



Research article

Transformation of PET raw data into images for event classification using convolutional neural networks

Paweł Konieczka^{1*}, Lech Raczyński¹, Wojciech Wiślicki¹, Oleksandr Fedoruk¹, Konrad Klimaszewski¹, Przemysław Kopka¹, Wojciech Krzemień², Roman Y. Shopa¹, Jakub Baran^{3,4}, Aurélien Coussat^{3,4}, Neha Chug^{3,4}, Catalina Curceanu⁵, Eryk Czerwiński^{3,4}, Meysam Dadgar^{3,4}, Kamil Dulski^{3,4}, Aleksander Gajos^{3,4}, Beatrix C. Hiesmayr⁶, Krzysztof Kacprzak^{3,4}, Łukasz Kapłon^{3,4}, Grzegorz Korcyl^{3,4}, Tomasz Kozik^{3,4}, Deepak Kumar^{3,4}, Szymon Niedźwiecki^{3,4}, Szymon Parzych^{3,4}, Elena Pérez del Río^{3,4}, Sushil Sharma^{3,4}, Shivani Shivani^{3,4}, Magdalena Skurzok^{3,4,5}, Ewa Łucja Stepien^{3,4}, Faranak Tayefi^{3,4} and Paweł Moskal^{3,4}

¹ Department of Complex Systems, National Centre for Nuclear Research, 05-400 Świerk, Poland

² High Energy Physics Division, National Centre for Nuclear Research, 05-400 Świerk, Poland

³ Marian Smoluchowski Institute of Physics, Jagiellonian University, 31-348 Cracow, Poland

⁴ Center for Theranostics, Jagiellonian University, 31-348 Cracow, Poland

⁵ INFN, National Laboratory of Frascati, 00044 Frascati, Italy

⁶ University of Vienna, Faculty of Physics, 1090 Vienna, Austria

* **Correspondence:** Email: pawelkonieczka@gmail.com.

Abstract: In positron emission tomography (PET) studies, convolutional neural networks (CNNs) may be applied directly to the reconstructed distribution of radioactive tracers injected into the patient's body, as a pattern recognition tool. Nonetheless, unprocessed PET coincidence data exist in tabular format. This paper develops the transformation of tabular data into n -dimensional matrices, as a preparation stage for classification based on CNNs. This method explicitly introduces a nonlinear transformation at the feature engineering stage and then uses principal component analysis to create the images. We apply the proposed methodology to the classification of simulated PET coincidence events originating from NEMA IEC and anthropomorphic XCAT phantom. Comparative studies of neural network architectures, including multilayer perceptron and convolutional networks, were conducted. The developed method increased the initial number of features from 6 to 209 and gave the best precision results (79.8%) for all tested neural network architectures; it also showed the smallest decrease when changing the test data to another phantom.

Keywords: positron emission tomography; convolutional neural network; kernel principal component analysis; medical imaging

1. Introduction

In the majority of machine learning methods, e.g., the multilayer perceptron (MLP), input data are unstructured and altogether represented in the form of 1-dimensional (1-D) feature vectors. The performance of classification techniques is therefore highly dependent on pre-processing steps, i.e., the feature extraction or feature selection. On the other hand, in deep learning, particularly the convolutional neural networks (CNNs), the features are learned automatically and represented hierarchically at subsequent levels [1, 2]. Since convolutional units analyze only a small subset of the output data from the preceding layer, the network may be much deeper with fewer parameters [3]. Recent advances in CNNs have presented an opportunity for solving classification problems in many disciplines [4–8]. Most of these methods are either applied to 1-D vectors or 2-D matrices as inputs to the CNN. Greater dimensionality of input images (e.g., 3-D) requires much larger GPU memory consumption. For this reason, a few methods have been developed to reduce the dimensionality of the data to decrease computational costs [9, 10].

In the case of the positron emission tomography (PET) [11–13], CNNs may be applied directly to the distribution of a radioactive tracer injected into the patient's body, as a pattern recognition tool. However, much PET data are provided as 1-D vectors, and this brings questions of whether CNNs can be effectively trained on them. Examples of such tasks are an estimation of time of flight from signals registered in scintillators [14] and classification of coincidence events acquired by PET scanners [15].

The first approach to the transformation of non-image data into an image form for CNN architectures was presented in Ref. [16]. The method, called DeepInsight, constructs an image by placing the similar elements of 1-D input vectors together and the dissimilar ones further apart, thus enabling collective use of neighboring elements. Arrangement of the features on the 2-D image space is crucial to exploring their relative importance and correlation. But, the toughest problem of visualization of the input space in the image form is determining how to represent low-dimensional input data as matrices with a large number of pixels that can be efficiently treated by a CNN. This problem is tackled in this paper. We propose a method to increase the size of a 1-D vector. Namely, if one has only a small number of features for a data point, then we introduce a method to involve higher-order correlations of features. Those are arranged with the DeepInsight methodology [16] to a 2-D image for further processing by CNN architectures. We investigate the quality of the proposed approach by considering the problem of coincidence event classification in the Jagiellonian PET (J-PET) detector [17–23].

PET studies rely on the determination of the spatial distribution of the concentration of a selected substance in the body, and in some cases, also its kinetics, i.e., dependence of these distributions on time [24, 25]. For this purpose, the patient is administered a radiopharmaceutical, i.e., a pharmaceutical containing a radioactive isotope. Tomography uses the fact that the positrons emitted from the β^+ radioactive marker and electrons from the patient's atoms are annihilated, resulting in two γ photons being emitted back-to-back. The PET detection system is usually arranged in layers, forming a ring around the diagnosed patient [26]. In the basic measurement scheme, information about the single event of the electron-positron annihilation is collected in the form of a line joining the detected locations that passes directly through the point of annihilation, called the line of response (LOR). The set of registered LORs forms the basis for PET image reconstruction.

During the measurement, an event is regarded as valid if two γ photons are registered within a pre-selected coincidence time window and the energy deposited in scintillators by both γ photons exceeds the predefined level. Nonetheless, a number of acquired coincidences having met the above criteria are still spurious events and are unwanted. Different types of events in a routine PET examination are usually classified as follows:

- *True*: Two γ photons originate from a single positron-electron annihilation and reach the scintillators without prior scattering.
- *Phantom-scattered*: Two γ photons originate from a single positron-electron annihilation when one or both of them have undergone Compton interactions in the patient.
- *Detector-scattered*: Two γ photons originate from a single positron-electron annihilation with one or both of them undergoing a Compton interaction(s) inside the detector, before being registered.
- *Random*: Two γ photons originate from two different positron-electron annihilations occurring within the coincidence time window.

Only the true events are useful for PET imaging. The random and scattered ones distort the reconstructed distribution of the radiotracer and constitute a background. A variety of methods are known for estimation of the contribution of random, scattered and true events during a PET examination [27–29]. This paper is not dedicated to background corrections, but to the event classification scheme for the preselection of the true events using a deep learning approach.

2. Data and methods

2.1. PET data description

We assume that each event is described by the following six features and thus represents a point in 6-D space:

1. The angular difference in the transaxial section between the detection points.
2. The absolute value of the registration time difference of two γ photons.
3. The distance between the reconstructed positions in the scintillators of both γ photons.
4. The sum of energies deposited by both γ photons in the scintillators.
5. The absolute value of the difference of energies deposited by both γ photons in the scintillators.
6. The attenuation coefficient along each LOR extracted based on the attenuation map of the phantom, $e^{-x/l}$, where x stands for the actual distance traveled by a photon, and l for its mean-free path in matter.

2.2. Transformation of non-image data into an image

We follow the idea presented in Ref. [16] and apply the training data to find the location of features in the 2-D space. Consider data stored in the matrix $\mathbf{X} \in \mathbb{R}^{L \times N}$, where L is the number of collected events and N stands for the number of features. The data matrix may be expressed as

$$\mathbf{X} = [\mathbf{x}_1 | \mathbf{x}_2 | \dots | \mathbf{x}_N], \quad (2.1)$$

where each feature vector \mathbf{x}_j consists of L training samples. In order to distinguish the column vectors denoted by $\mathbf{x}_j \in \mathbb{R}^L$ from the row vectors of matrix \mathbf{X} describing the i^{th} event, the latter ones will be

denoted by an upper index $\mathbf{x}^i \in \mathbb{R}^N$. In the DeepInsight approach [16], the dimensionality reduction methods, e.g., t-SNE [30] or kernel principal component analysis (PCA) [31], are applied to the data matrix \mathbf{X} in order to obtain the feature positions in the 2-D plane.

