1. Radosław WAJMAN, 2. Robert BANASIAK, 3. Maciej SPUTOWSKI, 4. Volodymyr MOSOROV

ORCID: 1. 0000-0002-6372-5960; 2. 0000-0002-1234-4949; 3. Student; 4. 0000-0001-6016-8671

OPEN ACCESS

DOI: 10.15199/48.2025.06.17

Supervised learning for lower urinary tract identification with limited and imbalanced training data

Uczenie nadzorowane do identyfikacji dolnych dróg moczowych przy użyciu ograniczonych i niezrównoważonych danych treningowych

***Abstract**. In recent years, the advantages of AI in data processing and analysis have become increasingly evident in the medical field. There has been a rapid growth in the application of AI in clinical medicine, including its use in urinary system disease detection. AI offers the capability to process and utilize diagnostic information, presenting new opportunities for precise and individualized treatment while promoting non-invasive diagnostic and therapeutic approaches. This paper introduces an original approach to addressing the supervised learning problem in convolutional neural network (CNN) models for lower urinary tract identification when confronted with a scarce and imbalanced training dataset. The proposed solution involves significantly expanding the diagnostic urinary bladder dataset by increasing the number of samples through various augmentation strategies. However, this approach also intensifies the computational complexity of AI training, rendering it infeasible to load all training datasets into memory simultaneously. To overcome this challenge, a distributed computing approach has been developed, incorporating dynamic loading of training data alongside programming-level memory optimization.*

***Streszczenie.** W ostatnich latach zalety sztucznej inteligencji (AI) w przetwarzaniu danych i diagnostyce medycznej stały się coraz bardziej widoczne. Wpływ na to mają postępy w technologii komputerowej i integracja wielu dyscyplin. Warto zauważyć, że nastąpił szybki wzrost zastosowań AI w medycynie klinicznej, w tym jej wykorzystania w technologiach wykrywania chorób układu moczowego. AI oferuje możliwość przetwarzania informacji diagnostycznych, co stwarza nowe możliwości precyzyjnego, spersonalizowanego leczenia, jednocześnie promując nieinwazyjne podejścia diagnostyczne. W niniejszym artykule przedstawiono oryginalne podejście do rozwiązania problemu uczenia nadzorowanego w modelach sieci neuronowych splotowych (CNN) do identyfikacji dolnych dróg moczowych w przypadku ograniczonego i niezrównoważonego zestawu danych treningowych. Proponowane rozwiązanie obejmuje znaczne rozszerzenie zestawu danych diagnostycznych dotyczących pęcherza moczowego poprzez zwiększenie liczby próbek za pomocą różnych strategii. Jednak podejście to zwiększa również złożoność obliczeniową treningu AI, co sprawia, że niewykonalne jest jednoczesne załadowanie wszystkich zestawów danych treningowych do pamięci. Aby przezwyciężyć to wyzwanie, opracowano podejście rozproszonego przetwarzania, obejmujące dynamiczne ładowanie danych wraz z optymalizacją pamięci.*

Keywords: supervised machine learning; convolutional neural networks; urinary tract; global consistency error; image segmentation
Słowa kluczowe: uczenie maszynowe nadzorowane; splotowe sieci neuronowe; dolne drogi moczowe; globalny błąd spójności; segmentacja obrazu

1. Introduction

Functional urinary tract disorders are a common issue in the human population. Urinary system defects are often asymptomatic; therefore, regular monitoring of disease progression is essential to determine whether conservative treatment is sufficient or if surgical intervention is required. Consequently, performing magnetic resonance imaging (MRI) or computed tomography (CT) is necessary for accurate diagnosis. Lower urinary tract diseases encompass both anatomical abnormalities and functional disorders. Voiding cystourethrography (VCUG) is considered the "gold standard" for anatomical evaluation of the bladder and urethra, while urodynamic examination is employed to diagnose urinary tract dysfunctions. However, both diagnostic methods are invasive and frequently challenging, particularly due to the lack of cooperation from paediatric patients during the procedures. When repeat testing is required, children are exposed to ionising radiation (VCUG), an increased risk of urinary tract infections, and potential damage to the lower urinary tract caused by catheter insertion.

Although urinary system diagnostic tests are generally well tolerated by adults, they can be difficult to be performed in children. Physicians must carefully assess the necessity of repeating these tests to monitor disease progression. Consequently, there is a pressing need to develop new, less invasive diagnostic methods to enhance understanding of urinary tract diseases in paediatric patients.

The absence of non-invasive diagnostic techniques and the lack of a comprehensive analysis of the urinary tract significantly reduce the likelihood of accurate diagnosis and effective treatment. Current measurement methods yield unreliable results. The integration of PET/CT and PET/MRI with various radiopharmaceuticals presents a promising approach for characterising the urinary tract. PET/MRI, in particular, holds significant future potential due to its superior soft tissue contrast, which may improve urinary tract diagnosis. However, at present, the sensitivity of PET/MRI remains insufficient for the diagnosis of urinary diseases.

In summary, PET/CT and PET/MRI represent promising imaging modalities in urinary tract diagnostics, offering greater accuracy than conventional CT. Future clinical trials involving novel radiopharmaceuticals and machine-learning-driven PET technologies have the potential to advance early disease detection, staging, monitoring, and precision medicine approaches. In recent years, artificial intelligence (AI) has been widely applied to data processing and model analysis, including in the medical field. John McCarthy first proposed AI at the Dartmouth Conference in 1956 in New Hampshire, introducing it as a technological science dedicated to researching and developing theories and methods for intelligent systems. AI enables computers to simulate human behaviour and extend human capabilities beyond direct instructions. The application of AI in medicine has progressively deepened and broadened [1], particularly with the rapid development of clinical medicine intelligent decision-making. One of its key advantages is the ability to enhance diagnostic accuracy and efficiency based on medical data. Significant efforts have been made to address challenges in medical image analysis, including organ detection [2], segmentation, and the identification of disorders [3], [4] and diseases [5], [6] using AI. Urinary tract diagnostics also benefit from these advancements in clinical decision-making.

The integration of machine learning (ML), particularly supervised learning, into urinary tract diagnostics represents a transformative step towards more reliable and less invasive diagnostic methods. Supervised learning has demonstrated a remarkable efficacy in medical image segmentation, facilitating organ and pathology detection. However, a major limitation in its application to lower urinary tract (LUT) identification arises from the scarcity and imbalance of annotated training data. In medical imaging, obtaining pixel-wise annotations requires expert knowledge, making the process both time-consuming and costly [7]. Consequently, models trained on small, imbalanced datasets often suffer from overfitting, leading to poor generalisation on unseen data [8].

To mitigate the impact of data scarcity, various techniques have been proposed, including semi-supervised learning, data augmentation, and contrastive learning. Semi-supervised methods leverage unlabelled data to improve segmentation accuracy. Recent advancements, such as Dual-debiased Heterogeneous Co-training (DHC), utilise pseudo-labels to guide model learning while addressing class imbalance [8]. However, despite its advantages, DHC heavily relies on the quality of pseudo-labels, which may introduce errors when generated from small or noisy datasets. Furthermore, its dependence on distribution-aware debiased weighting assumes a relatively stable class distribution, making it less effective in highly heterogeneous datasets. These limitations hinder its scalability across diverse medical imaging tasks and necessitate refinements in pseudo-label verification and bias mitigation strategies. Similarly, Multi-Scale Cross Contrastive Learning (MCSC) enhances the segmentation of small or infrequent anatomical structures by enforcing feature consistency across different scales [9]. While this method improves robustness through multi-scale representations, it presents challenges. The reliance on contrastive learning requires large-scale training to effectively capture meaningful inter-slice relationships, which is impractical for highly imbalanced medical datasets. Additionally, the model's performance deteriorates when class distributions are significantly skewed, favouring dominant classes while underrepresenting rare pathological features. Addressing these issues requires tailored sampling strategies and adaptive weighting mechanisms to ensure fair learning across all anatomical structures.

