Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2023) 31: 1021 – 1038

doi:10.55730/1300-0632.4032

Infrared imaging segmentation employing an explainable deep neural network

Xinfei LIAO[1], Dan WANG[1,*], Zairan LI[1], Nilanjan DEY[2], R. Simon SHERRATT[3], Fuqian SHI[4]

[1]Department of Computer Science, Wenzhou Polytechnic, Wenzhou, China
[2]Techno International New Town, Kolkata, India
[3]Department of Biomedical Sciences and Biomedical Engineering, School of Biological Sciences, University of Reading, Reading, UK
[4]Department of System Biology, Rutgers Cancer Institute of New Jersey, New Brunswick, NJ, USA

Received: 04.07.2023 • **Accepted/Published Online:** 30.09.2023 • **Final Version:** 27.10.2023

Abstract: Explainable AI (XAI) improved by a deep neural network (DNN) of a residual neural network (ResNet) and long short-term memory networks (LSTMs), termed XAIRL, is proposed for segmenting foot infrared imaging datasets. First, an infrared sensor imaging dataset is acquired by a foot infrared sensor imaging device and preprocessed. The infrared sensor image features are then defined and extracted with XAIRL being applied to segment the dataset. This paper compares and discusses our results with XAIRL. Evaluation indices are applied to perform various measurements for foot infrared image segmentation including accuracy, precision, recall, F1 score, intersection over union (IoU), Dice similarity coefficient, mean intersection of union, boundary displacement error (BDE), Hausdorff distance, and receiver operating characteristic (ROC). Compared to results from the literature, XAIRL shows the highest overall performance, achieving accuracy of 0.93, precision of 0.91, recall of 0.95, and F1 score of 0.93. XAIRL also displays the highest IoU, Dice similarity coefficient, and ROC curve and the lowest BDE and Hausdorff distance. Although U-Net performs well for most metrics, Mask R-CNN shows slightly worse performance but still outperforms the random forest and support vector machine algorithms. By building a high-quality foot infrared imaging dataset, machine learning-based algorithms can accurately analyze foot temperature and pressure distribution. These models can then be used to customize shoes for individual wearers, improving their comfort and reducing the risk of foot injuries, particularly for those with high blood pressure.

Key words: Convolutional neural networks, deep learning, explainable AI, image recognition, long short-term memory networks

1. Introduction

An infrared sensor is a type of sensor that detects infrared radiation. Any object in nature, as long as its stability is above absolute zero, will radiate infrared energy. Therefore, an infrared sensor is called a very practical sensor [1, 2]. By observing the optical system and the infrared detector, an optical mechanical scanning mechanism scans the infrared thermal image of the measured object and focuses on the unit or spectral detector. The detector converts the infrared radiation energy into electrical signals, which are amplified, converted, or standard video signals displayed through a television screen or a monitor [3, 4].

*Correspondence: wangdan_2023@126.com

In the preprocessing of infrared images, image segmentation is a very important step in early processing. Researchers have achieved a large number of results in the field of image segmentation, which stem from the utilization of different features in images, such as the similarity of features within a region, connectivity between pixels, and discontinuity between targets and backgrounds. However, to date, there is no segmentation algorithm that can be used for all image segmentation, which also promotes continuous research on image segmentation by researchers. Convolutional neural networks (CNNs) have achieved state-of-the-art performance in automatic medical image segmentation. However, the poor interpretability of the existing CNNs limits their application in clinical decision-making. It becomes crucial to explain the results of the algorithm output to the user [5, 6]. The interpretability of artificial intelligence (AI) refers to the ability of people to understand the choices made by AI models in their decision-making processes, including the reasons for making decisions, the methods, and the contents of the decisions.

The biggest problem with current AI systems based on big data and deep learning is their lack of explainability and understanding, leading to reduced performance when faced with dynamic environments and incomplete or false information and preventing human–machine interaction and collaboration. This black-box problem is crucial because users will not trust AI if they do not understand how AI makes its decisions. Explainable AI (XAI) is a crucial area of study and may become the core of future machine learning, but as models become more complex, it becomes increasingly difficult to determine simple, explainable rules. New machine learning systems will be able to explain their basic principles, represent their advantages and disadvantages, and convey an understanding of how they will perform in the future. This goal will be achieved by developing new or improved machines with more explainable models, which will then be combined with cutting-edge human–machine interface technology to provide interpretable explanations to end users. Our strategy is to adopt various techniques to generate a range of methods and provide a range of design options for future developers that can offer a balance between performance and explainability. Despite the good performance of deep CNNs in medical image processing, the lack of explainability of the black box impedes the development of intelligent medical diagnosis. Despite deep learning having made significant progress in medical diagnosis, its lack of interpretability has become a significant hindrance to its broad adoption in the medical domain [7].

There are several reasons why XAI and deep neural networks (DNNs) work well for infrared sensor imaging dataset segmentation. First, they have the ability to learn complex features; DNNs are capable of learning and representing complex and advanced attributes. In the case of infrared sensor imaging, this enables the network to extract meaningful features from the image that aid in accurate segmentation. Second, by using large amounts of data, a DNN requires a significant training dataset, and the availability of large datasets for infrared sensor imaging enables the network to learn robust and accurate representations of the data. Third, with their nonlinear mapping, DNNs can learn intricate relationships between inputs and outputs that can be nonlinear in nature, which is essential for the accurate segmentation of infrared sensor imaging data that have complex and nonlinear relationships between input features. Finally, the XAI techniques for interoperability may help explain how the DNN arrived at its segmentation results, making it easier for experts to validate and improve the segmentation performance [8]. Conventional machine learning models, which rely on statistical analysis, are generally easier to interpret than their deep learning counterparts. For example, linear models enable interpretation of the significance and implications of parameters in neural networks by analyzing their weights and ranges of fluctuation. Decision tree models also offer user-friendly decision-making criteria by presenting a sequence of decision points. Variable selection criteria based on information theory assist in identifying the variables that have a more significant impact on the models. Rule-based expert systems utilize domain-specific knowledge bases and strategy libraries to interpret contextual logic relationships [9].

As the complexity of deep learning models continues to grow, unraveling the decision-making processes within multilayer neural networks that consist of multiple nonlinear functions and comprehending their neural pathways has become progressively intricate. Consequently, the pursuit of interpretability in the domain of AI can be categorized into two principal domains: model-driven and user-driven. The realm of AI is intricately constructed to optimize behaviors predicated upon mathematical target systems. For instance, a directive may be set to "maximize the accuracy of positive movie reviews within the test dataset." While AI has the capability to deduce overarching patterns from the test set, such as discerning that "reviews containing the term 'horror' tend to be associated with negativity," it might also internalize less desirable associations, as exemplified by the inference that "reviews mentioning 'Daniel Day Lewis' are generally indicative of positivity." The latter type of learned associations could potentially lack the capacity to extend to future real-world data beyond the confines of the test set or might be viewed as instances of "unfairness" or "cheating" in nature. In such cases, these acquired rules could be regarded as undesirable outcomes. XAI emerges as a transformative approach that empowers human auditors to critically examine these learned rules. The overarching aim of this approach is to assess the viability of the system's ability to generalize its acquired insights to forthcoming real-world data that exist outside the realm of the test set. By employing XAI, humans gain the ability to scrutinize the system's underlying decision-making mechanisms, evaluating the likelihood of its applicability beyond controlled test conditions [10].

