. The mini-batch size used in the stochastic gradient descent was 150 waveform pairs. The initial learning rate was 10^{-4}, and stepped down to 10^{-5} and 10^{-6} at 30% and 60% of the maximum number of epochs, respectively. Training was performed using MATLAB 2017a Neural Network Toolbox and a GTX 1070 GPU. The only difference in training the CNNs described in Table 1 was the maximum number of epochs to ensure convergence (e.g. 200 epochs were used for the 3-layer networks, while 1000 epochs were used for the 6-layer networks). Additionally, the sensitivity of CNN performance with regards to the size of the training dataset was examined by training the 5-layer tapered CNN using 100 – 5000 training waveforms from each source position.

2.5. Testing the CNNs

A set of 87,000 waveform pairs (3000 from each source position) were used to evaluate the trained CNNs. The test waveform pairs were chosen randomly, with no overlap with the training waveforms for each realization. The test waveform pairs were passed through the trained CNNs to predict the time-of-flight for each waveform pair.

3. Results

Figure 3a shows the timing resolution obtained with leading edge discrimination, constant fraction discrimination, and each of the CNNs. The results shown are the average for the central five source positions and three independent realizations for all cases (average timing resolution for each of the five source positions individually are provided in Supplemental Figure 1). Coincidence timing resolution was computed as the FWHM obtained from Gaussian fits to histograms of the estimated time-of-flight values. A comparison of timing spectra obtained from the central source position are provided in Figure 2b for all three timing methods.

The best timing resolution was obtained with the tapered 6-layer CNN (185 +/− 2 ps), representing a 20% improvement compared to leading edge discrimination (231 +/− 3 ps), and a 23% improvement compared to constant fraction discrimination (242 +/− 4 ps). The depth of the CNNs had the largest influence on timing resolution, with timing resolution ranging from ~ 220 ps with 3-layer CNNs, to ~ 185 ps with 6-layer CNNs. In general, there was no significant difference between the tapered networks vs. the fixed networks. Similarly, doubling the number of feature maps in each convolutional layer with the 6-layer network did not change the resulting timing resolution. The effect of training dataset size on timing resolution with the 5-layer tapered network is shown in Supplemental Figure 2: the timing resolution is stable (i.e. within 5% difference) when at least 1000 waveforms from each source position were used.

We also investigated the offset in the timing spectra for different source positions. Figure 4 shows the timing spectra obtained with the 6-layer tapered CNN for the central five source

positions. The peak positions of the 5 histograms are provided on the figure, demonstrating time-of-flight offsets between neighboring source locations within 31 - 35 ps, in close agreement with the expected 33.3 ps offset.

4. Discussion

Here we demonstrated the applicability of using convolutional neural networks, a technique currently capable of achieving state-of-the-art performance in image recognition, for processing waveforms for time-of-flight PET. Aside from demonstrating the proof-of-concept of using CNNs for estimating time-of-flight, we also compared the impact of several CNN parameters on timing resolution. The only significant factor found here was network depth, specifically the number of convolutional layers; in general, a larger number of convolutional layers led to improved timing resolution. The exception were the 7-layer CNNs, which performed slightly poorer than the 6-layer CNNs, possibly as a result of over-fitting the training data. Timing resolution was largely insensitive to the exact network parameters, such as filter size and the number of feature maps. Of course, the practical trade-off with increasing the network depth is longer training time, for example, training the 3-layer networks required ~ 30 minutes, whereas the 6-layer networks required ~ 6 – 8 hours. In general, training time, defined as the number of iterations to reach convergence, increases approximately quadratically with the number of layers and feature maps, but does not strongly depend on the size of the training dataset.