One primary area of focus is improving data augmentation techniques. Traditional augmentation methods, such as affine transformations and intensity perturbations, often fail in MRI imaging due to the strict spatial constraints of LUT scans. Task-driven data augmentation techniques, which generate synthetic intensity and shape variations tailored for medical images, have shown promise in overcoming these limitations [10]. Additionally, self-supervised contrastive learning can improve segmentation by ensuring robust feature representation, particularly in low-data regimes [11].

Overfitting remains a persistent issue due to the limited dataset size. Regularization techniques such as dropout and weight decay are integrated with self-supervised pretraining to mitigate this. Self-supervised learning (SSL) enables models to learn high-level feature representations from unlabelled MRI scans, enhancing their ability to generalise on small, annotated datasets. Furthermore, curriculum learning strategies, wherein networks first learn simpler segmentation tasks before progressively addressing complex structures, can enhance model stability and reduce overfitting [12]. However, while curriculum learning improves convergence, it may introduce biases by prioritising simpler tasks, potentially leading to suboptimal performance on complex anatomical structures. Additionally, the reliance on predefined task complexity metrics may limit the adaptability of this approach in highly heterogeneous datasets. Enhancing robustness through domain-specific priors is another critical aspect. Rather than relying solely on large-scale supervised learning, integrating anatomical priors via transformer-based architectures and hybrid CNN-transformer models can improve segmentation accuracy [13]. While these architectures offer superior spatial attention mechanisms, their increased complexity demands significantly greater computational resources, which may not be feasible in real-world clinical applications. Additionally, transformers are highly data-dependent, and their performance tends to degrade in low-data scenarios, making them less effective when only limited annotated MRI scans are available.

The integration of supervised learning with advanced augmentation techniques presents a viable pathway for improving LUT identification despite data constraints. In this study, as a part of the research project [14]–[16], the authors faced the challenge of training a neural network model to recognise lower urinary tract organs using an exceptionally small number of MRI source images to construct a meaningful training set. In such cases, data augmentation techniques are typically employed to enhance model performance. However, the unique characteristics of MRI imaging introduced additional constraints. MRI scans centre the patient's body in the image, permitting minimal spatial transformations, with rotation tolerances restricted to no more than two degrees in any direction. Additionally, the source dataset exhibited irregularities in sex, pathological conditions, and resolution, which, under normal circumstances (i.e., with a sufficiently large dataset), could be advantageous. Despite these challenges, the conducted research and applied methodologies demonstrated the feasibility of developing an adaptive neural network model capable of detecting lower urinary tract organs.

The research findings and subsequent discussion addressed the following research questions:

1. What impact do data augmentation techniques have on improving the generalisation of neural networks in lower urinary tract organ segmentation?
2. What strategies can mitigate overfitting when training deep learning models with small and highly heterogeneous datasets?

This paper is structured as follows. Section 2 defines the characteristics of the training data and outlines the data preparation procedure. Section 3 details the AI model designs and the implemented algorithm optimisation and parallelisation mechanisms. Section 4 presents the conducted experiments and discusses the obtained results. Finally, the last section provides a summary of the study's conclusions.

2. Methods for Processing and Organizing Training Data

This chapter outlines the medical source data's origin, format, and specifications. The following subsections detail the methods used for selecting and preparing the training dataset.

2.1. Data description and selection

The medical dataset used in this study was both imbalanced and limited. It was provided in the form of 51 DICOM files containing MRI (Magnetic Resonance Imaging) scan results of the lower abdomen from male and female patients of various ages, including children. The images exhibited significant diversity, having been acquired through different imaging techniques, such as T2W TSE (T2-

weighted Turbo Spin Echo) and mDixon (point Dixon). Additionally, the images varied in scale, size, and anatomical positioning.

It is important to note that DICOM images differ from standard image formats. In these files, pixel values occupy 16 bits of memory, of which only 12 bits are utilized for data representation. Consequently, the pixel intensity values range from 0 to 4095, as opposed to the conventional 0 to 255. Moreover, all images contained only a single channel.

Many of the images were deemed unsuitable for training purposes. Some were of poor quality, while others were excessively distinct from the majority of the dataset. Additionally, certain images did not contain the bladder at all. Ultimately, only 19 images were selected as viable for use in the neural network training process.

2.2. Data annotation

Supervised neural network training requires each image to be associated with a corresponding ground truth—an expected output in the form of a segmentation mask. This mask consists exclusively of pixels assigned to a value of either 1 or 0, where the former represents the bladder region and the latter denotes the background.

To generate these masks, a clustering process was applied to differentiate the bladder from surrounding anatomical structures. The k-means algorithm [17] was selected for this task due to its computational efficiency and adaptability, as the number of clusters (k) does not need to be predefined but can instead be determined empirically. Although the clustering process was applied differently to each image, the optimal k value was consistently found to lie between 6 and 10.

Following clustering, only one cluster within the segmented image contained the bladder. This cluster was extracted for further processing. A manual annotation process was performed in the next step to ensure accurate segmentation. A shape was manually placed around the bladder to encompass the entire organ. For this purpose, a dedicated Picking Tool was developed, providing a graphical user interface with functionalities designed to enhance the visibility and annotation of the bladder within the image.

Finally, the segmentation mask was generated as a blank image, in which the region corresponding to the bladder was filled with pixels assigned a value of 1, while all other pixels remained 0. This annotation process was repeated for every image included in the training dataset. A visual representation of the workflow for this procedure is shown in Figure 1.

2.3. Data augmentation

Given the minimal number of available samples, it was necessary to augment the dataset extensively. Each image was subjected to every possible combination of the following operations and their respective parameters:

- Translation along the x and y axes: [-4, 0, 4] pixels,
- Rotation along the x, y, and z axes: [-3, 0, 3] degrees,
- Scaling: [0.95, 1.0, 1.05],
- Intensity adjustment: [0.9, 1.0, 1.1].

Additionally, random noise was added to each augmented image. The noise values were sampled from a normal distribution with a mean of 0.0 and a standard deviation of 10.0. All resulting samples were normalised and reshaped to a uniform size of (256, 256, 96), as required by the network. These dimensions were selected because they closely approximate the dataset's average dimensions (330, 330, 134) and can be repeatedly divided by two without remainder. This property eliminates the need for cropping or padding during downsampling and upsampling.

The corresponding segmentation masks were augmented alongside their respective images. However, they were only subjected to translations, rotations, and scaling, as modifying voxel values in binary masks would be redundant.

The augmented dataset was systematically stored on a hard drive. Augmenting each image generated 2,187 new samples, increasing the total dataset size to 41,553 images. These images and their associated masks occupied approximately 1.8 TB of disk space.

3. Training and Optimizing CNN Models: Methods and Challenges

The sheer size of the augmented dataset made it impractical to load into memory in its entirety. Instead, samples were dynamically loaded during training. Initially, the dataset was represented as a list of filenames, which was then randomised and split into training, validation, and test subsets using the standard 80/20 split, with validation and test sets each receiving 10% of the total samples. For each subset, an input pipeline was constructed, consisting of the following sequential stages: file listing → shuffling → transformation (file loading) → batching → prefetching.