In kernel PCA, we consider a function Φ that maps the original vector into a new space Ω , finite or infinite, i.e., each feature $\mathbf{x}_j \in \mathbb{R}^L$ is projected to a point $\Phi(\mathbf{x}_j)$, and these points build up the matrix $\Phi(\mathbf{X})$ of larger dimensionality than \mathbf{X} . We assume that the mean value of the data set $\Phi(\mathbf{X})$ is 0, i.e., $\sum_{j=1}^N \Phi(\mathbf{x}_j) = \mathbf{0}$. It may be shown that the standard PCA problem of finding eigenvalues (Λ) and a matrix built from eigenvectors (\mathbf{A}) of the covariance matrix

$$\mathbf{C} = \frac{1}{N} \cdot \Phi(\mathbf{X}) \cdot \Phi(\mathbf{X})^T, \quad (2.2)$$

in the form

$$\Lambda = \mathbf{A} \cdot \mathbf{C} \cdot \mathbf{A}^T, \quad (2.3)$$

may be replaced by the problem of diagonalization of the $N \times N$ kernel matrix \mathbf{B}

$$\Lambda = \frac{1}{N} \cdot \mathbf{V} \cdot \mathbf{B} \cdot \mathbf{V}^T, \quad (2.4)$$

where the kernel reads as

$$\mathbf{B} = \Phi(\mathbf{X})^T \cdot \Phi(\mathbf{X}), \quad (2.5)$$

and $\mathbf{A} = \mathbf{V} \cdot \Phi(\mathbf{X})^T$ (cf. ref. [31]). As it is seen from Eq. (2.4), \mathbf{V} stores the eigenvectors of the matrix \mathbf{B}/N , which is the kernel matrix \mathbf{B} normalized by the number of features (N). The application of a kernel matrix has a number of advantages. First of all, the compact representation $\Phi(\mathbf{Y})$ of each vector from the data matrix \mathbf{X} may be evaluated based on the following:

$$\begin{aligned} \Phi(\mathbf{Y}) &= \mathbf{A} \cdot \Phi(\mathbf{X}) \\ &= \mathbf{V} \cdot \Phi(\mathbf{X})^T \cdot \Phi(\mathbf{X}) \\ &= \mathbf{V} \cdot \mathbf{B}, \end{aligned} \quad (2.6)$$

where the function Φ is never explicitly used [32–34]. The technique to replace the dot product by the kernel matrix \mathbf{B} in Eq. (2.5) is called the "kernel trick".

In DeepInsight, two eigenvectors \mathbf{v}_1 and \mathbf{v}_2 , respectively corresponding to the largest eigenvalues λ_1 and λ_2 , are used to present the features as points on a 2-D plane. In this approach, it is proposed to use the nonlinear function $\Phi(\mathbf{x}_j)$ representing the transformation of each feature j from the sample space, i.e., each element of the kernel matrix in Eq. (2.5) is constructed as follows:

$$\mathbf{B}_{i,j} = \Phi(\mathbf{x}_i)^T \cdot \Phi(\mathbf{x}_j). \quad (2.7)$$

Nonlinear mappings of feature space provide much more efficient tools for event classification as compared to linear ones. These are ensured by the flexibility of the classification criteria, enlarged dimensionality of the feature space and better exploitation of the correlations between features. These aspects are especially important when the feature space dimensionality (N) is very small. For this particular case, we propose the DeepInsight "modified" process.

2.3. DeepInsight "modified"

First of all, we impose that the nonlinear function Φ has a finite support and

$$\Phi : \mathbb{R}^N \rightarrow \mathbb{R}^M. \quad (2.8)$$

The new representation

$$\mathbf{Z} = [\mathbf{z}_1 | \mathbf{z}_2 | \dots | \mathbf{z}_M] \quad (2.9)$$

is obtained, where \mathbf{Z} is the $L \times M$ matrix and

$$\mathbf{z}^i = \Phi(\mathbf{x}^i) \in \mathbb{R}^M \quad (2.10)$$

for $i = 1, 2, \dots, L$. After subtraction of the mean value from data set \mathbf{Z} , i.e., after requiring $\sum_{j=1}^M \mathbf{z}_j = 0$, the standard procedure of diagonalization of the kernel matrix \mathbf{B} is performed (see Eq. (2.4) for details). The only difference at this stage is that the simple dot product is used (nonlinearity was applied in Eq. (2.10)) and the $M \times M$ kernel matrix is defined as

$$\mathbf{B} = \mathbf{Z}^T \cdot \mathbf{Z}. \quad (2.11)$$

Finally, n eigenvectors $\mathbf{v}_1, \dots, \mathbf{v}_n$ corresponding to the n largest eigenvalues are used to present the positions of M features in the n -D image space. The values of each element of the i^{th} event, \mathbf{z}^i , are calculated explicitly according to the definition of the function Φ . In this work, we will consider only the polynomial functions. For the polynomial of degree d , the number of dimensions M in a new space is given by

$$M = \frac{(N + d)!}{N! d!} - 1, \quad (2.12)$$

and, for a fixed number of features, N is a strongly increasing function of d . For instance, for $N = 2$, one has

$$\mathbf{x}^i = [x_1^i, x_2^i], \quad (2.13)$$

and, for $d = 3$, the new space is 9-dimensional and

$$\begin{aligned} \mathbf{z}^i &= \Phi(\mathbf{x}^i) \\ &= [x_1^i, x_2^i, x_1^i \cdot x_2^i, (x_1^i)^2, (x_2^i)^2, (x_1^i)^2 \cdot x_2^i, x_1^i \cdot (x_2^i)^2, (x_1^i)^3, (x_2^i)^3]. \end{aligned} \quad (2.14)$$

Points in the Cartesian space spanned by the eigenvectors \mathbf{v}_1 to \mathbf{v}_n define only the location of features and not their values. The feature locations in the Cartesian coordinates are determined by their similarity. Feature values will be visible as the image intensity in a given location. Same as in the DeepInsight method, we define the final image as the rectangular convex hull of all features, framed in a horizontal or vertical direction. The transformation process for six feature vectors to an input image for CNN is shown in Figure 1.

For a detailed description of the DeepInsight pipeline, we refer the reader to Ref. [16]. Hereafter, we will refer to the original DeepInsight method as DeepInsight 'raw', and to the proposed approach as DeepInsight 'modified'.

Our results present an extension of the method that enables its application to new fields. Unlike DeepInsight, which employs an implicit nonlinear transformation using a kernel trick, our approach

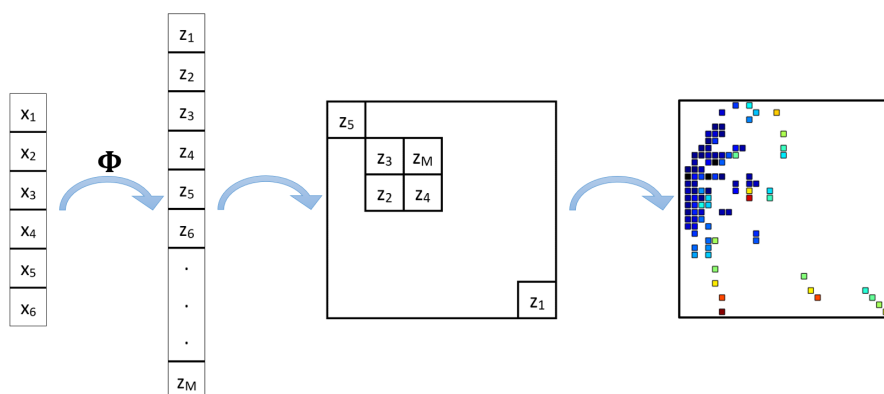


Figure 1. An illustration of transformation from feature vectors to a CNN input image.

explicitly expands the dimensions of the input data, allowing us to control an effective number of variables. This puts under supervision also the number of non-zero pixels and an image size, thus enabling optimization of the computational effort. All of this broadens the scope of the method from data of high dimensionality to data from detectors providing data of high granularity but low dimensionality, such as the J-PET.