An explainable deep neural network (xDNN) is a model that can provide insight into its decision-making processes through various methods such as attribution maps, saliency maps, and layerwise relevance propagation [11, 12]. These methods aim to make the model's decisions more transparent and understandable, allowing for improved trust and interpretability of the model's predictions. There are several different types of xDNN models, including:

a) Gradient-based methods: These methods use the gradient of the model's output with respect to its input to generate attribution maps that highlight the most important areas of the image for prediction [13].

b) Input perturbation: Taking the LIME model as an example, input perturbation can explain the output of the input image by generating random perturbations of input x and training an interpretable model (usually a linear model).

c) Relevance score propagation layer by layer: This a technology that integrates interpretability into highly complex deep learning neural networks. The prediction results are backpropagated in the neural network through a specially designed backpropagation rule [14].

The best method may depend on the specific application and the type of data being analyzed. This research aims to create a feasible framework for XAI applied in foot infrared imaging segmenting.

2. Modeling

A simple and easily interpretable regression or decision tree model can no longer fully meet technical and business needs. More and more people are using ensemble methods and DNNs to achieve better predictions and higher accuracy [15, 16]. However, those complex models are difficult to explain, debug, and understand. Researchers and machine learning practitioners have designed many model interpretation techniques. In this study, we provide a high-level overview of eight popular model interpretation techniques and tools, including Shapley additive explanations (SHAP), LIME (explaining individual predictions of machine learning models), explainable boosting machine (EBM) (interpretable augmentation machine), saliency maps, testing with concept activation vectors (TCAV) (a new linear interpretability method), distillation, counterfactuals, and lnterpretML

(which provides developers with multiple ways to experiment with AI models and systems and further explain the models) [17, 18].

2.1. Basic models

LIME is the abbreviation for "local interpretable model agnostic explanation." "Local" means it can be used to explain individual predictions of a machine learning model. It is also very simple to use, requiring only two steps: (1) import the module and (2) fit the interpreter using training values, features, and targets [19]. The key steps of the LIME algorithm are as follows:

- Example instance: A group of disturbed instances around the instance of interest is generated by randomly disturbing the characteristics of the original instance.
- Get predictions: The perturbed instances are then fed through the original machine learning model, and the predicted output for each instance is recorded.
- Fit a local model: LIME fits a simple, interpretable model such as a linear regression model to the perturbed instances and their corresponding predictions. This model approximates the complex model in the local neighborhood of the instance of interest.
- Compute feature importance: The coefficients of the local model are used as weights to determine the importance of each feature in the prediction.
- Generate explanation: Finally, the top features and their weights are presented as an explanation for the prediction of the complex model on the instance of interest.

LIME has been widely used in many fields [19–22]. Algorithm 1 can help users understand how the model predicts. In Algorithm 1, the LIME prediction function is defined by taking instance x, the machine learning model, the number of samples to be used, and the optional parameters of features as inputs. The working principle of the algorithm is to sample instances from the neighborhood of x, predict the model output of each sample, and then fit the local linear model to the prediction result.

Algorithm 1: A sample table including some styles.

REQUIRED: img: x, model, number of samples, number of features
OUTPUT: explanation: features and weights

```
# Sample instances from the neighborhood of x
samples ← generate_samples(x, num_samples)
# Predictions of the model for the samples
predictions ← model.predict(samples)
# Define the local linear model and fit it to the predictions
local_model ← LinearRegression()
features ← select_features(x, samples, predictions, num_features)
local_model.fit(features, predictions)
# Compute the local weights for each feature
weights ← compute_weights(x, local_model, samples, features)
# Return the explanation (features with their weights)
explanation ← [(f, w) for f, w in zip (features, weights)]
return explanation
```

Noted that the get_image_and_mask function was used to obtain the image and mask for the top five most important features in the prediction, and then we used the function of mark_boundaries to overlay the mask onto the image. This creates a visual representation of the important features in the image for the prediction of the model as shown in Algorithm 2.

Algorithm 2: Foot infrared sensor images feature extraction.

REQUIRED: img: foot sensor imaging dataset
OUTPUT: segmented_image

```
#Load the foot infrared image
img ← imread('foot_infrared_image.jpg');
#Preprocess the image (e.g., normalize, resize, etc.)
#Create a LIME explainer object
explainer ← lime();
#Set the explainer's parameters
explainer ← explainer.setModel('deeplabv3plus');
explainer ← explainer.setExplainer('lime');
explainer ← explainer.setSegmenter('felzenszwalb');
#Explain the model's prediction for the image
explanation ← explainer.explain(img);
# Extract the important features from the explanation
features ← explanation.getFeatures();
# Use the extracted features for segmentation
# (e.g., using a traditional machine learning algorithm)
#Display the segmented image
imshow(segmented_image);
```

2.2. XAI modeling improved by ResNet and LSTM

ResNet (Residual Network) was developed to address the issue of polishing gradients [23]. The ResNet architecture uses residual connections, where the input to a layer is added to its output, allowing for easier optimization and improved performance. The ResNet architecture has been widely adopted in various imaging tasks, such as image classification [24], object detection [25], and semantic segmentation [26]. ResNet can realize the flow of information within the network. Each skip-connected calculation unit is called a residual block. In a ResNet with residual blocks, the forward output of the lth residual block and the gradient of loss L, that is, its input y_i, are defined as [27]:

$$y_L = y_l + \sum_{n=l}^{L-l} F_n\left(y_n\right) \tag{1}$$

$$\frac{\partial \ell}{\partial y_l} = \frac{\partial \ell}{\partial y_L}\left(1 + \frac{\partial}{\partial y_l}\sum_{n=l}^{L-l} F_n\left(y_n\right)\right) \tag{2}$$

Among these, F_n consists of continuous batch normalization, a rectified linear unit (ReLU), and a convolution module. One jump connection in the residual block provides two information flow paths, so as the network goes deeper, the total number of paths in the network grows exponentially. This exponential integration improves network performance. The classification module connected to the convolutional layer in ResNet includes a global

average pooling layer and a fully connected layer as follows:

$$p^{class} = \sum_{k} w_k^{class} \sum_{i,y} y_L^{(k)}(i,j) \tag{3}$$

Here, p^{class} is the probability output for the classification of $class(i,j)$ in which (i,j) is the spatial coordinate, and w^{class} is the $class$th column of the weight matrix of the fully connected layer applied to p^{class}. When inserting Eq. (1) into Eq. (3), we have the following:

$$p^{class} = \sum_{i,j} w^{class} y_L = \sum_{i,j} w^{class} \left(y_1 + \sum_{n=1}^{L-1} F_n \right) \tag{4}$$

Continuing, we decouple the ensemble outputs and apply classifiers to them separately by:

$$p^{class} = \sum_{i,j} \left(w^{class} y_1 + \sum_{n=1}^{L-1} w_{n+1}^{class} \cdot F_n \right) \tag{5}$$

Using Eq. (4) and Eq. (5) to assign separate weights $(w_j)^{class})$ and $(w_L)^{class}$ to each ensemble output enables the classification module to independently decide the importance of information from different residual blocks. We then restructure the ResNet architecture to realize the above ideas to adopt a new way to jump and connect residual blocks, defined as follows:

$$y_{l+1} = F_l(y_l) \otimes y_l \tag{6}$$

Here, $\otimes$ is the connection operation. This jump connection scheme is defined as a collection connection (ensemble connection). It allows the output of the residual block to flow directly in parallel through the parallel feature map to the classification layer. As shown in Figure 1, this design also ensures the unimpeded flow of information and overcomes the vanishing gradient effect.