Compared to CNN-based time-of-flight estimation, only maximum likelihood methods provide similar improvements in timing resolution compared to conventional methods. For example, Van Dam *et al* (2013) demonstrated a ~20% improvement in timing resolution using time stamps from multiple SiPM sensors in a maximum likelihood algorithm compared to using only the earliest time stamp. The study by Ruiz-Gonzalez *et al* (2017) involving maximum likelihood timing estimation from the digitized waveforms also showed improved timing resolution compared to leading edge and constant fraction discrimination, but only when using relatively slow sampling rates (< 1 GS/s) where the overall timing resolution was relatively poor.

Given the somewhat “black-box” nature of CNNs, it is difficult to provide exact conclusions on the origin of the improvement in timing resolution. However, the following describe some of the possible effects contributing to the success of CNNs for time-of-flight estimation. First, one of the major contributors may be the ability of the CNNs to essentially deconvolve the photodetector response. For timing estimation, the photodetector pulse shape is an irrelevant parameter (does not contain timing information), but it does influence the timing pick-off. CNNs are likely able to find these patterns and essentially deconvolve the photodetector pulse shape, whereas timing estimation with leading edge and constant fraction discrimination may be more strongly influenced by the photodetector pulse shape. However, we showed in previous work that deconvolving the photodetector response provided ~ 10% improvement in timing resolution with leading edge discrimination (Berg *et al* 2016), and so it is unlikely that this effect is the sole contributor to the observed improvement. We will investigate this hypothesis by comparing CNN-based timing with photodetectors characterized by different response functions, and in simulations where the

photodetector response can be easily modified. Second, the CNNs may be well suited to find patterns in the waveforms that are related to depth-of-interaction effects, such as changes in photon transit time and differences in overall light collection, that are not fully accounted for with conventional timing discrimination. In this work, we used a 10 mm length crystal; short enough so that these effects wouldn't dominate the overall timing resolution, but long enough so not to completely eliminate depth-of-interaction effects. Experiments with crystals of different lengths, including depth-of-interaction encoding detectors, will be a subject of future studies to investigate this hypothesis. Lastly, the convolutional nature of CNNs may provide a robust solution to use information at the very start of the rising edge. Whereas leading edge discrimination requires a sufficiently high threshold as to not trigger on the noise baseline, the convolutional filters may be able to use some of this early information without being dominated by the noisy baseline. The contribution of this effect will be examined by comparing the performance of leading edge and CNN-based time-of-flight estimation over a range of signal-to-noise levels, both electronic noise as well as noise resulting from lower collection of scintillation photons.

The primary future application of this method is not yet obvious from this proof-of-concept study. Possibly, estimating time-of-flight with CNNs may remove the need for dedicated timing ASICs to obtain the timing pick-offs for each event. By using the raw waveforms in the CNNs to estimate time-of-flight, we do not require a precise time-to-digital converter, only a relative small number of digitized waveform values along the rising edge. Another benefit of this approach is that minimal pre-processing is required: here the only pre-processing is to baseline correct the waveforms, combine the pair of waveforms for each coincident event into a two-dimensional array, and crop the waveform pair to capture only the rising edges.

It is not clear how this method will perform with other detectors. That is, if used with current state-of-the-art detectors using bright scintillators with fast SiPMs, will CNN-based time-of-flight estimation provide a complementary effect to the fast timing provided by the detector hardware leading to similar improvements in timing resolution to those shown here? Alongside this, CNN-based time-of-flight estimation may provide a robust solution for detectors making use of prompt photons, such as Cerenkov emissions (Kwon *et al* 2016, Brunner and Schaart 2017), since these timing methods are based on low number of detected photons in addition to the scintillation light, and CNNs may be well suited to best make use of this information. These topics will be investigated in future experimental and simulation studies.

Aside from investigating CNN-based time-of-flight estimation in other detectors and further attempts to optimize the convolutional neural network architecture, there are many areas of the use of CNNs for time-of-flight PET that are beyond the scope of this work, but which warrant further investigation, including the effect of sampling rate, type of photodetector, influence of noise properties in the waveform (electronic and statistical noise), and how to make use of the waveforms from multiple photodetectors in a light sharing block detector. In future work, we will investigate these topics with both simulations and further experiments.