2.4. CNN architectures

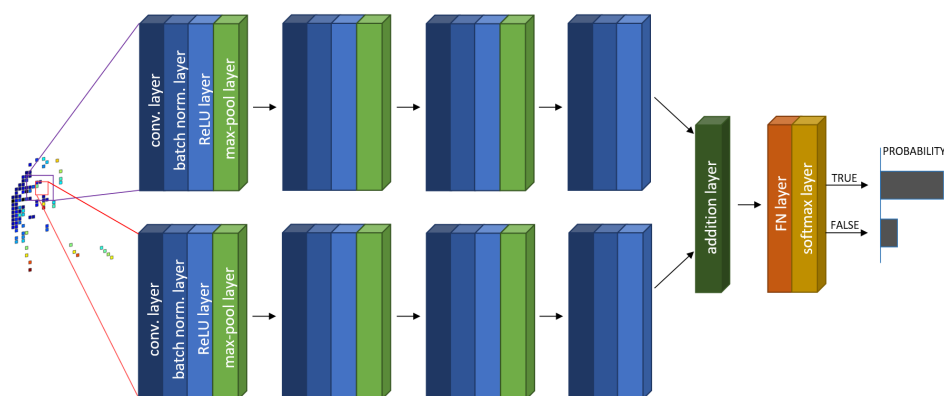


Figure 2. CNN architecture used in DeepInsight for the classification problem.

Once a 1-D vector that stores the information about features, which, in our case, describes the coincidence event, is transformed into a n -D matrix, it can be further processed to the CNN. The research has focused on the DeepInsight method and optimization of a single-path convolutional network.

DeepInsight CNN architecture consists of two parallel CNN architectures, where each consists of four convolutional layers and three layers designed to reduce dimensionality, called *max pooling* layers (see Figure 2). Each parallel convolutional "pathway" has a different filter size to focus on different areas of an image. Each convolutional layer is followed by a batch normalization layer and a rectified linear unit (ReLU) layer. Batch normalization layers are used to speed up the training of the CNN and prevent overfitting through normalization by recentering and rescaling each data batch. The max pooling layers are used to reduce the dimensionality. This enables to reduction of the number

of parameters, thus preventing overfitting and decreasing the training time. The ReLU is the most commonly used nonlinear activation function in CNNs. The major benefits are a reduction of the likelihood of a vanishing gradient and computational efficiency. The outputs of the fourth convolutional layers are later fused in the last fully connected layer. The last layer is called the *softmax* classifier and calculates the probabilities of class labels.

CNN architecture hyperparameters, such as the size of filters, initial number of filters, momentum value, L2 regularization value and initial learning rate, were optimized by using Bayesian optimization, with the aim being to find the model hyperparameters that yield the best score on the validation data. The biggest advantage, compared to the random or grid search, is that the past evaluations have an impact on the future ones. The algorithm spends time on selecting the next hyperparameters in order to make fewer evaluations. Bayesian optimization is one of the most effective techniques in terms of the number of function evaluations required [35].

2.5. Simulation scenario

The performance of the proposed classification scheme was investigated based on Monte Carlo simulation studies, where exact information about all four types of events, i.e., true, phantom-scattered, detector-scattered and random, is available. The Geant4 Application for Tomographic Emission (GATE) [36, 37] is open-source software for numerical simulations in the areas of medical imaging and radiotherapy. The J-PET scanner geometry, as shown in Figure 3, was implemented in GATE. The 2-layer detector consists of seven rings, and each ring is composed of 24 cylindrically arranged modules. Each module was built from 32 plastic scintillator strips (16 strips per layer) with a width of 30 mm and length of 330 mm. The gap length between adjoining rings is 20 mm. The detector is a cylinder with a radius of 415 mm. The simulation setup was consistent with the one used in our previous works [38].

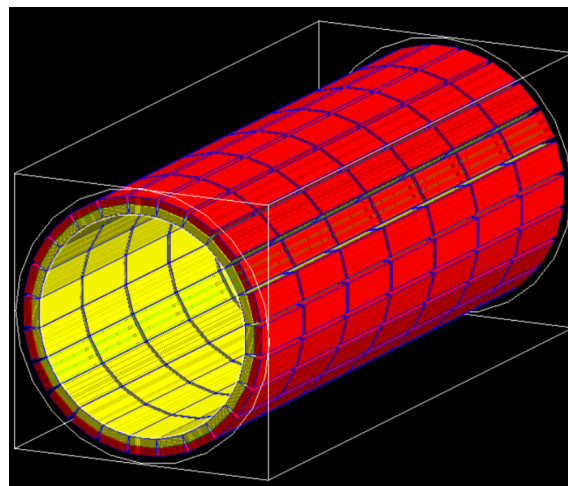


Figure 3. Schematic view of simulated J-PET scanner.

The investigated sources of radiation were the NEMA IEC phantom [39] and XCAT phantom [40]. The NEMA IEC phantom and XCAT phantom activity maps are depicted in Figure 4 and Figure 5, respectively.

A coincidence event was defined as a set of consecutive interactions of photons detected within

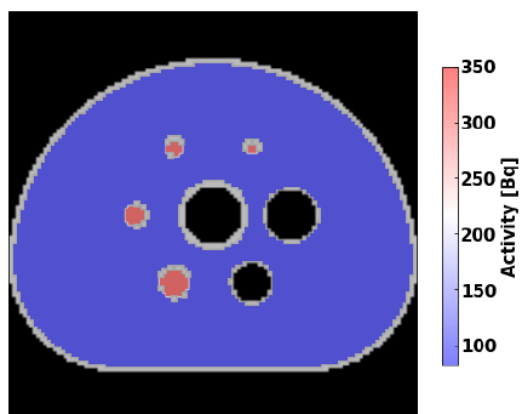


Figure 4. Activity distribution of the NEMA IEC phantom superimposed on the computed tomography (CT) image. Figure taken from [38].

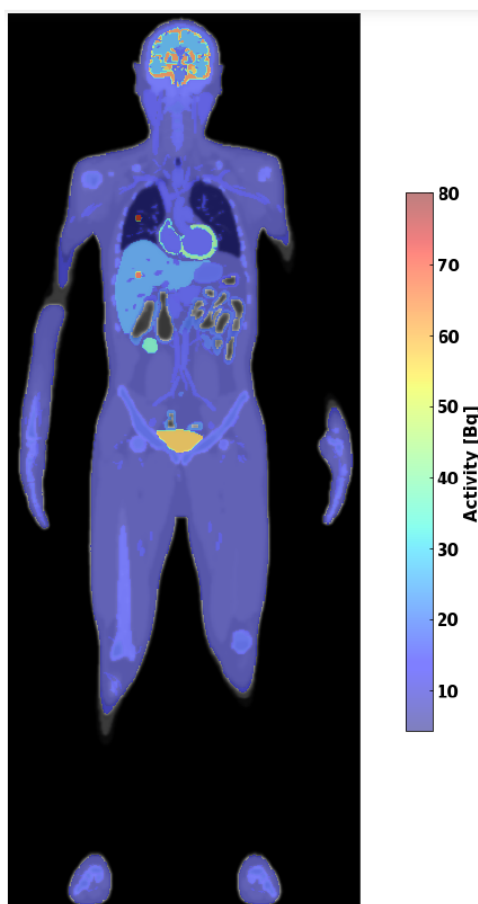


Figure 5. Activity distribution of the XCAT phantom superimposed on the CT image. Figure taken from [38].

the fixed time window of 3 ns. The data set was reduced in order to reject events from outside of the detector field of view. Two selection criteria were employed for each of the phantoms. The first

criterion ensured that the reconstructed position of the annihilation point within the (x, y) cross-section was confined to a circular region with a radius of 30 cm for the NEMA IEC phantom, and 40 cm for the XCAT phantom. The second criterion restricted the reconstructed position along the axial direction to be within 20 cm from the center of the detector for the NEMA IEC phantom, and 100 cm from the center of the detector for the XCAT phantom. Moreover, only events with exactly two interactions registered with an energy loss larger than 200 keV each were accepted [41]. For the NEMA IEC phantom simulation, a total of 6.5 million coincidences that fulfill above conditions were recorded, corresponding approximately to a five-minute scan for a real J-PET data acquisition. The total number of events included 3.9 million trues, 2.0 million phantom-scattered, 0.1 million detector-scattered and 0.5 million randoms.