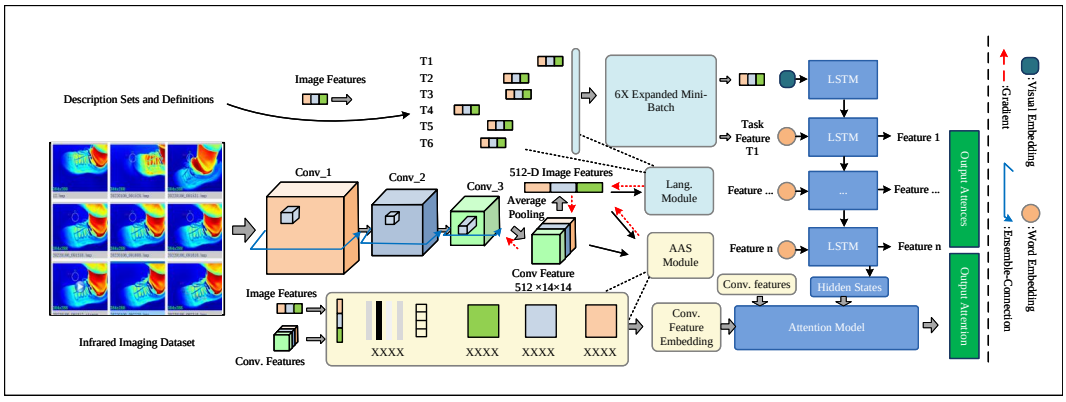

Figure 1. LSTM and ResNet-based feature input training process for the infrared imaging dataset.

The ResNet and LSTM parameters were first initialized and then the ResNet blocks were defined, where a series of convolutional operations with batch normalization and activation functions were performed, and the residual connection back to the output was added. After the ResNet blocks, the output passed through the LSTM layer was reshaped, where the temporal dependencies in the data were left. Subsequently, a dropout

layer to prevent overfitting was added, and the output was passed through another LSTM layer. Finally, a dense layer with a softmax activation function as the output layer to classify the input into one of the num_classes categories was used [28, 29]. Algorithm 3 shows the necessary processing for the given dataset.

Algorithm 3: ResNet and LSTM-improved XAI.

REQUIRED: matrix, labels, images
OUTPUT: explanation

```
# load dataset
data ← load('my_dataset.mat');
images ← data.images;
labels ← data.labels;
# Split data into training and validation sets
trainImages, trainLabels, valImages, valLabels ← splitData(images, labels);
#Train ResNet on the training set
resnetModel ← trainResNet(trainImages, trainLabels);
# Train LSTM on the training set
lstmModel ← trainLSTM(trainText, trainLabels);
# Use LIME to explain ResNet predictions on validation set
FOR i FROM 1 to length(valImages)
 explainer ← lime(resnetModel, valImages(i));
 explanation ← explain(explainer, valImages(i));
 # Display explanation for the current image
 display(explanation);
ENDFOR
# Use LIME to explain LSTM predictions on validation set
FOR i FROM 1 to length(valText)
 explainer ← lime(lstmModel, valText(i));
 explanation ← explain(explainer, valText(i));
 # Display explanation for the current text
 display(explanation);
ENDFOR
```

2.3. Preprocessing using convolutional encoder network learning in XAIRL

Convolutional encoder network learning (CENL) is a deep learning approach that uses convolutional neural networks (ConvNets) for feature extraction and encoding. It involves training a ConvNet to map an input image to a compact, low-dimensional representation (encoding) while preserving the important information in the image. The encoder–decoder architecture was used for infrared image preprocessing; the model is a type of neural network architecture that consists of two main components: an encoder and a decoder. This architecture is commonly used in fully convolutional layers for various computer vision tasks [30], such as image segmentation [31], image synthesis [32], and image-to-image translation [33]. Figure 2 shows the infrared imaging dataset preprocessing using an encoder–decoder with convolutional and full layers. CENL allows for effective feature extraction and can improve the performance of these tasks by using a compact representation that is learned from the data. It has been used in various imaging applications and researchers continue to explore new and improved variants of the approach [34, 35]. Throughout the training process, the decoder network is taught to construct an estimated version of input X by extracting vector z from the discovered latent manifold with d dimensions. In parallel, a predictive network (referred to as the prediction network in this article) is linked to

the mean vector z and comprises a multilayer perceptron (MLP) that learns to differentiate between volunteers and other individuals. End-to-end training is performed using the following loss function:

$$\ell = \ell_{rec} + \alpha \ell_{KL} + \beta \ell_{MLP} \tag{7}$$

Here, l_{rec} represents the reconstruction loss, which can be calculated by the Sorensen Dice loss between input X and the reconstruction. l_{KL} is the Kullback–Leibler divergence loss [36], which aims to make $N(\mu_i, \sigma_i)$ as close as possible to its previous distribution $N(0,1)$. l_{MLP} is the cross-entropy loss for the MLP classification task. The latent space dimension is $d = 64$. During the testing phase, each input segment is reconstructed by passing the predicted μ to z without sampling from the latent space, and, finally, the classification task is performed during the training phase. Using the weights learned by the MLP, the partial derivatives of class label $C(y_c)$ are computed by backpropagating the gradient from class label C to μ_i using the chain rule. Given a randomly chosen shape of healthy tissue, the derived gradient can be used to move the subject's latent representation along the direction of latently encoded variability, using an iterative algorithm to maximize the probability of classifying that variability into class C. Starting from the average latent representation of the healthy shape, μ_i is iteratively updated at each step t:

$$\mu_{i,t} = \mu_{i,t-1} + \lambda \frac{\partial y_1}{\partial \mu_{i,t-1}} \tag{8}$$