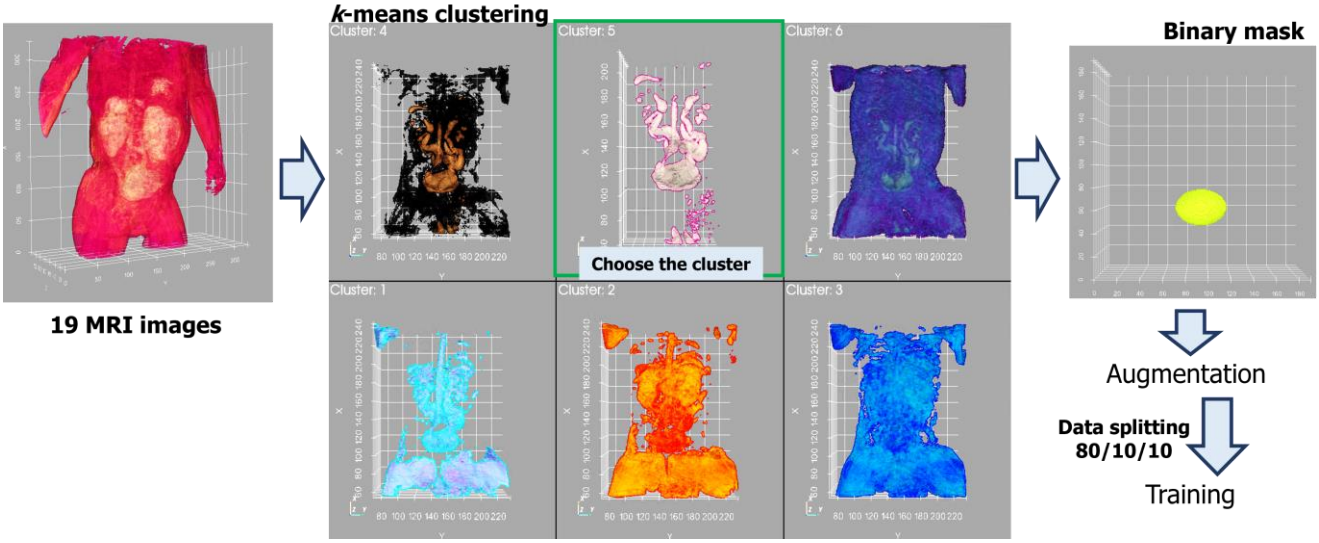

Fig. 1. Procedure developed for training dataset preparation

The first step in the pipeline involves shuffling the dataset. Although the data was already randomised before the split, additional shuffling was necessary to ensure variability after each epoch.

The transformation step replaces filenames with their corresponding loaded images and masks. Since 3D convolutions require a four-dimensional (4D) input, both the image and the mask were expanded to a shape of (256, 256, 96, 1), where the final dimension represents the single-channel nature of the images. This transformation step follows shuffling because shuffling can only be performed on data already loaded into memory. Given the dataset's size, it was not feasible to load all images simultaneously.

The next stage is batching. Grouping samples into batches accelerates training, as the network can process multiple records simultaneously using optimised matrix operations—an area where GPUs excel. However, excessively large batch sizes can degrade model performance [18]. In this study, the batch size was limited to 2 due to the large image dimensions. This constraint balances computational efficiency and hardware limitations, ensuring that training proceeds without exceeding memory capacity while still benefiting from batch processing optimisations.

The final stage of the pipeline is prefetching. Unlike previous operations, prefetching is a performance optimisation mechanism rather than a transformation. It prepares the next batch of data while the current batch is being processed, efficiently utilizing computational resources. Since data loading primarily relies on the CPU while network training is GPU-intensive, prefetching maximises throughput by overlapping data preparation with model computation.

An additional caching mechanism was considered to accelerate training beyond the first epoch. However, caching disrupts data randomness, as it preserves the original input order, leading to repeated exposure to the same data sequence. Given that caching would only yield a marginal (5–10%) speed improvement, it was deemed an inadequate trade-off against potential reductions in training quality.

3.1. Optimization

Compared to conventional images, volumetric images are significantly larger. Before shape alignment, the smallest image had dimensions of (190, 190, 80), while the largest measured (450, 450, 200). After normalisation, all images were resized to (256, 256, 96), with each file occupying 24 MB of storage. The maximum feasible batch size was determined to be 2, which coincides with the original V-Net implementation. However, V-Net's input dimensions were notably smaller at (128, 128, 64) [19]. Batch size is a key factor influencing both performance and training time, and larger powers of two (e.g., 32) are often recommended as a starting point [20].

Initially, data augmentation was performed on the fly within the input pipeline. However, this approach proved computationally prohibitive, as a single epoch required several days to complete. Consequently, augmentation was conducted as a one-time preprocessing step, with the transformed images stored and loaded directly during training.

A second inefficiency stemmed from retaining images in their original dimensions and resizing them dynamically during preprocessing. This approach was abandoned in favour of incorporating resizing into the data augmentation process, thereby standardising image dimensions before training. The most significant optimisation was the implementation of distributed training across ten GPUs. This was achieved using a mirrored strategy, an abstraction for multi-GPU training on a single machine. In this approach, all model variables are replicated across GPUs, and gradients are computed in parallel before being aggregated using the Hierarchical AllReduce algorithm [21]. The synchronised model update is then distributed across all GPUs, and the process repeats.

Under this scheme, each model replica processes a separate batch, increasing the number of samples trained per iteration from 2 to 20. Although synchronisation overhead slightly reduces the efficiency gain, the overall training time per epoch was reduced by a factor of approximately eight.

Lastly, the data type of augmented images and masks was adjusted. By default, NumPy uses float64, but float32 was deemed sufficient for DICOM images without compromising precision. This simple modification reduced dataset storage requirements by 50%. Before these optimisations, training required several weeks of continuous processing. After implementing all improvements, the total training time was reduced to 8–14 hours, depending on model complexity.

3.2. Neural networks architectures

Three neural network architectures were compared to determine the most effective model for bladder localisation in MRI images. Each utilised ReLU [22] as the activation function for convolutional layers and Adam [23] as the optimiser. The primary distinctions between the models lay in their complexity and the presence or absence of mechanisms such as skip connections and element-wise additions.

The first model, an FCN architecture (Figure 2), consists of downsampling and upsampling paths.

- The downsampling path employs convolutional layers and max-pooling to reduce image dimensionality while retaining essential features. It comprises two blocks of 3D convolutional layers, each followed by batch normalisation, an optional dropout layer, and a 3D max-pooling layer.
- The upsampling path restores the image to its original dimensions via transposed convolutions and upsampling layers. It consists of two blocks of 3D transposed convolutional layers, each followed by batch normalisation, an optional dropout layer, and a 3D upsampling layer.

The convolutions use a 3×3×3 kernel with a stride of 2, while both pooling and upsampling operations utilise a 2×2×2 kernel with a stride of 2. The final layer is a 1×1 3D convolutional layer with a Sigmoid activation function, producing a binary voxel-wise output identical in size to the input image. This model comprises 56,033 trainable parameters.

The second architecture (Figure 3) is the 3D adaptation of the original U-Net [24]. Compared to its 2D counterpart, it contains only two convolutional blocks in both its downsampling and upsampling paths. The key feature of this architecture is its skip connections, which concatenate feature maps from the contracting path (left side of the network) with those from the expanding path (right side). This mechanism facilitates the transmission of additional information throughout the network, resulting in smoother and more precise segmentations compared to the Fully Convolutional Network (FCN).