For the XCAT phantom simulation, a total of 8.6 million coincident events fulfilling the conditions were recorded—likewise corresponding to an approximately five-minute scan. The total number of events included 4.3 million trues, 2.2 million phantom-scattered, 0.1 million detector-scattered and 2.0 million randoms.

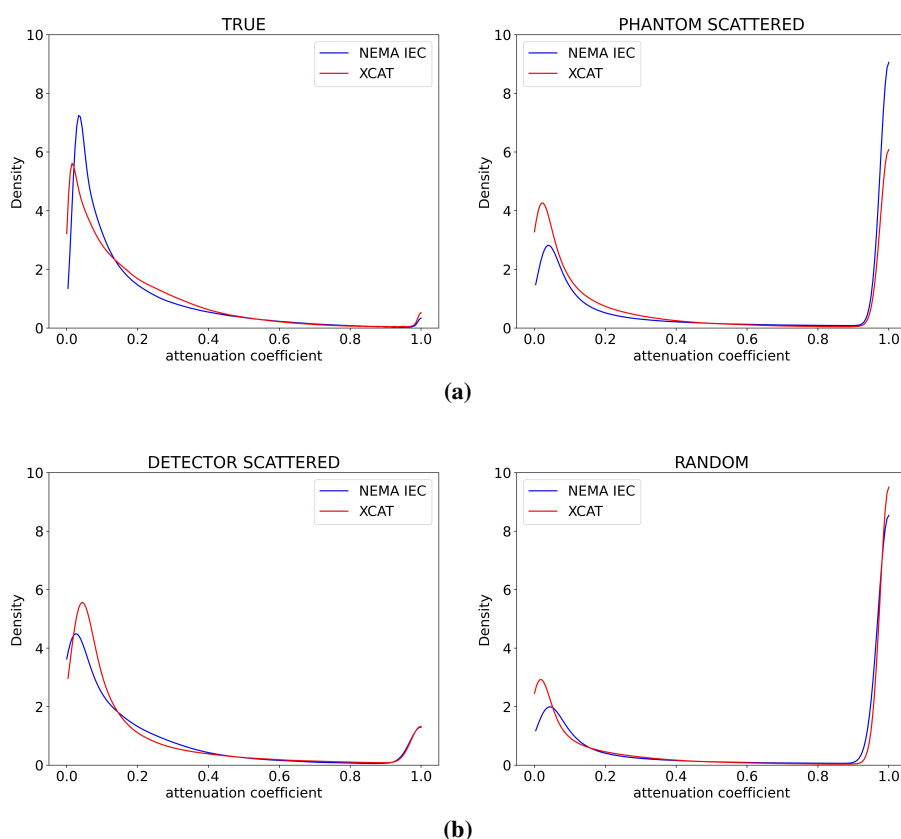


Figure 6. Distribution of attenuation coefficients for each class of events.

Before event classification using the CNN, this data set was reduced in order to reject events from outside of the phantom. Figure 6 shows the distribution of the attenuation coefficients for both phantoms. The shift of the distribution toward small values for the XCAT phantom was caused by the difference in the geometry of the phantoms. The influence of the phantom geometry is best seen in the example of the distribution of phantom-scattered events; particularly the greater share of events with an attenuation factor close to 1 for the NEMA IEC phantom is due to the fact that it is several times

smaller than XCAT, which means that there are more LORs that do not pass through the phantom. A threshold on the attenuation factor was applied: events whereby the factor was greater than 0.999 were rejected (Table. 1). Consequently, for the NEMA IEC phantom, the initial total number of 6.5 million coincidence events was reduced to 5.1 million. For the XCAT phantom, the initial total number of 8.6 million coincident events was reduced to 6.6 million. Note that the number of true coincidences remained intact. More details about the pre-processing of the J-PET data may be found in Refs. [42, 43].

Table 1. Event-type distribution (data in millions)

Phantom		True	Phantom Scattered	Detector Scattered	Random
NEMA IEC	Pre-cut	3.9	2.0	0.1	0.5
	After cut	3.9	0.9	0.1	0.2
XCAT	Pre-cut	4.3	2.2	0.1	2.0
	After cut	4.3	1.4	0.1	0.8

2.6. Quality measures

As mentioned in the previous section, the acquired data set consists of four types of coincidence events: true, phantom-scattered, detector-scattered and random. We consider both scattered and random events as negative instances, while the true events are positive ones. In order to measure the quality of classification, we calculated two parameters: the true positive rate (TPR) and positive predictive value (PPV). The TPR and PPV measure sensitivity and precision of the classification, respectively, and are defined as

$$\text{TPR} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}}, \quad (2.15)$$

$$\text{PPV} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}}. \quad (2.16)$$

The goal of the event selection is to maximize the classification precision (PPV) for assumed sensitivity (TPR) equal to 0.95, where each true coincidence rejected is associated with long-term exposure of the patient; therefore, loss of true signals should be limited and a level of 5% is still acceptable.

Since, in the proposed vector transformation scheme, the initial number of features increases from N to M , where $M - N$ features are naturally correlated with the original N features (cf. Eqs. (2.13) and (2.14) for details) an additional parameter that measures the feature overlapping (FO) is required. The FO indicates the fraction of the features that contribute to the same pixel in the output image. If no overlapping is observed, each feature corresponds to an individual pixel and FO is equal to 0. Moreover, we calculate the explained information (EI) parameter in order to estimate the percentage of variance explained by the first n most significant eigenvectors. The parameter EI is defined as

$$\text{EI} = \frac{\lambda_1 + \dots + \lambda_n}{\text{Tr}(\mathbf{\Lambda})}, \quad (2.17)$$

where $\text{Tr}(\mathbf{\Lambda})$ is the trace of the diagonal eigenvalue matrix $\mathbf{\Lambda}$. From Eq. (2.17), it is seen that $2/M \leq \text{EI} \leq 1$. The closer the value of EI to 1, the higher the compressibility of the feature space on the plane of the n most significant eigenvectors.

2.7. CNN training

The selection of training subset size was made by taking into account two criteria. First, the subset had to be a representative sample; for this purpose, the distributions of each feature were examined depending on the size of the selected training subset size. Second, the amount of training data has an impact on the training time; in this work, we set the upper time limit to 72 hours. The selected size of 30 thousand met both criteria, and the training time lasts 55 hours. The order of coincidences was randomized before training. The mini-batch size used in the stochastic gradient descent with momentum was 128 coincidences. Training was performed using the MATLAB 2019b Deep Learning Toolbox and 2x Tesla K80 GPU (24GB VRAM). Each hyperparameter optimization process took 30 epochs, and each CNN training took 300 epochs. Data were split in proportions of 9:1 (training to validation). We obtained a set of hyperparameters that gave the best performance on the validation set. A set of 100 thousand randomly chosen coincidences, without overlapping with the training and validation set, were used to evaluate the trained CNNs. The test set of coincidences was passed through trained CNNs to predict each class.

3. Results and discussion

3.1. Transformation of non-image data using kernel PCA

Experiments on the transformation of J-PET data using kernel PCA were performed using the NEMA IEC phantom.

In the original DeepInsight work [16], the number of image dimensions is fixed and equal to 2. We have conducted research on the effect of the number of dimensions on the amount of learnable parameters and efficiency of classification. Results in Table 2 show the values of the EI parameter according to the dimensionality of the data. Intuitively, the larger the dimensionality, the larger the EI parameter. Additionally, the number of learnable parameters increases by 1.5 orders of magnitude for an increase of dimension by 1.