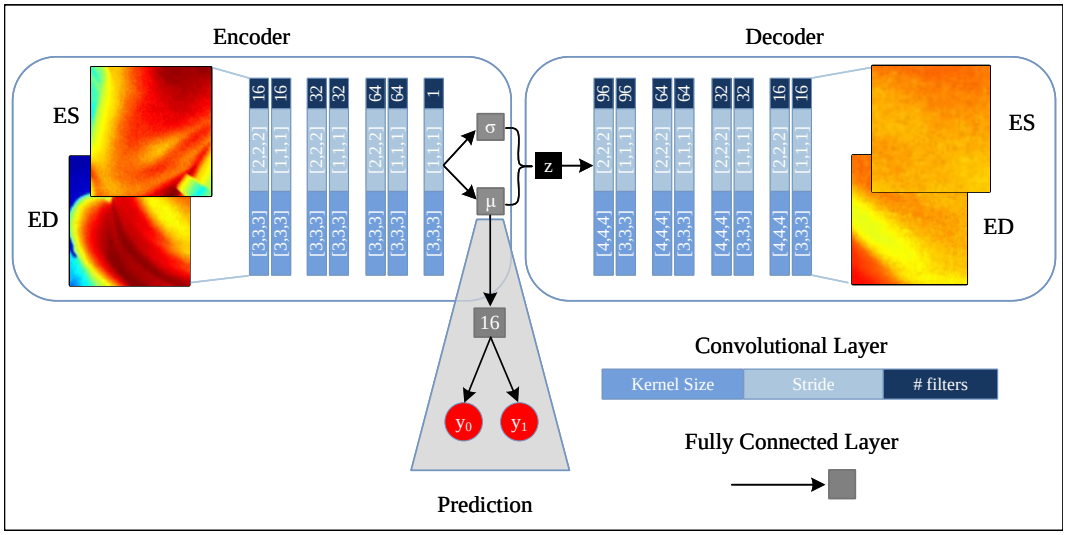

Figure 2. Infrared imaging dataset preprocessing using encoder–decoder with convolutional and fully connected layers.

2.4. Training with feature embedding matrix and optimization of XAIR

All description models in Section 2.3 shared their LSTM. In this way, each image characterization model becomes a function that generates a complete report, and this function is defined as K. In the training phase, given a mini-batch containing B pairs of images and reports, each sample is internally replicated after sending the

mini-batch to the image model, resulting in a $K \times B$-sized mini-batch as input to the LSTM. The input and output of the LSTM are defined as follows:

$$y_0^e = w_f g\left(i\right) \tag{9}$$

$$y_0^e = w_f g\left(i\right) \tag{10}$$

Here, w_f represents the learned image feature embedding matrix and $s(e)$ represents the one-hot representation of the Eth image feature type. We use y_1^E to notify the LSTM of the start of the target task. In the backpropagation stage, all replicated gradients $g(i)$ are fused. The whole model contains three sets of parameters: parameters θ_D of image model D, parameters θ_L of model L, and parameters θ_M of module M. The complete optimization problem for the net is as follows:

$$\max_{\theta_L,\theta_D,\theta_M} \ell_M\left(l_c, M\left(D\left(I:\theta_D\right);\theta_M\right)\right) + \ell_L\left(l_s, L\left(D\left(I;\theta_D\right);\theta_L\right)\right) \tag{11}$$

Here, $\{I, l_c, l_s\}$ represents the training triplet. θ_M and θ_L can be solved directly using the gradient descent algorithm. However, updating θ_D depends on the gradients of both modules at the same time. This paper proposes a backpropagation mechanism that allows the composite gradients of the two modules to adapt to each other. Gradients are computed based on a hybrid of recurrent generative networks and the MLP, and θ_D is updated as follows:

$$\theta_D = \theta_D - \lambda\left(\left(1-\beta\right)\frac{\partial \ell_M}{\partial \theta_D} + \beta\eta\frac{\partial \ell_L}{\partial \theta_D}\right) \tag{12}$$

3. Results

3.1. Data acquisition and experiments

One hundred healthy students (50 men and 50 women) without flat feet were tested statically and dynamically through a spacious and straight passageway that was 6 m long with an auxiliary 2 m as shown in Figure 3. The acquisition of the infrared imaging dataset was conducted with meticulous care and precision, utilizing the sophisticated IRTools v2.62 software (http://irtools.com). This endeavor was guided by a meticulous experimental protocol designed to capture high-quality infrared images while ensuring consistency and reliability across the dataset. The color shows the temperature and the curvature (middle) is the min/max/average of the temperature. The left-side subfigure shows the speed. The right-side subfigure shows infrared images at different times including the basic information of the images collected in the experiment.

Due to the dynamic differences in individual walking postures, there was considerable interference with the stability of the plantar pressure during gait testing. Therefore, before the experiment, participants were introduced to the experimental equipment and precautions. Additionally, sample illustrations of the basic testing methods and postures were posted on the horizontal sidewalls during the experiment. All participants received simple gait training, with instructions to keep their eyes level and forward and maintain their walking posture. These measures ensured the objectivity and validity of the experimental data. We extracted 60 pictures from

the 100 participants and 70% of them were adopted for neural network training as described in Algorithm 1. Fifteen percent of them were used for testing and 15% of them were used for validation.

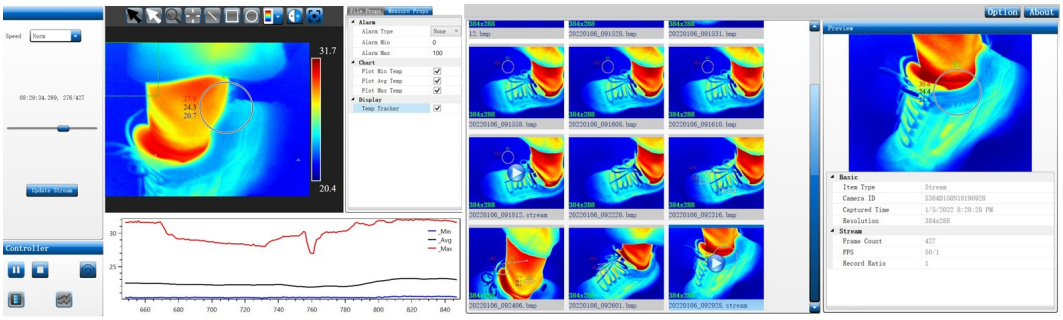

Figure 3. Infrared imaging dataset acquisition with IRTools v2.62.

A suggested framework for infrared image classification in a neural network using a feature map is shown in Figure 4. The processing includes resizing and normalizing the infrared images to the desired input size of the neural network, converting the images to gray-scale to reduce the input dimensionality, extracting features from the input images, adding batch normalization and activation layers after each convolutional layer to introduce nonlinearity and improve the training speed, visualizing the feature maps at each convolutional layer to understand how the CNN is learning the features, training the CNN using a suitable optimizer and loss function for image classification, and, finally, evaluating the trained model on the validation and test datasets to measure its classification accuracy. Figure 5 shows the infrared image component extraction using feature layers and traits based on row-centered sample compression technology. This is a technique used in imaging to extract relevant features from infrared images for classification and detection tasks. The technique involves the use of a CNN to learn feature maps from the input infrared images, followed by a row-centered sample compression to reduce the dimensionality of the feature maps. The extracted features can then be used as input to a classifier or detector, which is trained on a labeled dataset to recognize and classify different objects or patterns in the infrared images [37].

Figure 6 shows the segmentation results for the proposed XAI model. The segmentation results were evaluated using accuracy, precision, recall rate, and F1 score [38–40]. They can also be visualized using color-coded masks, highlighting different foot components and their boundaries, as described below.