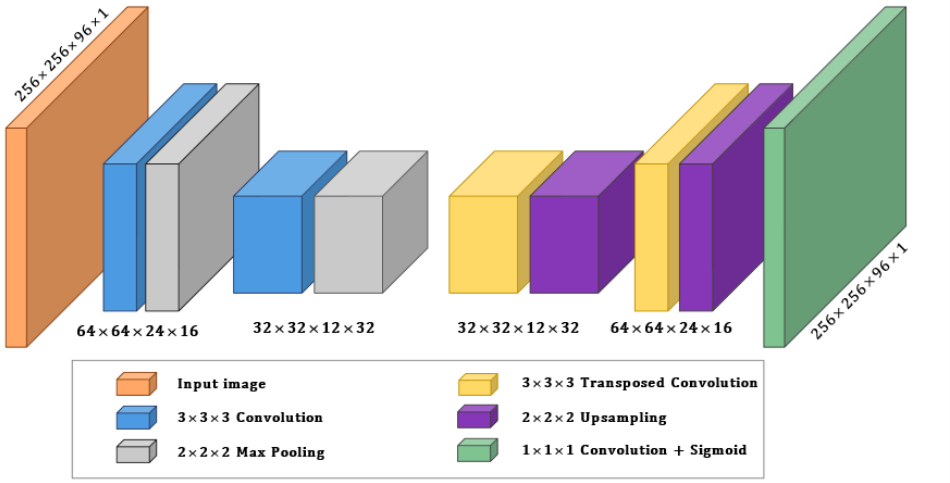

Fig. 2. Architecture of the basic 3D Fully Convolutional Network

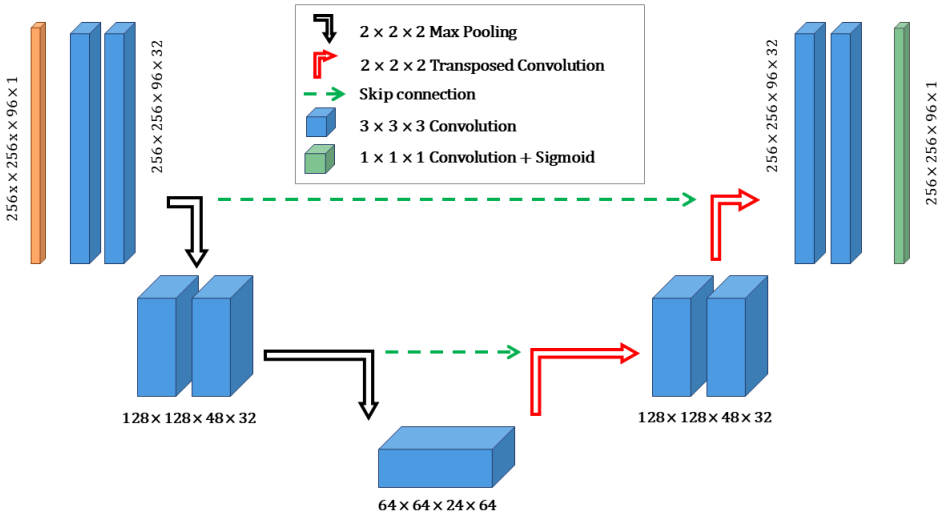

Fig. 3. Architecture of the custom 3D version of U-Net

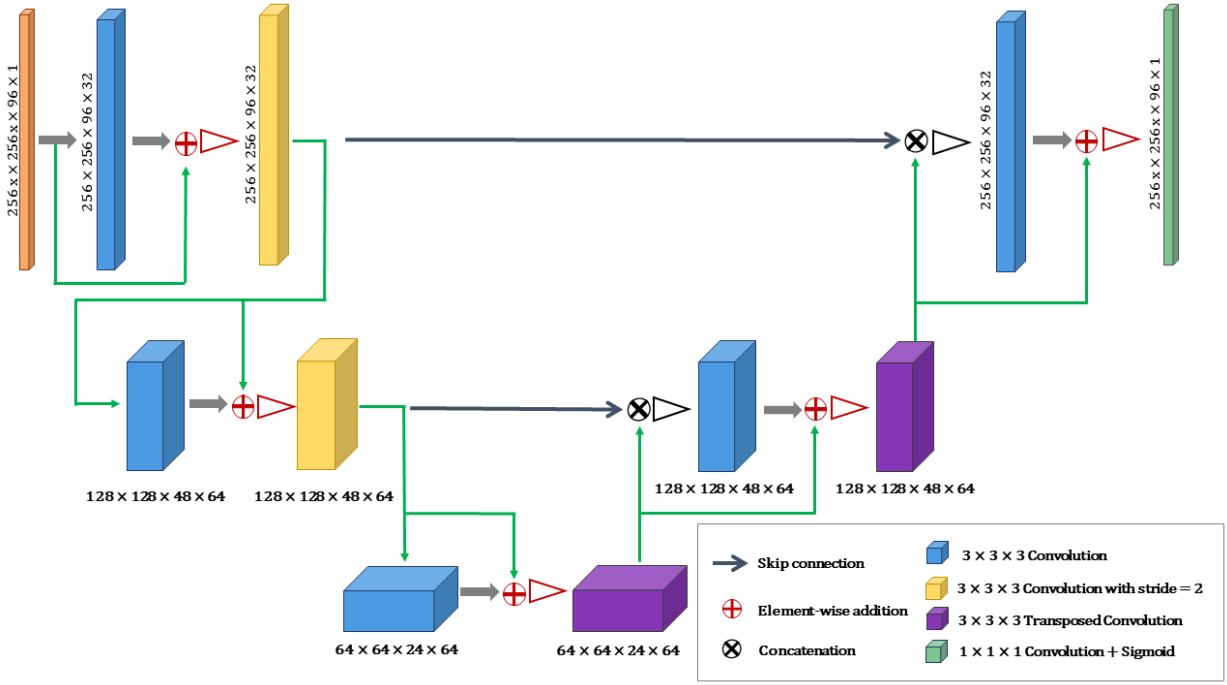

Fig. 4. Architecture of the custom version of V-Net

As with the previous architecture, batch normalisation layers are applied after each convolutional layer, followed by a dropout layer where applicable. Each convolutional block is followed by either a pooling or an upsampling layer. The convolutional, pooling, and upsampling layer parameters remain consistent with those of the previous model, as does the final layer. This model contains 339,889 trainable parameters.

The third and final model (Figure 4) is a custom implementation of the V-Net architecture [19]. Like U-Net, it utilises skip connections to transfer feature information between the two sides of the network. Additionally, it incorporates small-scale skip connections within each convolutional block, enabling more features to be preserved as images pass through the network.

Unlike the previous models, V-Net does not employ max-pooling and upsampling layers. Instead, it integrates a second convolutional layer within each block, using a stride of 2. This layer performs a similar function but possesses trainable weights, allowing for improved adaptability during training.

All convolutional layers, except the final one, utilise ReLU as the activation function. Compared to the original V-Net, this implementation uses 3×3×3 kernels instead of 5×5×5 kernels. The larger kernel size was found to introduce an excessive number of parameters, significantly increasing training time beyond acceptable limits. The final layer remains identical to those in the previous models. This model consists of 220,993 trainable parameters.

3.3. Loss functions

To effectively train the model, it is necessary to quantify the difference between its predicted output and the expected ground truth and subsequently minimise this error [25]. Selecting an appropriate loss function is crucial for ensuring proper network training. This study compares several loss functions commonly used in image segmentation tasks.