Table 2. Effects of dimensionality of input images

Dimensionality	EI	Learnable parameters	PPV at 0.95 TPR
1	60.2%	0.5 M	78.2%
2	78.6%	11 M	79.8%
3	82.1%	312 M	-

Experiments have shown that the classification efficiency (PPV at TPR = 0.95) of 2-D images is greater than that of 1-D images, i.e., 79.8% and 78.2%, respectively (3-D images have not been tested due to excessive GPU utilization—the number of learnable parameters for 3-D images was too large). Therefore, further studies were carried out on 2-D images.

The default input image size for the DeepInsight method is 120×120. We conducted research on the effect of image size on the classification effectiveness. The DeepInsight 'raw' results in Figure 7 show that the PPV starts to decrease for an image size smaller than 30×30. However, the smaller the image size, the faster the learning process; therefore, the size of 30×30 was selected as optimal for the following experiments. We examined the effect of image size for DeepInsight 'modified'; the size of 30×30 was also optimal.

As mentioned in Section 2.1, each event is described by six features and is treated as the 1-D vector. The first investigation is concerned with the analysis of the quality of vector transformation into images. For this purpose, we chose to apply the parameters FO and EI that were introduced in Section 2.6.

In the proposed approach, introduced in Section 2.2, the initial number of six features is increased by using the nonlinear mapping Φ with finite support.

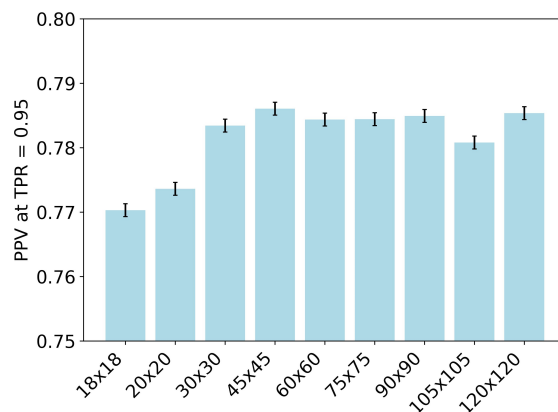


Figure 7. Comparison of PPVs at TPR = 0.95 for different sizes of input images for DeepInsight 'raw'.

In this work, we applied the polynomial function and carried out experiments for different polynomial degrees (d) from 1 to 5. In each case, the input data set \mathbf{X} was mapped into the new representation \mathbf{Z} as shown in Eq. (2.9). Next, the kernel matrix \mathbf{B} was evaluated according to Eq. (2.11). Finally, the eigenvalues ($\mathbf{\Lambda}$) and the eigenvectors (\mathbf{V}) of the kernel matrix were calculated and stored. The two most significant eigenvectors \mathbf{v}_1 and \mathbf{v}_2 , i.e., eigenvectors corresponding to the largest eigenvalues, were used to localize new features on the 2-D plane.

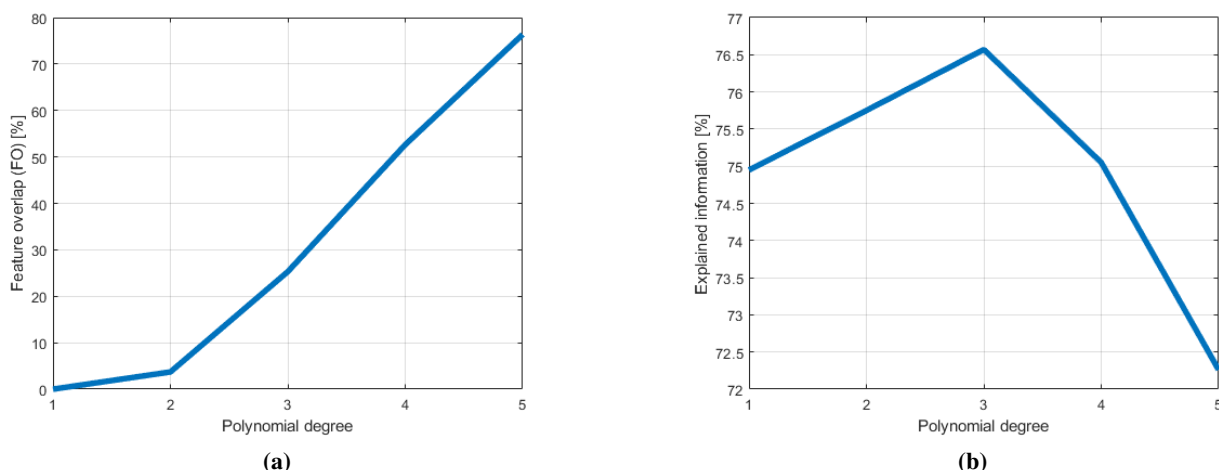


Figure 8. The quality of the transformation of 6-D feature vectors into images using the proposed methodology by optimizing the (a) FO and (b) EI. The calculations were performed for the polynomial function Φ (see Eq. (2.10)) with degrees from 1 to 5.

Table 3. FO and EI vs the degree of the polynomial

Degree number	No. features	No. non-zero pixels	FO	EI
1	6	6	0%	75%
2	27	26	4%	76%
3	83	62	25%	77%
4	209	99	53%	75%
5	461	109	75%	72%

The curves describing the FO and EI parameters as functions of the polynomial degree are shown in the left and right panels in Figure 8, respectively. From Figure 8, one can see that the features start to overlap, i.e., FO exceeds a low percentage, for polynomials of the second degree. In addition, for the third-degree polynomial, the EI parameter reaches maximum at the level of 77%. We selected the function Φ with polynomials of degree four for further analysis; the EI is only slightly smaller than the maximal value (EI = 75%), and the FO is 53%; but, compared to a third-degree polynomial, the number of non-zero pixels has increased by more than 50 %, and FO has increased twice (cf. Table 3).

In this context, it is worth characterizing the original transformation of 1-D vectors into images introduced by the DeepInsight authors [16]. Since, in this case, the final number of features is the same as the initial one and equal to N , the images contained only six non-zero pixels that stored the information about coincidence events (see top panels in Figure 9). The non-zero pixels were sparsely distributed inside the image and no overlapping is observed (FO = 0). We applied the Gaussian kernel (Eq. (2.5)) during the evaluations with the original DeepInsight methodology and we optimized the parameters of the kernel, i.e., the standard deviation of the Gaussian function, as described in the documentation [16]. For the optimal width of the Gaussian function, EI is equal to 72% and is slightly smaller than in our proposed approach. In Figure 9, the two exemplary images of two coincidence event types are shown. The results of processing using original DeepInsight methodology with the standard kernel PCA transform with an optimal size of Gaussian kernel are demonstrated in the top panels of Figure 9. Images obtained using the proposed approach with the fourth-degree polynomial mapping of six features are presented in the bottom panels of Figure 9. According to Eq. (2.12), the final number of the nonlinear combination of features in the proposed transformation scheme is $M = 209$. However, in the case of 110 features, the contribution to the same pixel is observed and, effectively, 99 pixels stored the information about each coincidence event. It can be seen that these 99 features are uniformly distributed in the image space. In Figure 9, the same true (left panels) and random (right panels) coincidence events are shown, and the values of the features, i.e., colors in the image, allow one to distinguish between the types of coincidences for both processing schemes.