4. Discussion

Using XAI combined with ResNet and LSTM to analyze the similarity of the infrared components of the foot, the segmentation results involve recognizing and separating the different components of the foot according to infrared image patterns. This can be achieved by combining the use of deep neural network models such as ResNet and LSTM with XAI technology. ResNet is a convolutional neural network architecture that is very suitable for image recognition tasks. It can learn the depth representation of image features by using skip

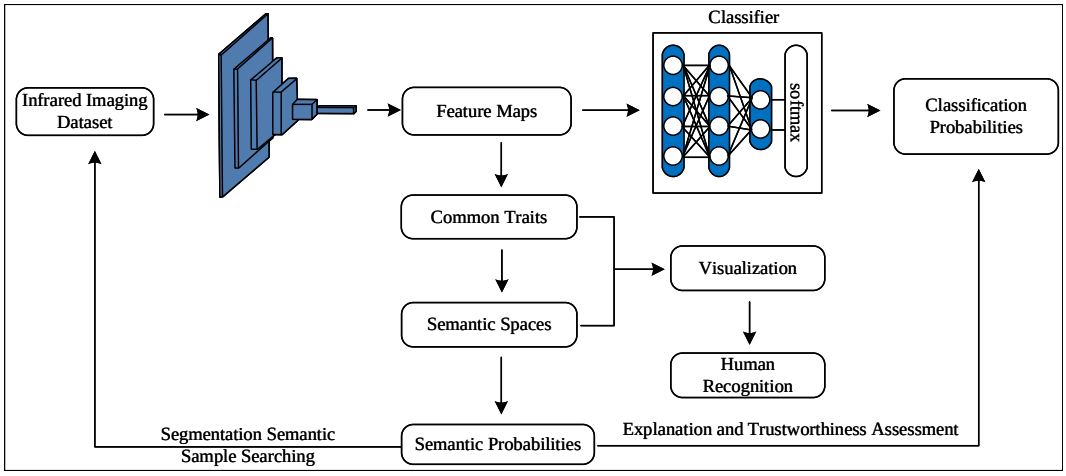

Figure 4. Framework for infrared image classification in neural network using feature map.

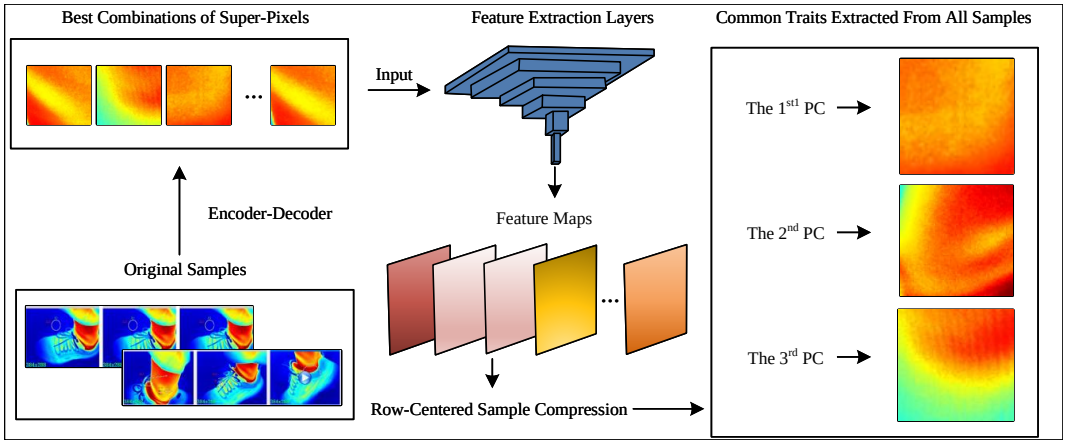

Figure 5. Foot infrared images feature extraction.

connections so that it can bypass some layers in the network. LSTM can capture the time dependency in data by controlling the flow of information using storage units and gates. To segment different foot components, ResNet can be used to extract feature images from the input infrared image. These feature maps highlight the different patterns and textures related to component segmentation in the image. We can then use LSTM to analyze the time series of these characteristic graphs and learn the dependencies between them. XAI technology can be used to interpret segmentation results and help identify specific features and patterns used by the model for prediction [41]. For example, saliency maps can be generated to highlight the regions of the input images that are most important for the segmentation task. Attention mechanisms can also be used to visualize the weights and activations of the models, showing how they are influenced by different parts of the input data. The segmentation results can be evaluated using various performance metrics, such as accuracy, precision, recall, and F1 score. They can also be visualized using color-coded masks that highlight the different foot components and their boundaries.

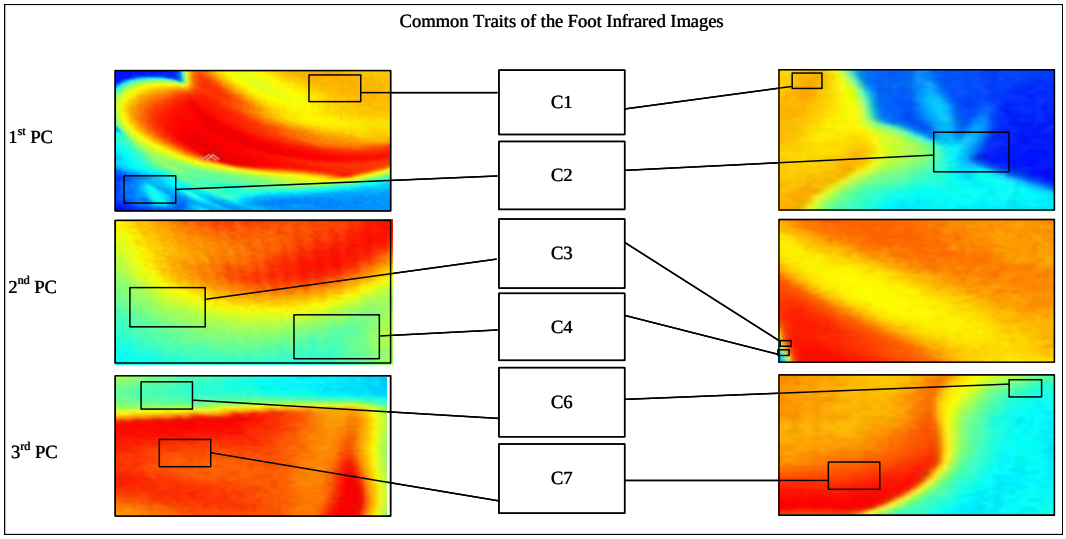

Figure 6. Segmentation results for the proposed XAI model.