Several abbreviations are used in the loss function formulas and throughout the paper: True Positives (TP): Number of voxels correctly identified as part of the bladder, True Negatives (TN): Number of voxels correctly identified as part of the background, False Positives (FP): Number of voxels incorrectly identified as part of the bladder (should have been classified as background) and False Negatives (FN): Number of voxels incorrectly identified as part of the background (should have been classified as bladder).

The Binary Cross-Entropy loss function [26] measures the difference between the predicted and true probability distributions in binary classification tasks. Semantic segmentation with two classes can be conceptualised as a per-pixel binary classification problem. For multi-class cases, the generalised version of this loss function, known as Categorical Cross-Entropy, is used instead.

Focal Loss [27] is designed to improve classification accuracy by emphasizing hard-to-classify examples more while downweighting the contribution of correctly classified, easy examples. This is achieved through the inclusion of a modulating factor $(1 - p_t)^{\gamma}$ in the Cross-Entropy loss function.

Dice Loss [28] is a region-based loss function widely employed in image segmentation. It measures the relative overlap between the predicted and ground truth segmentations. Crucially, Dice Loss is independent of region size, making it particularly valuable in imbalanced segmentation tasks, where the target object occupies only a small fraction of the image.

Tversky Loss [29] generalises Dice Loss by introducing weighting factors to adjust the relative impact of false positives and false negatives. This function is especially useful in medical imaging applications, where certain types of misclassifications (e.g., false negatives in disease detection) carry greater consequences than others.

3.4. Overfitting

Overfitting is a common challenge in deep learning, occurring when a model becomes excessively specialised to the training data, thereby losing its ability to generalise to unseen data. This phenomenon can be identified by monitoring the training and validation loss dynamics. If the training loss decreases while the validation loss increases, it is a clear indication of overfitting. Several techniques have been implemented to mitigate this issue.

Firstly, expanding the dataset through data augmentation helps combat overfitting by increasing the diversity of training samples. In image-based tasks, augmentations such as translations, rotations, and brightness adjustments generate new variations of existing images. A larger, more diverse dataset reduces the likelihood of the model memorising specific training examples, thereby improving its generalisation ability. Although primarily used to counteract overfitting, data augmentation is generally beneficial for training deep learning models.

Next, a well-trained model should remain robust even in the presence of minor imperfections in input data. In the case of medical images, slight blurriness or missing pixels should not significantly affect the network's ability to classify or segment correctly.

To improve robustness, many models incorporate dropout layers, which are active only during training. These layers function by randomly setting a fraction of the input units to zero, effectively preventing the model from becoming overly reliant on specific features. The dropout rate is expressed as a fraction of affected inputs. For example, setting dropout = 0.1 means that 10% of the inputs to the subsequent layer will be randomly nullified during training. Dropout has been shown to enhance the generalisation ability of neural networks, but its effect must be carefully tuned to achieve an optimal balance.

Finally, determining the optimal number of training epochs is critical. Training for too few epochs may result in an underfitted model that fails to capture important patterns. Training for too many epochs increases the risk of overfitting. One approach is to determine the best number of epochs experimentally. However, a more effective solution is to use early stopping, which automatically halts training when model performance ceases to improve. The most appropriate metric for early stopping is the validation loss function, as it provides a reliable indicator of how well the model generalises to unseen data.

Due to the limited dataset size and training computational expense of various architectures under data augmentation and distributed learning, none of the automated hyperparameter tuning approaches were utilized. Rather, learning rate and dropout were manually selected by taking reasonable values from exploratory experiments. This enabled controlled comparisons among the models without the risk of overfitting and maintaining interpretability in the low-data scenario.

4. Experiments and discussion

4.1. Metrics

Metrics are quantitative performance indicators used to monitor and evaluate a model's effectiveness. Unlike loss functions, which are integral to the learning process, metrics serve to assess and describe the model's behaviour.

Selecting the appropriate metrics is crucial for identifying and resolving issues during training [30].

Precision and recall were chosen due to their simplicity and intuitive interpretation. Additionally, Global Consistency Error (GCE) [31] and Volumetric Similarity (VS) [32] were employed. GCE quantifies the discrepancy between two segmentations, while VS measures the similarity between volumetric structures. Both metrics are widely used in medical image segmentation tasks [18].

Precision is the ratio of true positives (TP) to the total number of predicted positives (TP + FP). It reflects the reliability of the model's predictions, where a high precision value indicates that most positive predictions are correct, even if not all positives are detected.

Recall is the ratio of true positives to the sum of true positives and false negatives (TP + FN). It measures the model's effectiveness at detecting all positive cases, where a high recall value suggests that the model successfully identifies most positives, even if some predictions are incorrect.

Global Consistency Error (GCE) evaluates the region-wise inconsistency between the prediction mask and the ground truth. The formula for binary segmentation is expressed as:

$$GCE = \frac{1}{N} min \left\{ \frac{FN(FN+2TP)}{TP+FN} + \frac{FP(FP+2TN)}{TN+FP}, \frac{FP(FP+2TP)}{TP+FP} + \frac{FN(FN+2TN)}{TN+FN} \right\} \quad (1)$$

where N represents the total number of voxels. Notably, GCE is not symmetric and must be computed in both directions (prediction vs. ground truth and vice versa), with the smaller value being selected.

Volumetric Similarity measures how closely two volumetric segmentations match in terms of shape and size, independent of spatial placement within the image. The VS formula is:

$$VS = 1 - \frac{|FN - FP|}{2TP + FP + FN} \quad (2)$$

A high VS value suggests a successful segmentation when combined with high scores in other metrics. Notably, VS can reach its maximum value of 1 even when one of the volumes is disjointed from the actual region of interest.

4.2. Evaluation

Given the computational demands of training, testing all possible combinations of network architectures, loss functions, and hyperparameters was impractical. Instead, exploratory training sessions were conducted, varying a single parameter at a time. Although this approach did not guarantee an optimal configuration, it provided valuable insights into model behaviour under different conditions.

The dataset size was reduced to 20% of its original capacity to expedite training while maintaining a representative sample. The selected subset remained consistent across experiments, with two images excluded and allocated evenly to the validation and test sets. Each training session lasted 20 epochs, ensuring meaningful comparisons while mitigating overfitting risks.

The basic FCN architecture was used to evaluate different loss functions with a fixed learning rate of 0.001 and a dropout rate of 0.2. The Tversky Loss was tested twice:

- $\alpha > \beta$, placing greater emphasis on false positives,
- $\alpha < \beta$, penalising false negatives more heavily.

Table 1. Performance metrics for different loss functions, results after 20 training epochs

Loss function	GCE	VS	Precision	Recall
BCE	0.4203	0.3537	0.6863	0.0877
Focal Loss	0.7077	0.6564	0.7896	0.1193
Dice	0.3812	0.2473	0.8324	0.084
Focal Dice	0.3774	0.2073	0.8897	0.0722
Tversky (α=0.3, β=0.7)	0.3951	0.2708	0.7099	0.0818
Tversky (α=0.7, β=0.3)	0.3920	0.3956	0.7608	0.1501

The exceptionally low loss values for BCE and Focal BCE are misleading, as they fail to capture spatial overlap between the predicted segmentation and the ground truth. Given that the bladder occupies only a small fraction of the volumetric image, predicting the entire volume as background results in a deceptively low loss value, despite poor segmentation performance.

Dice Loss provided more predictable results, stabilising at approximately 0.75 during validation. However, Focal Dice emerged as the best performer, achieving the highest precision and GCE scores, albeit with slightly lower recall and VS values.