3.2. Events classification comparative studies

Figure 10 shows the PPVs obtained for each of the neural network architectures considered in this work. The error bars indicate standard deviations and were estimated as the square root of measured numbers for Poisson data. Several configurations of the single-path CNNs were compared. Each of the CNNs consists of an input layer, several convolutional layers, a fully connected layer, a softmax layer and a classification layer. The analyzed architectures consist of 3–9 convolutional layers. CNN hyperparameters for single-path architectures, such as the initial number of filters, the size of filters,

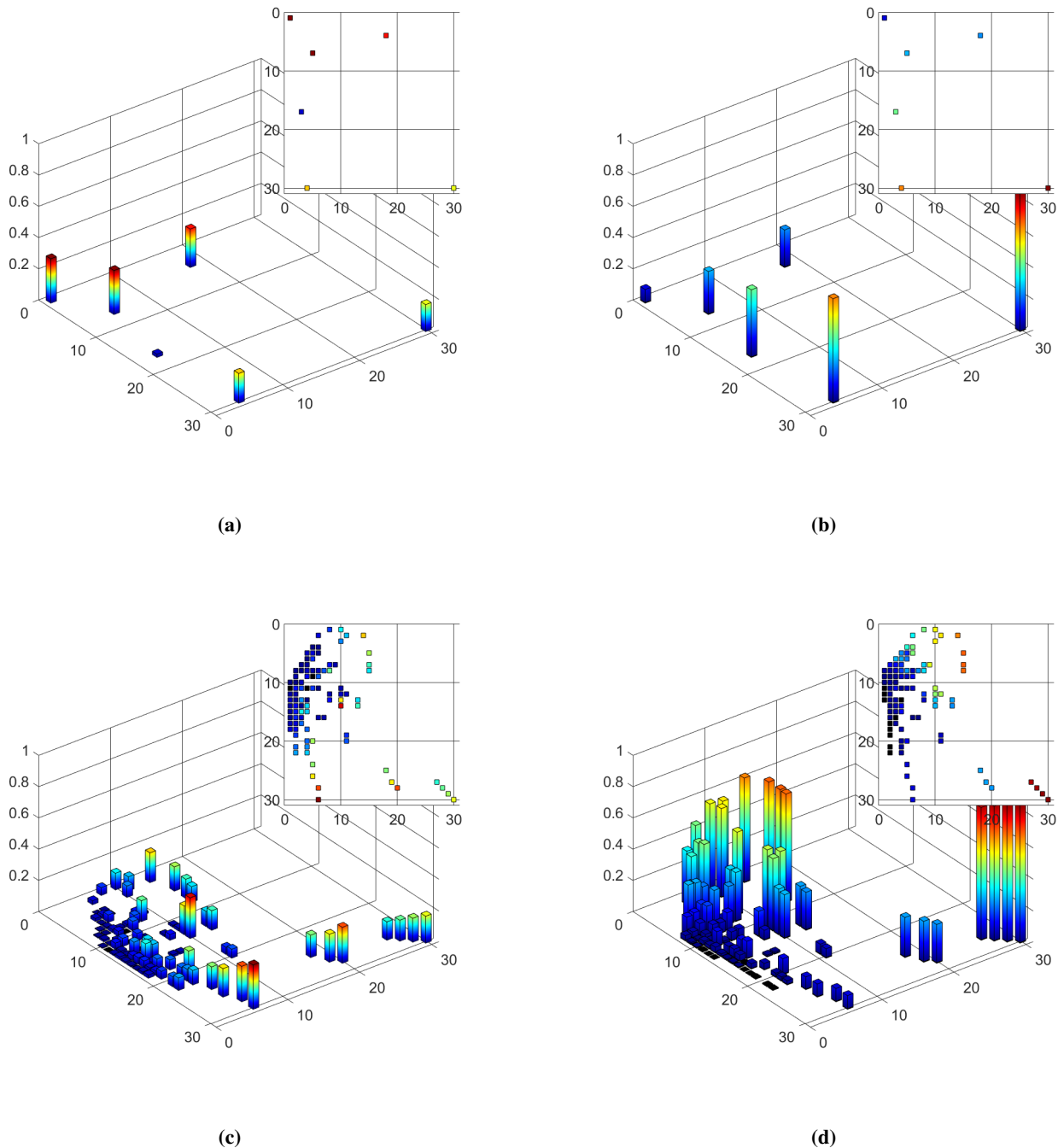


Figure 9. Two exemplary images (30×30) of different coincidence event types. Data were processed using DeepInsight 'raw' (top) and DeepInsight 'modified' with 4th-degree polynomial mapping of six features (bottom). The same true (left) and random (right) coincidence events are shown for both processing schemes.

initial learning rate, momentum value and L2 regularization, were optimized, similar to the DeepInsight methods, by Bayesian optimization. An increase in the efficiency of the classification can be observed along, with the rise of the number of convolutional layers in the range of three to six. The best results

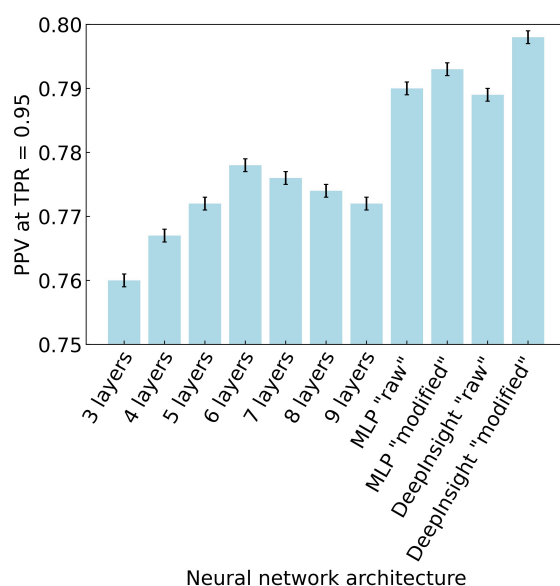


Figure 10. Comparison of PPVs at TPR = 0.95 for all neural network configurations.

for a single-path network were obtained for the architecture with six layers. A slight decrease in PPV for more than six layers suggests that the model is too complex and exhibits overfitting.

The best PPVs were given by the DeepInsight 'modified' method. In general, for CNNs, single-path architectures provide smaller PPVs than the double-path ones (DeepInsight).

DeepInsight 'modified' obtained better PPV results than DeepInsight 'raw'. The experiments were carried out on 30×30 images; for DeepInsight 'raw', reducing the size of the images also improved the classification performance. As a consequence of increasing the number of variables (DeepInsight 'modified'), and thus increasing the number of non-zero pixels in the images, better PPV results were obtained across virtually the entire TPR range (Figure 11).

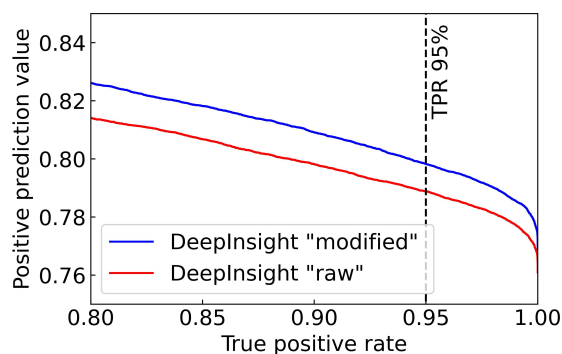


Figure 11. Comparison of PPV as a function of TPR for 'raw' DeepInsight and 'modified' DeepInsight methods.

Comparative experiments with an MLP were also carried out [44]. MLPs are widely used in classification applications in many fields [45]. It is commonly used as a supervised classifier, including the data which are not linearly separable [46]. The optimal network had two hidden layers with 50 neurons each. The MLP "raw" architecture refers to raw data input, i.e., tabular data consisting of six variables,

and the MLP "modified" architecture refers to modified data input, i.e., tabular data with polynomial mapping (Figure 10). As for the DeepInsight method, increasing the number of variables increased the PPV. The feature engineering step had a greater impact on the PPV increase in the CNN than in the MLP-CNN, which, based on the images, finds correlations between features more effectively.

Subsequently, experiments were carried out to show the precision of the optimized model when tested on data from a different phantom than the training data.

Table 4. Comparison of method precision for different configurations: training vs. test model

Method	Training model	Precision [%]				Precision loss - average [%]
		NEMA IEC		XCAT		
	Test model	NEMA IEC	XCAT	NEMA IEC	XCAT	
DeepInsight "modified"		79.8	66.3	77.8	69.7	2.7
MLP "modified"		79.3	65.7	76.6	69.6	3.4

The average precision loss was calculated as the mean PPV decrease for the same test set tested on models trained on different phantom data (Table 4). The advantage of the DeepInsight "modified" model is the lower decrease in precision when testing on data from a different phantom than the training data, as compared to the MLP model. Additionally, the PPV obtained for the DeepInsight "modified" method was higher for all training phantom / testing phantom combinations.