It is beyond the scope of this proof-of-concept study to investigate or discuss the practical feasibility of implementing CNNs for waveform processing in a PET system, or developing methods of how to train a CNN for a large number of detectors (i.e. acquiring stepped point source for all detector pairs in the scanner). However, there likely exists efficient methods that may facilitate this, which will also be investigated in future work.

5. Conclusion

The feasibility and benefit of using convolutional neural networks to estimate time-of-flight directly from PET detector waveforms was demonstrated in this proof-of-concept study. Compared to conventional methods such as leading edge and constant fraction discrimination, CNN-based time-of-flight estimation improved coincidence timing resolution up to 23%.

Supplementary Material

Refer to Web version on PubMed Central for supplementary material.

Acknowledgments

The authors would like to thank members of the MIPET group at UC Davis for helpful discussions related to this work. This work was funded by NIH R35 CA197608.

References

Anghinolfi F, Jarron P, Martemiyanov A, Usenko E, Wenninger H, Williams M, Zichichi A. NINO: an ultra-fast and low-power front-end amplifier/discriminator ASIC designed for the multigap resistive plate chamber. Nucl Instrum Methods Phys Res A. 2004; 533:183–7.

Ashmanskas W, Legeyt B, Newcomer F, Panetta J, Ryan W, Van Berg R, Wiener R, Karp J. Waveform-sampling electronics for a whole-body time-of-flight PET scanner. IEEE Trans Nucl Sci. 2014; 61:1174–81. [PubMed: 25484379]

Auffray E, Frisch B, Geraci F, Ghezzi A, Gundacker S, Hillemanns H, Jarron P, Meyer T, Paganoni M, Pauwels K. A comprehensive & systematic study of coincidence time resolution and light yield using scintillators of different size and wrapping. IEEE Trans Nucl Sci. 2013; 60:3163–71.

Barrett HH, Hunter WCJ, Miller BW, Moore SK, Chen Y, Furenlid LR. Maximum-likelihood methods for processing signals from gamma-ray detectors. IEEE Trans Nucl Sci. 2009; 56:725–35. [PubMed: 20107527]

Berg E, Roncali E, Cherry SR. Optimizing light transport in scintillation crystals for time-of-flight PET: an experimental and optical Monte Carlo simulation study. Biomed Opt Express. 2015; 6:2220. [PubMed: 26114040]

Berg E, Roncali E, Hutchcroft W, Qi J, Cherry SR. Improving depth, energy and timing estimation in PET detectors with deconvolution and maximum likelihood pulse shape discrimination. IEEE Trans Med Imaging. 2016; 35:2436–46. [PubMed: 27295658]

Brunner S, Schaart D. BGO as a hybrid scintillator/Cherenkov radiator for cost-effective time-of-flight PET. Phys Med Biol. 2017; 62:4421–39. [PubMed: 28358722]

Conti M. Focus on time-of-flight PET: the benefits of improved time resolution. Eur J Nucl Med Mol Imaging. 2011; 38:1147–57. [PubMed: 21229244]

Karp JS, Surti S, Daube-Witherspoon ME, Muehllehner G. Benefit of time-of-flight in PET: experimental and clinical results. J Nucl Med. 2008; 49:462–70. [PubMed: 18287269]

Krizhevsky A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural networks. Adv Neural Inf Process Syst. 2012:1097–105.

Kwon SI, Gola A, Ferri A, Piemonte C, Cherry SR. Bismuth germanate coupled to near ultraviolet silicon photomultipliers for time-of-flight PET. Phys Med Biol. 2016; 61:L38. [PubMed: 27589153]

Lecoq P, Auffray E, Knapitsch A. How photonic crystals can improve the timing resolution of scintillators. IEEE Trans Nucl Sci. 2013; 60:1653–7.