In this research, we calculated several indices to compare the proposed method to others. Accuracy: The proportion of correctly classified pixels or regions in the segmentation result. Precision: The proportion of true positives (correctly segmented pixels or regions) among all positively classified pixels or regions. Recall (also known as sensitivity): The proportion of true positives among all actual positive pixels or regions in the ground truth. F1 score: A harmonic mean of precision and recall, which provides a balance between those two measures. Intersection over union (IoU): The ratio of the intersection between the segmentation result and ground truth to their union, which measures the overlap between the two. Dice similarity coefficient (DSC): Another measure of overlap between the segmentation result and ground truth, which is defined as twice the intersection divided by the sum of the sizes of the two regions. Mean intersection over union (mIoU): The average IoU over all classes or regions in the segmentation result. Boundary displacement error (BDE): The average distance between the boundary of the segmentation result and the ground truth boundary, which measures the accuracy of the boundary delineation. Hausdorff distance: The maximum distance between any two points in the segmentation result and ground truth, which measures the overall difference between the two regions. Receiver operating characteristic (ROC) curve: A plot of the true positive rate against the false positive rate, which measures the trade-off between sensitivity and specificity in binary segmentation tasks. Table 1 shows the foot infrared image segmentation using XAI with ResNet and LSTM compared to other methods, including various image segmentation metrics. The metrics used for evaluation include accuracy, precision, recall, F1 score, IoU, DSC, mIoU, BDE, Hausdorff distance, and ROC. The results show that XAIRL achieves the highest overall performance, with accuracy of 0.93, precision of 0.91, recall of 0.95, and F1 score of 0.93. XAIRL also achieves the highest IoU, DSC, and ROC curve and the lowest BDE and Hausdorff distance. U-Net performs well on most metrics, while Mask R-CNN performs slightly worse but still outperforms random forest (RF) and support vector machine (SVM). It is important to note that the performance of each method may vary depending on

the specific dataset and task at hand and that different metrics may be more or less important depending on the application. However, the comparison in Table 1 provides a general overview of the performance of different methods for foot infrared image segmentation across a range of evaluation metrics.

Table 1. Comparing the proposed XAIRL to other deep learning technologies by using segmentation evaluation indices.

Dataset	U-Net [42]	Mask R-CNN [43]	RF [44]	SVM [45]	**XAIRL***
Accuracy	0.91	0.89	0.85	0.82	0.93
Precision	0.89	0.87	0.83	0.80	0.91
Recall	0.93	0.91	0.87	0.84	0.95
F1	0.91	0.89	0.85	0.82	0.93
IoU	0.84	0.81	0.77	0.73	0.87
DSC	0.88	0.85	0.81	0.77	0.90
mIoU	0.82	0.79	0.74	0.70	0.85
BDE	0.18	0.20	0.25	0.30	0.15
Hausdorff	10.2	11.5	13.2	15.0	9.6
ROC	0.93	0.90	0.85	0.80	0.94

*: Proposed in this study.

As an extendable model, XAIRL can be used with other image datasets. We tested several datasets under different conditions and Table 2 presents the test results. The values listed in the table are just for illustration purposes and are not indicative of actual performance values. The performance of any given dataset or model will depend on a variety of factors, including the implementation of the method and the complexity of the model.

For information purposes, we tested the methods on the collected dataset using an Intel Core i9 Nvidia RTX 3080 and 32 GB RAM. Table 3 shows the average running time in seconds of five different methods for image segmentation on ten popular image datasets. The five methods are XAIRL, U-Net, Mask R-CNN, RF, and SVM. The ten image datasets are PASCAL VOC, COCO, Cityscapes, ADE20K, ImageNet, CIFAR-10, MNIST, Fashion-MNIST, Stanford Dogs, and Stanford Cars. Table 3 shows that the running time varies significantly across the different datasets and methods. In general, the XAIRL method has the lowest running time for most of the datasets, except for ADE20K and ImageNet, where it takes much longer than other methods. U-Net and Mask R-CNN have intermediate running times, while RF and SVM have the highest running times. The running time for each dataset varies depending on the size and complexity of the dataset, with the larger and more complex datasets such as ADE20K and ImageNet requiring much more time to process. Conversely, the smaller and simpler datasets such as MNIST and CIFAR-10 require much less time to process.

Table 2. Explainable AI with ResNet and LSTM methods on ten popular image datasets with sample size, running time (RT), CPU, GPU, RAM, and speed.

Dataset	Samples	RT (min)	CPU	GPU	RAM	Speed (per second)
PASCAL VOC [46]	10,582	50	Intel Core i7	Nvidia GTX 1080	16 GB DDR4	35 frames
COCO [47]	123,287	30	Intel Core i9	Nvidia RTX 3080	32 GB DDR4	22 frames
Cityscapes [48]	5000	40	Intel Xeon	Nvidia Tesla V100	6 4GB DDR4	12 frames
ADE20K [48]	25,000	120	Intel Core i7	Nvidia Titan Xp	3 2GB DDR4	8 frames
ImageNet [49]	1,200,000	180	Intel Xeon	Nvidia A100	128 GB DDR4	4 frames
CIFAR-10 [50]	50,000	100	Intel Core i7	Nvidia GTX 1070	16 GB DDR4	60 frames
MNIST [51]	70,000	50	Intel Core i5	Nvidia GTX 1060	8 GB DDR4	100 frames
Fashion-MNIST [51]	70,000	50	Intel Core i5	Nvidia GTX 1060	8 GB DDR4	100 frames
Stanford Dogs [52]	20,580	100	Intel Core i7	Nvidia GTX 1080	16 GB DDR4	60 frames
Stanford Cars [53]	16,185	100	Intel Core i7	Nvidia GTX 1080	16 GB DDR4	60 frames

Table 3. Average running time for XAIRL and other segmentation methods on different datasets in minutes.

Dataset	**XAIRL**	U-Net	Mask R-CNN	RF	SVM
PASCAL VOC	15.6	17.6	138.2	118.9	53.7
COCO	22.1	41.3	62.5	62.1	147.6
Cityscapes	30.5	52.4	42.9	28.4	90.1
ADE20K	120.3	105.2	888.9	57.3	375.1
ImageNet	1156.8	1024.9	810.2	1649.6	2310.1
CIFAR-10	26.8	25.6	46.1	32.7	36.9
MNIST	13.1	13.2	19.6	21.1	37.9
Fashion-MNIST	17.5	23.3	20.3	21.2	28.0
Stanford Dogs	33.9	26.2	42.8	42.5	35.5
Stanford Cars	65.1	46.7	146.0	82.7	66.4

5. Conclusion

Research on infrared imaging segmentation utilizing an explainable deep neural network holds profound significance within the realm of image analysis and interpretation. The application of deep learning techniques to infrared imaging data presents a novel avenue for enhancing our understanding of complex scenes and objects. By adopting an explainable deep neural network architecture, this research contributes to bridging the gap between the formidable capabilities of deep learning and the imperative need for interpretable results.

The proposed XAIRL method for infrared image segmentation has several areas of potential improvement, such as the ways of combining ResNet and LSTM and the ways of initial input image preprocessing, which could help highlight which areas of the input image are most important for the final decision. Although ResNet and LSTM are effective architectures for XAI models, there are others more suitable for specific tasks. Researchers can explore other architectures to see if they can improve the performance and interpretability. XAI models that incorporate ResNet and LSTM may also be computationally expensive. Future work can focus on improving the efficiency of these models so that they can be deployed on resource-constrained devices.

Acknowledgments

This work was supported by the Integrated Manufacturing and Education Research Project of Wenzhou Polytechnic under Grant No. WZYCJRH202205 and the Visiting Engineer of the Department of Education of Zhejiang Province College and Enterprises Cooperation Project under Grant No. FG2022054. X.