4. Conclusions

The goal of this study was to develop a methodology for the transformation of 1-D data vectors into n -D matrices. These matrices are suitable for further analysis using tools for images with large numbers of pixels. In particular, the problem of processing 1-D vectors with a small number of features as compared to the number of pixels in the output images is discussed.

The proposed method, based on the DeepInsight methodology, was applied to the problem of the classification of PET coincidence events. Unlike DeepInsight, which employs an implicit nonlinear transformation through a kernel trick, our approach introduces a unique explicit nonlinear transformation. This transformation explicitly increases the dimensionality of the input data, resulting in a noticeable expansion of the points within the plane defined by the first two principal components. In the experimental section, it was shown that classification precision improved after the introduced modification of the general DeepInsight methodology. Increasing the number of features by using the 4th-degree polynomial mapping enhanced the classification performance. The PPV obtained by DeepInsight "modified" (0.798) improved by 0.9 percentage points relative to DeepInsight "raw" (0.789), by 0.5 percentage points relative to MLP "modified" (0.793) and by 2.0 percentage points relative to the single-path architecture (0.778).

The proposed method was tested on two different phantoms, i.e., NEMA IEC and XCAT. The results comparing the precision of models for each phantom show the universality of this method. The method was validated through comprehensive testing on multiple neural network architectures, with the DeepInsight CNN serving as a benchmark. The results demonstrate the superiority and improved performance of our method, showcasing its novelty and potential impact in the field. The clear value

added to DeepInsight by our work is an extension of the method to data obtained from instruments providing a limited number of features, such as reconstructed data from J-PET.

There are two major limitations of this study that could be addressed in future research. First, the developed method was designed to be applied to low-dimensional data. It is believed that the original method is more suitable for high-dimensional data; however, this approach is planned to be evaluated in supplementary studies. Second, further experiments should focus on optimizing the code to enable analysis for 3-D images. The results (Table 2) showed that the information stored in such images is greater than in 2-D images. Therefore, an increase in the efficiency of classification in the analysis of 3-D images can be expected.

Use of AI tools declaration

The authors declare that they have not used artificial intelligence tools in the creation of this article.

Acknowledgments

This work was supported by the Foundation for Polish Science through the TEAM POIR.04.04.00-00-4204/17 programs, the National Science Centre of Poland through grant nos. 2019/35/B/ST2/03562, 2021/42/A/ST2/00423 and 2021/43/B/ST2/021, the Ministry of Education and Science under the grant no. SPUB/SP/530054/2022, the Jagiellonian University via the project CRP/0641.221.2020, and via SciMat and qLife Priority Research Areas under the Strategic Programme Excellence Initiative. The contribution of Paweł Konieczka has been done in the frame of the National Centre for Research and Development Project, number POWR.03.02.00-00-I009/17 (Radiopharmaceuticals for molecularly targeted diagnosis and therapy, RadFarm. Operational Project Knowledge Education Development 2014–2020, co-financed by European Social Fund). BCH also acknowledges gratefully that this research was funded in whole, or in part, by the Austrian Science Fund (FWF) project P36102.

Conflict of interest

The authors declare that there is no conflict of interest.

References

1. Y. Lecun, Y. Bengio, G. Hinton, Deep Learning, *Nature*, **521** (2015), 436–444. <https://doi.org/10.1038/nature14539>
2. M. Z. Alom, T. M. Taha, C. Yakopcic, S. Westberg, P. Sidike, M. S. Nasrin, et al., A state-of-the-art survey on deep learning theory and architectures, *Electronics*, **8** (2019), 292. <https://doi.org/10.3390/electronics8030292>
3. A. H. Habibi, H. E. Jahani, *Guide to Convolutional Neural Networks: A Practical Application to Traffic-Sign Detection and Classification*, Springer International Publishing, 2017. <https://doi.org/10.1007/978-3-319-57550-6>

4. K. He, X. Zhang, S. Ren, J. Sun, Deep Residual Learning for Image Recognition, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, (2016), 770–778. <https://doi.org/10.1109/CVPR.2016.90>
5. J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You only look once: Unified, real-time object detection, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, (2016), 779–788. <https://doi.org/10.1109/CVPR.2015.7298594>
6. C. Szegedy, W. Liu, Y. Q. Jia, P. Sermanet, S. Reed, D. Anguelov, et al., Going deeper with convolutions, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, (2015), 1–9. <https://doi.org/10.48550/arXiv.1409.4842>
7. C. Szegedy, S. Ioffe, V. Vanhoucke, A. A. Alemi, Inception-v4, inception-ResNet and the impact of residual connections on learning, in *Proceedings of the AAAI Conference on Artificial Intelligence*, (2017), 4278–4284. <https://doi.org/10.48550/arXiv.1602.07261>
8. D. S. Kermany, M. Goldbaum, W. J. Cai, C. C. S. Valentim, H. Y. Liang, S. L. Baxter, et al., Identifying medical diagnoses and treatable diseases by image-based deep learning, *Cell*, **172** (2018), 1122–1131. <https://doi.org/10.1016/j.cell.2018.02.010>
9. S. Cheng, Y. F. Jin, S. P. Harrison, C. Quilodrán-Casas, L. C. Prentice, Y. K. Guo, et al., Parameter flexible wildfire prediction using machine learning techniques: Forward and inverse modelling, *Remote Sens.*, **14** (2022), 3228. <https://doi.org/10.3390/rs14133228>
10. Y. Zhuang, S. Cheng, N. Kovalchuk, M. Simmons, O. K. Matar, Y.-K. Guo, et al., Ensemble latent assimilation with deep learning surrogate model: application to drop interaction in a microfluidics device, *Lab Chip*, **22** (2022), 3187–3202. <https://doi.org/10.1039/D2LC00303A>
11. J. L. Humm, A. Rosenfeld, A. Del Guerra, From PET detectors to PET scanners, *European J. Nucl. Med. Mol. Imag.*, **30** (2003), 1574–1597. [10.1007/s00259-003-1266-2](https://doi.org/10.1007/s00259-003-1266-2)
12. D. L. Bailey, *Positron Emission Tomography: Basic Sciences*, Springer-Verlag, 2005. <https://doi.org/10.1007/b136169>
13. A. Alavi, T. J. Werner, E. L. Stepień, P. Moskal, Unparalleled and revolutionary impact of PET imaging on research and day to day practice of medicine, *Bio-Algor. Med-Syst.*, **17** (2021), 203–212. <https://doi.org/10.1515/bams-2021-0186>
14. E. Berg, S. Cherry, Using convolutional neural networks to estimate time-of-flight from PET detector waveforms, *Phys. Med. Biol.*, **63** (2018), 02LT01. <https://doi.org/10.1088/1361-6560/aa9dc5>
15. J. Bielecki, *Application of the machine learning methods to the multi-photon event classification in the J-PET scanner*, M.Sc thesis, Warsaw University of Technology, 2019. Available from: <https://pet.ncbj.gov.pl/wp-content/uploads/2019/10/JanBieleckiMasterThesis.pdf>
16. A. Sharma, E. Vans, D. Shigemizu, K. A. Boroevich, T. Tsunoda, Deepinsight: A methodology to transform a non-image data to an image for convolution neural network architecture, *Sci. Rep.*, **9** (2019), 11399. <https://doi.org/10.1038/s41598-019-47765-6>
17. P. Moskal, Sz. Niedźwiecki, T. Bednarski, E. Czerwiński, Ł. Kapłon, E. Kubicz, et al., Test of a single module of the J-PET scanner based on plastic scintillators, *Nucl. Instrum. Meth. Phys. Res. A*, **764** (2014), 317–321. <https://doi.org/10.1016/j.nima.2014.07.052>