Among the Tversky Loss variants, the α=0.7, β=0.3 configuration yielded notable results. Although its training curve remained relatively flat, it ultimately outperformed the α=0.3, β=0.7 variant in both precision and recall. These findings suggest penalising false positives more than false negatives leads to slower convergence and better overall performance. Based on these observations, Focal Dice was selected as this dataset's most effective loss function.

Next, the three CNN architectures (FCN, U-Net, and V-Net) were evaluated using Dice Loss with a dropout rate of 0.2. The default learning rate (0.001) was adjusted to 0.0001, 0.0005, and 0.002 to analyse its impact. Table 2 presents the final metric values after 20 training epochs across different architectures.

The decision to limit training to 20 epochs was motivated by the risk of overfitting, which is particularly pronounced when working with a small and imbalanced dataset. In such cases, the model's generalisation ability is inherently constrained, and prolonged training increases the likelihood of memorising training data rather than learning meaningful patterns.

The V-Net model exhibited significant instability, characterised by a sudden increase in training loss and persistently high validation loss, preventing meaningful convergence. As a result, the model failed to generate valid segmentation masks in certain cases, leading to undefined (NaN) or zero-valued metric scores. This behaviour highlights the incompatibility of V-Net with the selected training settings, further justifying its exclusion from subsequent evaluations.

In the subsequent phase of the study, the fundamental FCN architecture was employed in conjunction with the Dice Loss function and an initial learning rate of 0.001 to determine the optimal dropout rate [33]. Initially, the dropout rate was set to 0.2, a value previously identified as a suitable starting point for CNN-based networks [34]. Subsequently, this parameter was increased to 0.4, a value generally considered excessive, then decreased to 0.1. Additionally, a variant without dropout was tested. The final metric values after 20 epochs are presented in Table 3.

The influence of the dropout rate on the training loss curves is not immediately evident. However, the validation loss values and corresponding metrics are significantly affected by this parameter, decreasing as the dropout rate is reduced and reaching their lowest values in the absence of dropout. The trends in precision and recall indicate that precision increases as the dropout rate rises, whereas recall declines. The baseline model achieves the most

balanced trade-off between these two metrics, corresponding to a lower Focal Dice loss value. As precision and recall diverge further with increasing dropout, the loss function values also increase. Volumetric Similarity (VS) follows a pattern similar to recall, whereas Global Consistency Error (GCE) exhibits a distinct trend: it begins at a high value, decreases to its minimum at a dropout rate of 0.2, and then increases again.

Table 2. Performance metrics for different learning rates

Model	GCE	VS	Precision	Recall
Learning rate 0,0001				
FCN	**0.3774**	0.2073	**0.2073**	0.2073
U-Net	0.9378	**0.3431**	0.3431	**0.3431**
V-Net	0.4302	0.2459	0.2459	0.2459
Learning rate 0,0005				
FCN	**0.3856**	0.342	**0.7759**	0.1257
U-Net	0.4870	**0.4125**	0.4471	**0.2447**
V-Net	0.3951	0.085	0.287	0.0098
Learning rate 0,001				
FCN	**0.3686**	0.2597	**0.9452**	0.102
U-Net	0.5515	**0.4295**	0.3992	**0.3125**
V-Net	0.3898	0.0697	0.4259	0.0128
Learning rate 0,002				
FCN	0.3827	0.3346	0.7982	0.1246
U-Net	**0.3722**	**0.4156**	**0.8036**	**0.1754**
V-Net	NaN	0	0	0

Table 3. Performance metrics for different dropout factors

Variant	GCE	VS	Precision	Recall
No dropout	0.5974	**0.5151**	0.3957	**0.3559**
0.1	0.3926	0.5054	0.7615	0.223
0.2	**0.3686**	0.2597	0.9452	0.102
0.3	0.3764	0.071	0.9829	0.0231
0.4	0.3767	0.0134	**1**	0.0041

For brevity, Table 3 presents results exclusively for the FCN model. Notably, dropout exhibited a comparable influence across all tested architectures, with variations in prediction performance remaining within the 36%–42% range. As such, including results for U-Net and V-Net would not provide additional insights beyond those already discussed.

4.3. Main training

The findings from the exploratory training exercises indicate that V-Net is not a viable candidate for further consideration, as it has demonstrated a clear lack of performance. The suboptimal outcomes observed for this architecture are likely attributable to modifications introduced to enhance computational efficiency, such as reducing the kernel size and decreasing the number of convolutional blocks. It has been done to limit the number of trainable parameters and reduce model complexity. While this adjustment was made to improve generalisation in the context of a small training set, it did so at the cost of slightly reducing segmentation accuracy. This is the unavoidable trade-off for the general problem of balancing model expressiveness against overfitting risk when training on highly constrained data. These results illustrate the importance of architectural optimisation in low-data environments. Therefore, the relatively modest complexity of the problem may have contributed to V-Net's poor performance.

Among the remaining two models, U-Net exhibits a slight advantage over FCN. However, the disparity in performance is minimal, warranting further evaluation of both models on the entire dataset. Regarding hyperparameters, the findings from the previous section do not consistently indicate a single superior configuration.

Concerning the dropout rate, both 0.2 and 0.1 demonstrate strong performance. However, 0.1 was selected due to its superior balance between precision and recall, its twofold increase in VS, and only a marginal difference in GCE. Regarding the learning rate, the results suggest that different values are optimal for each model. FCN performed best with a learning rate of 0.001 and U-Net achieved optimal performance with a learning rate of 0.0005. These considerations are summarised in Figure 5.

The entire dataset was utilized for the primary training phase, with samples divided in the same manner as in the exploratory training sessions. The final metrics results are provided in Table 4. As in previous experiments, validation loss fluctuations precluded the application of an early stopping strategy. Moreover, neither network exhibited clear signs of overfitting, making it difficult to determine an optimal number of training epochs.

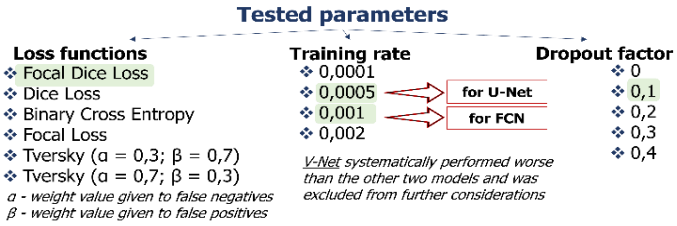

Fig. 5. Optimal variation of training parameters determined from exploratory training results

Table 4. Performance metrics for full-scale training

Model	GCE	VS	Precision	Recall
FCN	**0.4512**	**0.4674**	**0.5275**	0.1883
U-Net	0.6679	0.3335	0.2632	**0.2583**

In contrast to the shorter training exercises, the FCN model demonstrates a marked advantage over U-Net in full-scale training. The validation loss remains consistently lower for FCN compared to U-Net. Furthermore, GCE and precision scores show substantial improvements, VS is marginally higher, and the difference in recall is negligible.

The Dice coefficient for FCN is 0.277. While this value may be considered suboptimal in scenarios where sufficient training data is available, it is a satisfactory outcome given the dataset's limitations. Both the Dice coefficient and Dice Loss penalise discrepancies harshly, which explains why this relatively low value still results in reasonable segmentation performance, as illustrated in Figure 6. Several factors contributed to this, including anatomical variability, low contrast of MRI scans, and the inherent complexity of target structures. While this score is not clinically sufficient, it is an encouraging indicator of feasibility. Despite the challenges, the models could pick up on pertinent spatial patterns and structural features. The centre of Figure 6 presents the original MRI image, previously unseen by any of the trained models during training or validation. The left column displays the predicted segmentation masks generated by U-Net and FCN, respectively. The predictions are overlaid onto the original image in the right column, visually representing the segmentation results.