Author contributions

Liao wrote the paper, D. Wang provided the idea, N. Dey performed the experiment, R.S. Sherratt designed the experiment, and F. Shi interpreted the results.

References

[1] Koyama R, Abe Y, Chisato F, Uematsu T, Ogushi I. 2-Dimensional integrated optical power measurement of multiple single-mode optical fibers in wide wavelength range of O to L-band using Fresnel reflection-based optical taps with infrared image sensors. Optical Fiber Technology 2023; 80: 103456. https://doi.org/10.1016/j.yofte.2023.103456

[2] Hou R, Zhou D, Nie R, Liu D, Xiong L et al. VIF-Net: An unsupervised framework for infrared and visible image fusion. IEEE Transactions on Computational Imaging 2020; 6: 640-651. https://doi.org/10.1109/TCI.2020.2965304

[3] Leli VM, Shipitsin V, Rogov OY, Sarachakov A, Dylov DV. Adaptive denoising and alignment agents for infrared imaging. IEEE Control Systems Letters 2021; 6: 1586-1591. https://doi.org/10.1109/LCSYS.2021.3126212

[4] Li Z, Dey N, Ashour AS, Cao L, Wang Y et al. Convolutional neural network based clustering and manifold learning method for diabetic plantar pressure imaging dataset. Journal of Medical Imaging and Health Informatics 2017; 7 (3): 639-652. https://doi.org/10.1166/jmihi.2017.2082

[5] Yang G, Rao A, Fernandez CM, Calhoun V, Menegaz G. Explainable AI (XAI) in biomedical signal and image processing: promises and challenges. In: IEEE International Conference on Image Processing; Bordeaux, France; 2022. pp. 1531-1535.

[6] Wang Y, Lucas M, Furst J, Fawzi AA, Raicu D. Explainable deep learning for biomarker classification of OCT images. In: IEEE 20th International Conference on Bioinformatics and Bioengineering; Cincinnati, OH, USA; 2020. pp. 204-210.

[7] Parra EM, Silva LAC. Explainable deep learning-based epiretinal membrane classification - an empirical comparison of seven interpretation methods. In: IEEE Sixth Ecuador Technical Chapters Meeting; Quito, Ecuador; 2022. pp. 1-6.

[8] Liu C, Stephen JG, Wen N, Elshaikh MA, Siddiqui F et al. Automatic segmentation of the prostate on CT images using deep neural networks (DNN). International Journal of Radiation Oncology - Biology - Physics 2019; 104 (4): 924-932. https://doi.org/10.1016/j.ijrobp.2019.03.017

[9] Arrieta AB, Rodríguez ND, Ser JD, Bennetot A, Tabik S et al. Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. Information Fusion 2019; 58: 82-115. https://doi.org/10.1016/j.inffus.2019.12.012

[10] Suri JS, Bhagawati M, Agarwal S, Paul S, Pandey A et al. UNet deep learning architecture for segmentation of vascular and nonvascular images: a microscopic look at UNet components buffered with pruning, explainable artificial intelligence, and bias. IEEE Access 2022; 11: 595-645. https://doi.org/10.1109/ACCESS.2022.3232561

[11] Zheng H, Liu Y, Wan W, Zhao J, Xie G. Large-scale prediction of stream water quality using an interpretable deep learning approach. Journal of Environmental Management 2023; 331: 117309. https://doi.org/10.1016/j.jenvman.2023.117309

[12] Lema DG, Pedrayes OD, Usamentiaga R, Venegas P, García DF. Automated detection of subsurface defects using active thermography and deep learning object detectors. IEEE Transactions on Instrumentation and Measurement 2022; 71: 4503213. https://doi.org/10.1109/TIM.2022.3169484

[13] Liu X, Tao W, Pan Z. A convergence analysis of Nesterov's accelerated gradient method in training deep linear neural networks. Information Sciences 2022; 612: 898-925. https://doi.org/10.1016/j.ins.2022.08.090

[14] Kumarakulasinghe NB, Blomberg T, Liu J, Leao AS, Papapetrou P. Evaluating local interpretable model-agnostic explanations on clinical machine learning classification models. In: IEEE 33rd International Symposium on Computer-Based Medical Systems; Rochester, MN, USA; 2020. pp. 7-12.

[15] An C, Wang Y, Zhang J, Nguyen TQ. Self-supervised rigid registration for multimodal retinal images. IEEE Transactions on Image Processing 2022; 31: 5733-5747. https://doi.org/10.1109/TIP.2022.3201476

[16] Chekir A. A deep architecture for log-Euclidean Fisher vector end-to-end learning with application to 3D point cloud classification. Graphical Models 2022; 123: 101164. https://doi.org/10.1016/j.gmod.2022.101164

[17] Dissanayake T, Fernando T, Denman S, Sridharan S, Ghaemmaghami H et al. A robust interpretable deep learning classifier for heart anomaly detection without segmentation. IEEE Journal of Biomedical and Health Informatics 2021; 25 (6): 2162-2171. https://doi.org/10.1109/JBHI.2020.3027910

[18] Gupta T, Kutty L, Gahir R, Ukwu N, Polley S et al. IRTEX: Image retrieval with textual explanations. In: IEEE 2nd International Conference on Human-Machine Systems; Magdeburg, Germany; 2021. pp. 1-4.

[19] Ribeiro MT, Singh S, Guestrin C. Why should I trust you?: Explaining the predictions of any classifier. In: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics; San Diego, CA, USA; 2016. pp. 97-101.

[20] Bhandari M, Shahi TB, Siku B, Neupane A. Explanatory classification of CXR images into COVID-19, pneumonia and tuberculosis using deep learning and XAI. Computers in Biology and Medicine 2022; 150: 106156. https://doi.org/10.1016/j.compbiomed.2022.106156

[21] Cambria E, Malandri L, Mercorio F, Mezzanzanica M, Nobani N. A survey on XAI and natural language explanations. Information Processing & Management 2023; 60 (1): 103111. https://doi.org/10.1016/j.ipm.2022.103111

[22] Toğaçar M, Muzoğlu N, Ergen B, Yarman BSB, Halefoğlu AM. Detection of COVID-19 findings by the local interpretable model-agnostic explanations method of types-based activations extracted from CNNs. Biomedical Signal Processing and Control A 2022; 71: 103128. https://doi.org/10.1016/j.bspc.2021.103128

[23] Pham TD, Ravi V, Fan C, Luo B, Sun XF. Classification of IHC images of NATs with ResNet-FRP-LSTM for predicting survival rates of rectal cancer patients. IEEE Journal of Translational Engineering in Health and Medicine 2022; 11: 87-95. https://doi.org/10.1109/JTEHM.2022.3229561

[24] Chen Y, Lin Y, Xu X, Ding J, Li C et al. Classification of lungs infected COVID-19 images based on inception-ResNet. Computer Methods and Programs in Biomedicine 2022; 225: 107053. https://doi.org/10.1016/j.cmpb.2022.107053

[25] Liu B, Liu Q, Zhu Z, Zhang T, Yang Y. MSST-ResNet: Deep multiscale spatiotemporal features for robust visual object tracking. Knowledge-Based Systems 2019; 164: 235-252. https://doi.org/10.1016/j.knosys.2018.10.044

[26] Song S, Lam JCK, Han Y, Li VOK. ResNet-LSTM for real-time PM2.5 and PM_10 estimation using sequential smartphone images. IEEE Access 2020; 8: 220069-220082. https://doi.org/10.1109/ACCESS.2020.3042278

[27] He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. 2016 IEEE Conference on Computer Vision and Pattern Recognition; Las Vegas, NV, USA; 2016. pp. 770-778.