18. L. Raczyński, P. Moskal, P. Kowalski, W. Wiślicki, T. Bednarski, P. Białas, et al., Compressive sensing of signals generated in plastic scintillators in a novel J-PET instrument, *Nucl. Instrum. Meth. Phys. Res. A*, **786** (2015), 105–112. <https://doi.org/10.1016/j.nima.2015.03.032>
19. P. Moskal, O. Rundel, D. Alfs, T. Bednarski, P. Białas, E. Czerwiński, et al., Time resolution of the plastic scintillator strips with matrix photomultiplier readout for J-PET tomograph, *Phys. Med. Biol.*, **61** (2016), 2025–2047. <https://doi.org/10.1088/0031-9155/61/5/2025>
20. S. Niedźwiecki, P. Białas, C. Curceanu, E. Czerwiński, K. Dulski, A. Gajos, et al., J-PET: A new technology for the whole-body PET imaging, *Acta Phys. Polon. B*, **48** (2017), 1567–1576. <https://doi.org/10.5506/APhysPolB.48.1567>
21. G. Korcyl, P. Białas, C. Curceanu, E. Czerwiński, K. Dulski, B. Flak, et al., Evaluation of single-chip, real-time tomographic data processing on FPGA—SoC devices, *IEEE Trans. Med. Imag.*, **37** (2018), 2526–2535. <https://doi.org/10.1109/TMI.2018.2837741>
22. P. Moskal, K. Dulski, N. Chug, C. Curceanu, E. Czerwiński, M. Dadgar, et al., Positronium imaging with the novel multiphoton PET scanner, *Sci. Adv.*, **7** (2021), eabh4394. <https://doi.org/10.1126/sciadv.abh4394>
23. P. Moskal, A. Gajos, M. Mohammed, J. Chhokar, N. Chug, C. Curceanu, et al., Testing CPT symmetry in ortho-positronium decays with positronium annihilation tomography, *Nat. Commun.*, **12** (2021), 5658. <https://doi.org/10.1038/s41467-021-25905-9>
24. R. D. Badawi, H. C. Shi, P. C. Hu, S. G. Chen, T. Y. Xu, P. M. Price, et al., First human imaging studies with the EXPLORER total-body PET scanner, *J. Nuclear Med.*, **60** (2019), 299–303. <https://doi.org/10.2967/jnumed.119.226498>
25. E. N. Holy, A. P. Fan, E. R. Alfaro, E. Fletcher, B. A. Spencer, S. R. Cherry, et al., Non-invasive quantification and SUVR validation of [18F]-florbetaben with total-body EXPLORER PET, *Alzheimer's Dement.*, **18** (2022), e066123. <https://doi.org/10.1002/alz.066123>
26. S. Vandenberghe, P. Moskal, J. S. Karp, State of the art in total body PET, *EJNMMI Phys.*, **7** (2020), 1–33. <https://doi.org/10.1186/s40658-020-00290-2>
27. A. Rahmim, M. Lenox, A. J. Reader, C. Michel, Z. Burbar, T. J. Ruth, et al., Statistical list-mode image reconstruction for the high resolution research tomograph, *Phys. Med. Biol.*, **49** (2004), 4239–4258. <https://doi.org/10.1088/0031-9155/49/18/004>
28. R. Accorsi, L.-E. Adam, M. E. Werner, J. S. Karp, Optimization of a fully 3D single scatter simulation algorithm for 3D PET, *Phys. Med. Biol.*, **49** (2004), 2577–2598. <https://doi.org/10.1088/0031-9155/49/12/008>
29. C. C. Watson, Extension of Single Scatter Simulation to Scatter Correction of Time-of-Flight PET, *IEEE Trans. Nucl. Sci.*, **54** (2007), 1679–1686. <https://doi.org/10.1109/TNS.2007.901227>
30. L. J. Maaten, G. Hinton, Visualizing High-Dimensional Data using t-SNE, *J. Mach. Learn. Research*, **9** (2008), 2579–2605. Available from: <https://www.jmlr.org/papers/volume9/vandermaaten08a/vandermaaten08a.pdf>.
31. B. Scholkopf, S. A. Bernhard, K. R. Muller, Nonlinear component analysis as a kernel eigenvalue problem, *Neural Comput.*, **10** (1998), 1299–1319. <https://doi.org/10.1162/089976698300017467>

32. M. A. Aizerman, E. M. Braverman, L. I. Rozonoer, Theoretical foundations of the potential function method in pattern recognition learning, *Autom. Remote Control*, **25** (1964), 821–837. Available from: <https://cs.uwaterloo.ca/~y328yu/classics/kernel.pdf>
33. V. Vapnik, *The Nature of Statistical Learning Theory*, Springer-Verlag, 1995. <https://doi.org/10.1007/978-1-4757-2440-0>
34. V. Vapnik, *Statistical Learning Theory*, Wiley, 1998. Available from: <https://www.wiley.com/en-ie/Statistical+Learning+Theory-p-9780471030034>
35. J. Mockus, Application of Bayesian approach to numerical methods of global and stochastic optimization, *J. Global Optim.*, **4** (1994), 347–365. <https://doi.org/10.1007/BF01099263>
36. S. Jan, G. Santin, D. Strul, S. Staelens, K. Assié, D. Autret, et al., GATE: A simulation toolkit for PET and SPECT, *Phys. Med. Biol.*, **49** (2004), 454–4562. <https://doi.org/10.1088/0031-9155/49/19/007>
37. D. Sarrut, M. Bała, M. Bardiès, J. Bert, M. Chauvin, K. Chatzipapas, et al., Advanced Monte Carlo simulations of emission tomography imaging systems with GATE, *Phys. Med. Biol.*, **66** (2021), 10TR03. <https://doi.org/10.1088/1361-6560/abf276>
38. J. Baran, W. Krzemien, L. Raczyński, M. Bała, A. Coussat, S. Parzych, et al., Realistic Total-Body J-PET Geometry Optimization–Monte Carlo Study, preprint arXiv e-prints, (2022), arXiv:2212.02285. <https://doi.org/10.48550/arXiv.2212.02285>
39. *NEMA Standards Publication NU 2-2007: Performance measurements of Positron Emission Tomographs*, Nat. Elect. Manuf. Assoc.,(2007). Available from: https://psec.uchicago.edu/library/applications/PET/chien_min_NEMA_NU2_2007.pdf
40. W. P. Segars, G. Sturgeon, S. Mendonca, J. Grimes, B. M. W. Tsui, 4D XCAT phantom for multimodality imaging research, *Med. Phys.*, **37** (2010), 4902–4915. <https://doi.org/10.1118/1.3480985>
41. P. Kowalski, W. Wiślicki, L. Raczyński, D. Alfs, T. Bednarski, P. Bialas, et al., Scatter fraction of the J-PET tomography scanner, *Acta Phys. Pol. B*, **47** (2016), 549–560. <https://doi.org/10.5506/APhysPolB.47.549>
42. M. Pawlik-Niedźwiecka, S. Niedźwiecki, D. Alfs, P. Bialas, C. Curceanu, E. Czerwiński, et al., Preliminary studies of J-PET detector spatial resolution, *Acta Phys. Polon. A*, **132** (2017), 1645–1648. <https://doi.org/10.12693/APhysPolA.132.1645>
43. P. Moskal, P. Kowalski, R. Y. Shopa, L. Raczyński, J. Baran, N. Chug, et al., Simulating NEMA characteristics of the modular total-body J-PET scanner - an economic total-body PET from plastic scintillators, *Phys. Med. Biol.*, **66** (2021), 175015. <https://doi.org/10.1088/1361-6560/ac16bd>
44. F. Murtagh, Multilayer perceptrons for classification and regression, *Neurocomputing*, **2** (1991), 183–197. [https://doi.org/10.1016/0925-2312\(91\)90023-5](https://doi.org/10.1016/0925-2312(91)90023-5)
45. H. Ramchoun, M. A. Janati Idrissi, Y. Ghanou, M. Ettaouil, Multilayer perceptron: Architecture optimization and training, *Int. J. Interact. Multim. Artif. Intell.*, **4** (2016), 26–30. <https://doi.org/10.9781/ijimai.2016.415>
46. A. Landi, P. Piaggi, M. Laurino, D. Menicucci, Artificial neural networks for nonlinear regression and classification, in *2010 10th International Conference on Intelligent Systems Design and Applications*, (2010), 115–120. <https://doi.org/10.1109/ISDA.2010.5687280>



AIMS Press

©2023 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0>)