Lecun Y, Bengio Y, Hinton G. Deep learning. Nature. 2015; 521:436–44. [PubMed: 26017442]

Moskal P, Zoń N, Bednarski T, Bialas P, Czerwiński E, Gajos A, Kamińska D, Kochanowski A, Korcyl G, Kowal J. A novel method for the line-of-response and time-of-flight reconstruction in TOF-PET detectors based on a library of synchronized model signals. Nucl Instrum Methods Phys Res A. 2015; 775:54–62.

Nemallapudi MV, Gundacker S, Lecoq P, Auffray E, Ferri A, Gola A, Piemonte C. Sub-100 ps coincidence time resolution for positron emission tomography with LSO: Ce codoped with Ca. Phys Med Biol. 2015; 60:4635. [PubMed: 26020610]

Ritt S. Design and performance of the 6 GHz waveform digitizing chip DRS4. Nuclear Science Symposium Conference Record, 2008 NSS'08 IEEE. 2008:1512–5.

Rolo M, Bugalho R, Goncalves F, Mazza G, Rivetti A, Silva J, Silva R, Varela J. TOFPET ASIC for PET applications. J Instrum. 2013; 8:C02050.

Ruiz-Gonzalez M, Bora V, Furenlid LR. Maximum-likelihood estimation of scintillation pulse timing. IEEE Trans Radiat Plasma Med Sci. 2017

Schaart DR, Seifert S, Vinke R, Van Dam HT, Dendooven P, Löhner H, Beekman FJ. LaBr3: Ce and SiPMs for time-of-flight PET: achieving 100 ps coincidence resolving time. Phys Med Biol. 2010; 55:N179. [PubMed: 20299734]

Seifert S, Van Der Lei G, Van Dam HT, Schaart DR. First characterization of a digital SiPM based time-of-flight PET detector with 1 mm spatial resolution. Phys Med Biol. 2013; 58:3061. [PubMed: 23587636]

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

Spanoudaki V, Levin CS. Photo-detectors for time of flight positron emission tomography (ToF-PET). Sensors. 2010; 10:10484–505. [PubMed: 22163482]

Spanoudaki VC, Levin C. Investigating the temporal resolution limits of scintillation detection from pixellated elements: comparison between experiment and simulation. Phys Med Biol. 2011; 56:735. [PubMed: 21239845]

Spurrier MA, Szupryczynski P, Yang K, Carey AA, Melcher CL. Effects of co-doping on the scintillation properties of LSO: Ce. IEEE Trans Nucl Sci. 2008; 55:1178–82.

Surti S, Karp J, Muehllehner G, Raby P. Investigation of lanthanum scintillators for 3D PET. Nuclear Science Symposium Conference Record, 2002 IEEE. 2002:1177–81.

Surti S, Karp JS. Advances in time-of-flight PET. Phys Medica. 2016; 32:12–22.

Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A. Going deeper with convolutions. Proceedings of the IEEE conference on computer vision and pattern recognition. 2015:1–9.

Van Dam HT, Borghi G, Seifert S, Schaart DR. Sub-200 ps CRT in monolithic scintillator PET detectors using digital SiPM arrays and maximum likelihood interaction time estimation. Phys Med Biol. 2013; 58:3243–57. [PubMed: 23611889]

Vandenberghe S, Mikhaylova E, D'hoe E, Mollet P, Karp J. Recent developments in time-of-flight PET. Eur J Nucl Med Mol Imaging. 2016; 3:1.

Vinke R, Löhner H, Schaart D, Van Dam H, Seifert S, Beekman F, Dendooven P. Time walk correction for TOF-PET detectors based on a monolithic scintillation crystal coupled to a photosensor array. Nucl Instrum Methods Phys Res A. 2010; 621:595–604.