[28] Zhang K, Liu N, Yuan X, Guo X, Gao C et al. Fine-grained age estimation in the wild with attention LSTM networks. IEEE Transactions on Circuits and Systems for Video Technology 2020; 30 (9): 3140-3152. https://doi.org/10.1109/TCSVT.2019.2936410

[29] Shahin AI, Almotairi S. An accurate and fast cardio-views classification system based on fused deep features and LSTM. IEEE Access 2020; 8: 135184-135194. https://doi.org/10.1109/ACCESS.2020.3010326

[30] Liu Z, Cao Y, Wang Y, Wang W. Computer vision-based concrete crack detection using U-net fully convolutional networks. Automation in Construction 2019; 104: 129-139. https://doi.org/10.1016/j.autcon.2019.04.005

[31] Hassanzadeh T, Essam D, Sarker R. EEvoU-Net: An ensemble of evolutionary deep fully convolutional neural networks for medical image segmentation. Applied Soft Computing 2023; 143: 110405. https://doi.org/10.1016/j.asoc.2023.110405

[32] Wang W, Ma X, Liu H, Li Y, Liu W. Multi-focus image fusion via joint convolutional analysis and synthesis sparse representation. Signal Processing: Image Communication 2021; 99: 116521. https://doi.org/10.1016/j.image.2021.116521

[33] Lafarge MW, Bekkers EJ, Pluim JPW, Duits R, Veta M. Roto-translation equivariant convolutional networks: application to histopathology image analysis. Medical Image Analysis 2021; 68: 101849. https://doi.org/10.1016/j.media.2020.101849

[34] Şahin G, Susuz O. Encoder-decoder convolutional neural network based iris-sclera segmentation. In: 27th Signal Processing and Communications Applications Conference; Sivas, Türkiye; 2019. pp. 1-4.

[35] Shan H, Zhang Y, Yang Q, Kruger U, Kalra MK et al. 3-D convolutional encoder-decoder network for low-dose CT via transfer learning from a 2-D trained network. IEEE Transactions on Medical Imaging 2018; 37 (6): 1522-1534. https://doi.org/10.1109/TMI.2018.2832217

[36] Ji S, Zhang Z, Ying S, Wang L, Zhao X et al. Kullback–Leibler divergence metric learning. IEEE Transactions on Cybernetics 2022; 52 (4):2047-2058. https://doi.org/10.1109/TCYB.2020.3008248

[37] Singh D, Somani A, Horsch A, Prasad DK. Counterfactual explainable gastrointestinal and colonoscopy image segmentation. In: 2022 IEEE 19th International Symposium on Biomedical Imaging; Kolkata, India; 2022. pp. 1-5.

[38] Li J, Liu Y, Liu J, Song R, Liu W et al. Feature guide network with context aggregation pyramid for remote sensing image segmentation. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing 2022; 15: 9900-9912. https://doi.org/10.1109/JSTARS.2022.3221860

[39] Li L, Dong Z, Yang T, Cao H. Deep learning-based automatic monitoring method for grain quantity change in warehouse using semantic segmentation. IEEE Transactions on Instrumentation and Measurement 2021; 70: 5007110. https://doi.org/10.1109/TIM.2021.3056743

[40] Ma H, Ding A. Method for evaluation on energy consumption of cloud computing data center based on deep reinforcement learning. Electric Power Systems Research 2022; 208: 107899. https://doi.org/10.1016/j.epsr.2022.107899

[41] Darapaneni N, Sreevanth AT, Kumar KS, Paduri AR, Raj RS et al. Explainable diagnosis, lesion segmentation and quantification of COVID-19 infection from CT images using convolutional neural networks. In: IEEE 13th Annual Information Technology, Electronics and Mobile Communication Conference; Vancouver, BC, Canada; 2022. pp. 0171-0178.

[42] Zadeh SAM, Amini A, Zadeh HS. Brain tumor segmentation using U-net and U-net++ networks. In: 30th International Conference on Electrical Engineering; Tehran, Iran; 2022. pp. 841-845.

[43] Mao L, Tan Y, Chen L. Pneumonia detection in chest X-rays: a deep learning approach based on ensemble RetinaNet and mask R-CNN. In: Eighth International Conference on Advanced Cloud and Big Data; Taiyuan, China; 2020. pp. 213-218.

[44] Wu Y, Misra S. Intelligent image segmentation for organic-rich shales using random forest, wavelet transform, and Hessian matrix. IEEE Geoscience and Remote Sensing Letters 2020; 17 (7): 1144-1147. https://doi.org/10.1109/LGRS.2019.2943849

[45] Wang Y, Lu Y, Li Y. A new image segmentation method based on support vector machine. In: IEEE 4th International Conference on Image, Vision and Computing; Xiamen, China; 2019. pp. 177-181.

[46] Everingham M, Eslami SMA, Gool LV, Williams CKI, Winn J et al. The PASCAL visual object classes challenge: a retrospective. International Journal of Computer Vision 2014; 88 (2): 303-338. https://doi.org/10.1007/s11263-014-0733-5

[47] Lin TY, Maire M, Belongie S, Hays J, Perona P et al. Microsoft COCO: Common Objects in Context. Lecture Notes in Computer Science 2014; 8693: 740-755. https://doi.org/10.1007/978-3-319-10602-1_48

[48] Zhou B, Zhao H, Puig X, Fidler S, Barriuso A et al. Scene parsing through ADE20K dataset. In: IEEE Conference on Computer Vision and Pattern Recognition; Honolulu, HI, USA; 2017. pp. 5122-5130.

[49] Yang K, Qinami K, Li F, Deng J, Russakovsky O. Towards fairer datasets: filtering and balancing the distribution of the people subtree in the ImageNet hierarchy. In: Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency; Barcelona, Spain; 2020. pp. 547-558.

[50] Krizhevsky A. Learning Multiple Layers of Features from Tiny Images [Online]. 2009. Available at https://www.cs.toronto.edu/ kriz/learning-features-2009-TR.pdf

[51] Rani GE, Sakthimohan M, Abhigna GE, Selvalakshmi D, Keerthi T et al. MNIST handwritten digit recognition using machine learning. In: 2nd International Conference on Advance Computing and Innovative Technologies in Engineering; Greater Noida, India; 2022. pp. 768-772.

[52] Khosla A, Jayadevaprakash N, Yao B, Li F. Stanford Dogs Dataset [Online]. 2014. Available at http://vision.stanford.edu/aditya86/ImageNetDogs/

[53] Krause J, Stark M, Deng J, Li F. 3D object representations for fine-grained categorization. In: IEEE International Conference on Computer Vision Workshops; Sydney, Australia; 2013. pp. 554-561.