Due to the limited availability of original DICOM images, the authors prioritised ensuring an adequate number of training and validation sets. Consequently, only one image was retained for final prediction.

The segmentation produced by U-Net appears slightly inferior to that of FCN. Although the predicted volume is smoother and potentially captures a larger portion of the bladder, it erroneously includes part of an adjacent organ as an artefact. This significant misclassification suggests that U-Net may be impractical for further consideration.

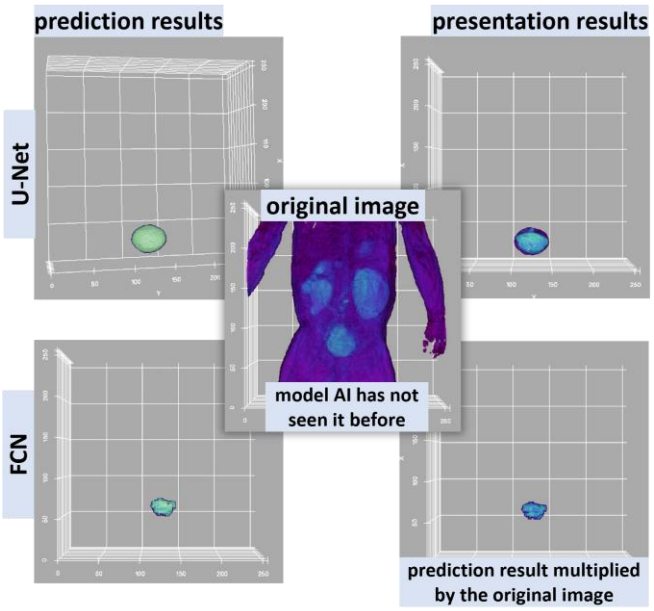

Fig. 6. Visualization of bladder detection

5. Conclusions

This paper proposes a new supervised learning approach for artificial intelligence-based segmentation of the lower urinary tract in the context of limited and imbalanced training data. Despite the constraints of a small dataset, our results confirm the feasibility of constructing effective models for segmentation by employing dedicated data augmentation, computational optimization, and stringent model evaluation.

To improve model performance under data scarcity conditions, the authors introduced a task-driven augmentation strategy that ensures maximum variability in the dataset with anatomical consistency. It involves using multiple augmentation strategies optimized for the intrinsic nature of MRI data so that it becomes generalizable without compromising medical validity.

A dynamic data pipeline was created that supports distributed GPU computation to fit the increased computational demand necessitated by augmentation. It allowed us to efficiently train models on large augmented datasets, overcoming hardware limitations without sacrificing scalability.

Comparison of three CNN models (FCN, U-Net, and V-Net) showed that FCN was superior to the others on full-scale training, with up to 94% precision and global consistency error less than 37%. This confirms that good model selection and strong augmentation and training strategies can lead to acceptable segmentation outcomes even in low-data scenarios.

Several overfitting reduction methods were used to ensure that the models learned generalizable features instead of artefacts specific to the dataset. The research highlights how curriculum learning and hybrid CNN architectures can enhance robustness and learning efficiency.

The proposed methods gave solutions to the key research questions:

- Task-driven data augmentation preserves anatomical consistency while enhancing dataset variability, resulting in better segmentation accuracy.
- Self-supervised pretraining, curriculum learning, and hybrid CNN architectures enhance model robustness and generalisation.

Although data augmentation is critical in low-resource settings, we are also aware of its limitations, such as increased computational needs and bias amplification potential. Augmentation cannot fully replace non-diversity in the data and relies heavily on trial-and-error to be effective. The model that has been established paves the way for adaptive, ever-evolving AI-based segmentation tools within the clinical setting. Instruments such as clustering and the Picking Tool offered enable model refinement by an expert, allowing for real-world diagnostic pipelines and future model retraining.

This work was supported in part by the National Centre for Research and Development (Warsaw, PL) under Grant POIR.04.01.04-00-0045/20, titled: "A system of non-invasive monitoring and diagnosis of functional disorders of the lower urinary tract by means of electric and ultrasound tomography.

The authors would like to thank Prof. M. Tkaczyk and Dr M. Krakós from Polish Mother's Memorial Hospital Research Institute for providing MRI images. Special thanks to students Andrzej Zieliński, Mateusz Juszczykowski, and Wojciech Kowalczyk for their contributions to the project's code.

Authors: *Radosław Wajman, Robert Banasiak, Maciej Sputowski, Volodymyr Mosorov; Institute of Applied Computer Science at Lodz University of Technology, Stefanowskiego 18, 90-537 Łódź, Poland.*

REFERENCES

[1] J. Tan, F. Qin, and J. Yuan, "Current applications of artificial intelligence combined with urine detection in disease diagnosis and treatment," *Transl. Androl. Urol.*, vol. 10, no. 4, pp. 1769779–1761779, Apr. 2021, doi: 10.21037/TAU-20-1405.

[2] B. Baran, E. Kozłowski, D. Majerek, T. Rymarczyk, M. Soleimani, and D. Wójcik, "Application of Machine Learning Algorithms to the Discretization Problem in Wearable Electrical Tomography Imaging for Bladder Tracking," *Sensors 2023, Vol. 23, Page 1553*, vol. 23, no. 3, p. 1553, Jan. 2023, doi: 10.3390/S23031553.

[3] G. Yuan *et al.*, "Machine learning-based bladder effusion estimation model construction on intravesical pressure data," *Biomed. Signal Process. Control*, vol. 86, p. 105207, Sep. 2023, doi: 10.1016/J.BSPC.2023.105207.

[4] A. Saad, U. U. Sheikh, and M. S. Moslim, "Developing Convolutional Neural Network for Recognition of Bone Fractures in X-ray Images," *Adv. Sci. Technol. Res. J.*, vol. 18, no. 4, pp. 228–237, 2024, doi: 10.12913/22998624/188656.

[5] A. Tantin, E. Bou Assi, E. van Asselt, S. Hached, and M. Sawan, "Predicting urinary bladder voiding by means of a linear discriminant analysis: Validation in rats," *Biomed. Signal Process. Control*, vol. 55, p. 101667, Jan. 2020, doi: 10.1016/J.BSPC.2019.101667.

[6] P. Powroznik, M. Skublewska-Paszkowska, K. Nowomiejska, A. Aristidou, A. Panayides, and R. Rejdak, "Deep convolutional generative adversarial networks in retinitis pigmentosa disease images augmentation and detection," *Adv. Sci. Technol. Res. J.*, vol. 19, no. 2, pp. 321–340, 2025, doi: 10.12913/22998624/196179.

[7] J. Dominic *et al.*, "Improving Data-Efficiency and Robustness of Medical Imaging Segmentation Using Inpainting-Based Self-Supervised Learning," *Bioengineering*, vol. 10, no. 2, Feb. 2023, doi: 10.3390/BIOENGINEERING10020207.

[8] H. Wang and X. Li, "DHC: Dual-debiased Heterogeneous Co-training Framework for Class-imbalanced Semi-supervised Medical Image Segmentation," pp. 582–591, doi: 10.48550/ARXIV.2307.11960.

[9] Q. Liu, X. Gu, P. Henderson, and F. Deligianni, "Multi-Scale Cross Contrastive Learning for Semi-Supervised Medical Image Segmentation," pp. 868–869, doi: 10.48550/ARXIV.2306.14293.