Xie Q, Kao C-M, Wang X, Guo N, Zhu C, Frisch H, Moses WW, Chen C-T. Potentials of digitally sampling scintillation pulses in timing determination in PET. IEEE Trans Nucl Sci. 2009; 56:2607–13. [PubMed: 20376278]

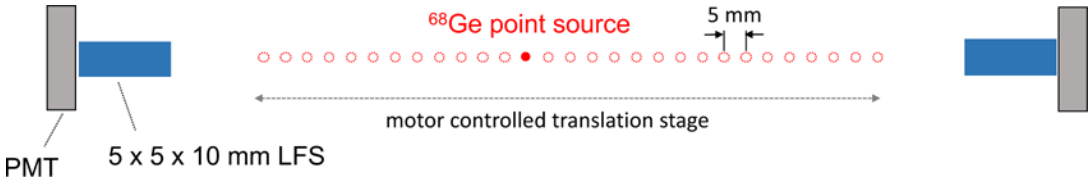

Figure 1.

Schematic of the detector setup used to acquire 511 keV coincidence waveforms. The point source was automatically stepped between the detectors (5 mm step size) using a motor-controlled translation stage. The red circles indicate the 29 source positions used for data acquisition.

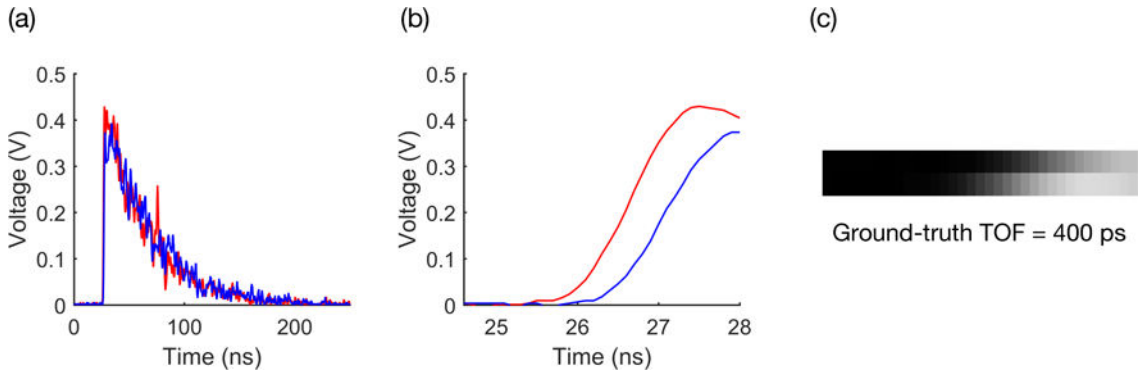

Figure 2.

(a) Sample of a digitized waveform pair for a 511 keV coincidence event. (b) Rising edges of the waveforms after applying the 3.5 ns crop. (c) Two-dimensional array of the coincident waveform pair used as the input to the CNN. The ground-truth time-of-flight (TOF) is stored for each waveform pair and used when training the CNNs.