[10] K. Chaitanya *et al.*, “Semi-supervised Task-driven Data Augmentation for Medical Image Segmentation,” *Med. Image Anal.*, vol. 68, Feb. 2021, doi: 10.1016/J.MEDIA.2020.101934.
[11] X. Hu, D. Zeng, X. Xu, and Y. Shi, “Semi-supervised Contrastive Learning for Label-efficient Medical Image Segmentation,” *ArXiv*, vol. abs/2109.07407, pp. 481–490, 2021, doi: 10.1007/978-3-030-87196-3_45.
[12] K. Wang *et al.*, “An Efficient Semi-Supervised Framework with Multi-Task and Curriculum Learning for Medical Image Segmentation,” *Int. J. Neural Syst.*, vol. 32, no. 9, Sep. 2022, doi: 10.1142/S0129065722500435.
[13] X. Zhao and W. Wang, “Semi-Supervised Medical Image Segmentation Based on Deep Consistent Collaborative Learning,” *J. Imaging*, vol. 10, no. 5, May 2024, doi: 10.3390/JIMAGING10050118.
[14] G. Rybak, K. Strzecha, Ł. Sturgulwski, M. Krakós, and D. Sankowski, “Aspekty architektoniczne tworzenia cyfrowej platformy gromadzenia danych medycznych,” *Przegląd Elektrotechniczny*, vol. 1, no. 12, pp. 147–150, 2021, doi: 10.15199/48.2021.12.29.
[15] M. Dziadosz, M. Mazurak, D. Wójcik, and M. Oleszek, “ Integracja algorytmów uczenia maszynowego z czujnikami noszonymi na ciele opartymi na elektrycznej tomografii impedancyjnej,” *Przegląd Elektrotechniczny*, vol. 3, p. 91, 2025, doi: 10.15199/48.2025.03.22.
[16] M. Gołąbek, T. Rymarczyk, P. Brożek, D. Stefańczak, and D. Wójcik, “Przenośny hybrydowy tomograf ultradźwiękowo-impedancyjny do monitorowania dolnych dróg moczowych w aspekcie kompatybilności elektromagnetycznej,” *Przegląd Elektrotechniczny*, vol. 3, p. 95, 2025, doi: 10.15199/48.2025.03.23.
[17] M. Parsian, “Data, Algorithms, and Code: Recipes For Scaling Up With Hadoop And Spark,” *Routledge Companion to Digit. Journal. Stud.*, p. 778, 2015, Accessed: Nov. 13, 2023. [Online]. Available: https://www.oreilly.com/library/view/data-algorithms/9781491906170/
[18] N. S. Keskar, J. Nocedal, P. T. P. Tang, D. Mudigere, and M. Smelyanskiy, “On Large-Batch Training for Deep Learning: Generalization Gap and Sharp Minima,” *5th Int. Conf. Learn. Represent. ICLR 2017 - Conf. Track Proc.*, Sep. 2016, Accessed: Aug. 30, 2023. [Online]. Available: https://arxiv.org/abs/1609.04836v2
[19] V. Badrinarayanan, A. Kendall, and R. Cipolla, “SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 12, pp. 2481–2495, Dec. 2017, doi: 10.1109/TPAMI.2016.2644615.
[20] Y. Bengio, “Practical recommendations for gradient-based training of deep architectures,” in *Montavon, G., Orr, G.B., Müller, KR. (eds) Neural Networks: Tricks of the Trade. Lecture Notes in Computer Science*, vol. 7700, Springer Verlag, 2012, pp. 437–478. doi: 10.1007/978-3-642-35289-8_26.
[21] K. Hasanov and A. Lastovetsky, “Hierarchical redesign of classic MPI reduction algorithms,” *J. Supercomput.*, vol. 73, no. 2, pp. 713–725, Feb. 2017, doi: 10.1007/S11227-016-1779-7.
[22] A. M. Fred Agarap, “Deep Learning using Rectified Linear Units (ReLU),” *Neural Evol. Comput.*, Mar. 2018, Accessed: Aug. 30, 2023. [Online]. Available: https://arxiv.org/abs/1803.08375v2
[23] G. Kłosowski, T. Rymarczyk, K. Niderla, M. Rzemieniak, A. Dmowski, and M. Maj, “Comparison of machine learning methods for image reconstruction using the LSTM classifier in industrial electrical tomography,” *Energies*, vol. 14, no. 21, Nov. 2021, doi: 10.3390/EN14217269.
[24] S. Minaee, Y. Boykov, F. Porikli, A. Plaza, N. Kehtarnavaz, and D. Terzopoulos, “Image Segmentation Using Deep Learning: A Survey,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 7, pp. 3523–3542, Jul. 2022, doi: 10.1109/TPAMI.2021.3059968.
[25] G. Chen, Y. Hong, K. M. Huynh, and P. T. Yap, “Deep learning prediction of diffusion MRI data with microstructure-sensitive loss functions,” *Med. Image Anal.*, vol. 85, p. 102742, Apr. 2023, doi: 10.1016/J.MEDIA.2023.102742.
[26] G. E. Hinton, P. Dayan, B. J. Frey, and R. M. Neal, “The ‘Wake-Sleep’ Algorithm for Unsupervised Neural Networks,” *Science (80-.).*, vol. 268, no. 5214, pp. 1158–1161, May 1995, doi: 10.1126/SCIENCE.7761831.
[27] T. Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, “Focal Loss for Dense Object Detection,” *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 2017-October, pp. 2999–3007, Dec. 2017, doi: 10.1109/ICCV.2017.324.
[28] C. H. Sudre, W. Li, T. Vercauteren, S. Ourselin, and M. Jorge Cardoso, “Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 10553 LNCS, pp. 240–248, 2017, doi: 10.1007/978-3-319-67558-9_28/FIGURES/4.
[29] S. S. M. Salehi, D. Erdogmus, and A. Gholipour, “Tversky loss function for image segmentation using 3D fully convolutional deep networks,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 10541 LNCS, pp. 379–387, 2017, doi: 10.1007/978-3-319-67389-9_44/FIGURES/4.
[30] K. Siddiqui and T. E. Doyle, “Trust Metrics for Medical Deep Learning Using Explainable-AI Ensemble for Time Series Classification,” *Can. Conf. Electr. Comput. Eng.*, vol. 2022-September, pp. 370–377, 2022, doi: 10.1109/CCECE49351.2022.9918458.
[31] R. Cárdenes, R. de Luis-García, and M. Bach-Cuadra, “A multidimensional segmentation evaluation for medical image data,” *Comput. Methods Programs Biomed.*, vol. 96, no. 2, pp. 108–124, Nov. 2009, doi: 10.1016/J.CMPB.2009.04.009.
[32] A. A. Taha and A. Hanbury, “Metrics for evaluating 3D medical image segmentation: Analysis, selection, and tool,” *BMC Med. Imaging*, vol. 15, no. 1, pp. 1–28, Aug. 2015, doi: 10.1186/S12880-015-0068-X/TABLES/5.
[33] S. Park and N. Kwak, “Analysis on the dropout effect in convolutional neural networks,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 10112 LNCS, pp. 189–204, 2017, doi: 10.1007/978-3-319-54184-6_12/TABLES/8.
[34] S. Hameetha Begum and S. N. Nisha Rani, “Model Evaluation of Various Supervised Machine Learning Algorithm for Heart Disease Prediction,” *Proc. - 2021 Int. Conf. Softw. Eng. Comput. Syst. 4th Int. Conf. Comput. Sci. Inf. Manag. ICSECS-ICOCSIM 2021*, pp. 119–123, Aug. 2021, doi: 10.1109/ICSECS52883.2021.00029.