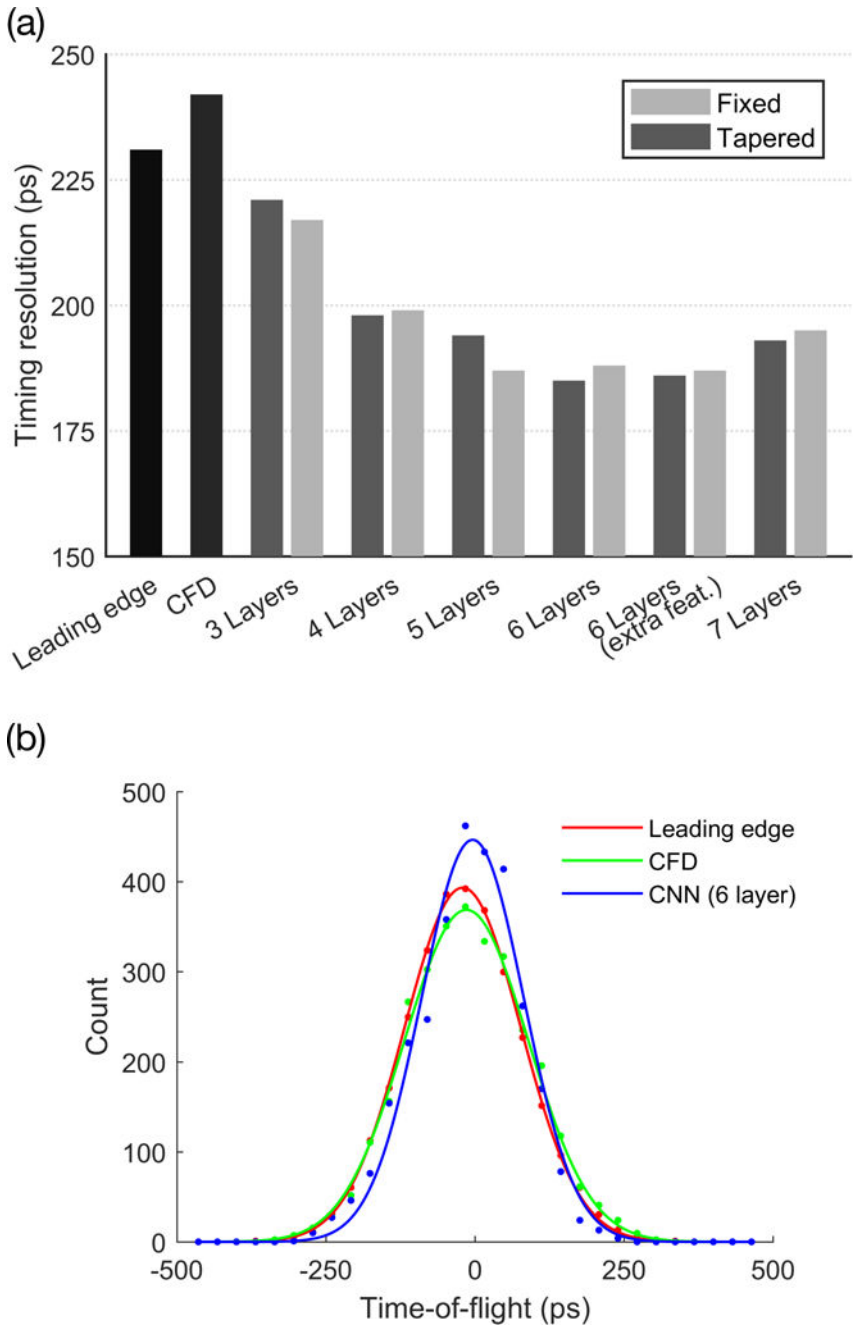

Figure 3.

(a) Comparison of coincidence timing resolution with all convolutional neural network configurations listed in Tables 1 and 2, leading edge discrimination, and constant fraction discrimination (CFD). Error bars are omitted since the standard deviation for all configurations was < 5 ps. (b) Sample timing spectra for the central source position for leading edge, CFD, and the tapered 6-layer CNN.

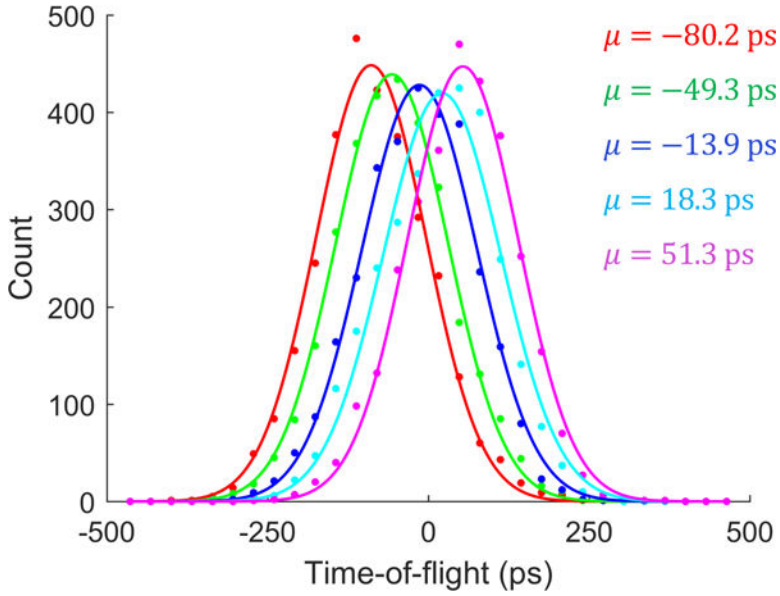

Figure 4.

Time-of-flight spectra for the five central point source positions, obtained with the 5-layer tapered CNN. The legend provides the mean time-of-flight for each of the five source positions, obtained from the Gaussian fits.

Phys Med Biol. Author manuscript; available in PMC 2019 January 11.

Table 1

Description of the layers for each convolutional neural network architecture for the fixed configuration (first table) and the tapered configuration (second table). The input layer (first layer) and rectifying linear units (ReLU) that follow the convolutional layers are not shown here. The two-dimensional values for each convolutional layer (conv) indicate the filter size, while the values for the max pooling layers (maxpool) indicate the pooling region. The numbers in parentheses for the convolutional layers indicate the number of feature maps, while the numbers in parentheses for the fully connected layers indicate the number of outputs.

Fixed CNN Configuration

3-Layer	4-Layer	5-Layer	6-Layer	6-Layer (2x feature maps)	7-Layer
conv 2×5 (64) *maxpool 1×2*	conv 2×5 (64)	conv 2×5 (64)	conv 2×5 (64)	conv 2×5 (128)	conv 2×5 (64)
	conv 1×5 (64) *maxpool 1×2*	conv 1×5 (64) *maxpool 1×2*	conv 1×5 (64) *maxpool 1×2*	conv 1×5 (128) *maxpool 1×2*	conv 1×5 (64) *maxpool 1×2*
		conv 1×5 (128) *maxpool 1×2*	conv 1×5 (128)	conv 1×5 (256)	conv 1×5 (128)
			conv 1×5 (128) *maxpool 1×2*	conv 1×5 (256) *maxpool 1×2*	conv 1×5 (128) *maxpool 1×2*
					conv 1×5 (196) *maxpool 1×1*
FullyConnected (256)					
Dropout (50%)					
FullyConnected (1)					
Regression					

Tapered CNN Configuration

3-Layer	4-Layer	5-Layer	6-Layer	6-Layer (2x feature maps)	7-Layer
conv 2×11 (64) *maxpool 1×2*	conv 2×11 (64)	conv 2×11 (64)	conv 2×11 (64)	conv 2×11 (128)	conv 2×11 (64)
	conv 1×7 (64) *maxpool 1×2*	conv 1×7 (64) *maxpool 1×2*	conv 1×7 (64) *maxpool 1×2*	conv 1×7 (128) *maxpool 1×2*	conv 1×7 (64) *maxpool 1×2*
		conv 1×5 (128) *maxpool 1×2*	conv 1×5 (128)	conv 1×5 (256)	conv 1×5 (128)
			conv 1×5 (128) *maxpool 1×2*	conv 1×5 (256) *maxpool 1×2*	conv 1×5 (128) *maxpool 1×2*
					conv 1×4 (196) *maxpool 1×1*

Tapered CNN Configuration

3-Layer	4-Layer	5-Layer	6-Layer	6-Layer (2x feature maps)	7-Layer
FullyConnected (256)					
Dropout (50%)					
FullyConnected (1)					
Regression					