Computers in Biology and Medicine 170 (2024) 108077

Contents lists available at ScienceDirect

Computers in Biology and Medicine

journal homepage: www.elsevier.com/locate/compbiomed

# Gait disorder classification based on effective feature selection and unsupervised methodology

Mohsen Shayestegan [a], Jan Kohout [b], Kateřina Trnková [c], Martin Chovanec [c], Jan Mareš [a,b,*]

[a] University of Pardubice, Faculty of Electrical Engineering and Informatics, Nam. Cs. Legii 565, Pardubice, 530 02, Czech Republic
[b] University of Chemistry and Technology Prague, Czech Republic, Department of Mathematics, Informatics and Cybernetics, Technická 1905/5, Prague, 166 28, Czech Republic
[c] Charles University Prague, 3rd Faculty of Medicine, Department of Otorhinolaryngology, University Hospital Kralovske Vinohrady, Šrobárova 1150/50, Prague, 100 34, Czech Republic

ARTICLE INFO

*Keywords:*
Deep learning
Classification
Gait disorders
Vision transformer
Autoencoder
Discriminator

ABSTRACT

In gait stability analysis, patients suffering from dysfunction problems are impacted by shifts in their dynamic balance. Monitoring the patients' progress is important for allowing physicians and patients to observe the rehabilitation process accurately. In this study, we designed a new methodology for classifying gait disorders to quantify patients' progress. The dataset in this study includes 84 measurements of 37 patients based on a physician's opinion. In this study, the system, which includes a Kinect camera to observe and store the frames of patients walking down a hallway, a key-point detector to detect the skeletal key points, and an encoder transformer classifier network integrated with generator–discriminator networks (ET-GD), is designed to evaluate the classification of gait dysfunction. The detector extracts the skeletal key points of patients. After feature engineering, the selected high-level features are fed into the proposed neural network to analyse patient movement and perform the final evaluation of gait dysfunction. The proposed network is inspired by the 1D encoder transformer, which is integrated with two main networks: a network for classification and a network to generate fake output data similar to the input data. Furthermore, we used a discriminator structure to distinguish between the actual data (input) and fake data (generated data). Due to the multi-structural networks in the proposed method, multi-loss functions need to be optimised; this increases the accuracy of the encoder transformer classifier.

## 1. Introduction

The rapid integration of IT systems and artificial intelligence (AI) across various domains has led to notable advancements, particularly in medicine and diagnostics. In particular, AI aids in the analysis of medical images, such as CT and NMR scans, alerting radiologists to potential areas of concern [1]. Moreover, it processes biochemical data, enabling the development of novel diagnostic procedures based on dynamic data analysis, which are often imperceptible to the human eye, thus enhancing accuracy [2,3].

A specific application of AI in diagnostics involves automatic gait analysis, addressing diverse factors influencing walking patterns, from orthopaedic to neurological conditions [4–6]. Human activity recognition (HAR) systems, using sensors such as Kinect, play a pivotal role in gait analysis, extracting relevant features for classification algorithms [7,8]. Vision-based studies, including per-pixel classification and skeleton-based methods, highlight the evolution of HAR systems [7,9]. In particular, the effectiveness of Kinect in measuring gait parameters is explored, emphasising the impact of joint angles on the accuracy of the system [10].

Machine learning methods, including Bayesian classifiers and deep learning, have been pivotal in gait disorder recognition, overcoming challenges posed by complex health datasets [10,11]. Deep neural networks (DNNs), specifically CNNs and RNNs, automate pattern extraction from intricate data, showing promise in abnormal gait recognition [12]. Recent studies also highlight the fusion of RNNs and CNNs for improved performance [12].

### 1.1. Biomedical background

Biomedical background underscores the significance of gait analysis in diagnosing disorders, with postural stability and gait intricately

* Corresponding author at: University of Chemistry and Technology Prague, Czech Republic, Department of Mathematics, Informatics and Cybernetics, Technická 1905/5, Prague, 166 28, Czech Republic.
*E-mail addresses:* mohsen.shayestegan@upce.cz (M. Shayestegan), jan.kohout@vscht.cz (J. Kohout), katerina.trnkova@fnkv.cz (K. Trnková), martin.chovanec@fnkv.cz (M. Chovanec), jan.mares@vscht.cz (J. Mareš).

https://doi.org/10.1016/j.compbiomed.2024.108077
Received 24 August 2023; Received in revised form 10 January 2024; Accepted 27 January 2024
Available online 30 January 2024

**Table 1**
Hospital rehabilitation exercises.

| Order | Exercise |
|---|---|
| 1 | Normal walking |
| 2 | Walking along a line |
| 3 | Walking with eyes closed |

linked to neurological and orthopaedic health [13,14]. Spatiotemporal parameters such as walking speed, cadence, and stride length serve as crucial indicators, forming the basis for clinical evaluations [14]. The systematic breakdown of the gait cycle aids in the identification of abnormalities, emphasising the importance of a comprehensive analysis [14].

Dynamic stability assessments, crucial in understanding gait disorders, often rely on clinical tests like the Timed Up and Go Test and Six-Minute Walk Test [15]. However, there are limitations in quantifying gait stability in real-life scenarios, leading to the need for more objective examinations [16].

Gait disorder analysis is the extensive research on automatic gait analysis, spanning from Parkinson's disease diagnosis using SVM [17] to unsupervised learning approaches for cerebral palsy [18]. In particular, studies explore gait rehabilitation in patients with stroke and traumatic brain injury, showcasing the diverse applications of gait disorder analysis in healthcare care [19,20].

The evolving landscape of AI methodologies in gait disorder analysis involves a shift towards transformer networks, particularly for their ability to capture long-term dependencies [21,22]. Furthermore, generator–discriminator networks, such as GANs, demonstrate potential to generate new samples and improve classification accuracy [23,24]. This study incorporates an encoder transformer for temporal information and a generator–discriminator network to improve overall classification performance [25].

### 1.2. Main aims

Our goal was to design a system that can achieve a higher accuracy in classifying gait disorders to help physicians in hospitals and clinics and to allow patients to visualise their progress. The development of the proposed framework involved designing a suitable deep learning structure to reach a higher classification accuracy with preprocessed image data, which are recorded by a Kinect camera. Compared to existing evaluation methods, in addition to designing a novel deep learning algorithm, the main contribution of the proposed framework is that it can create and select significant features that can improve the classification results. These features can be created from the skeleton key points extracted from the patient in each image; they can include the angle and distance between particular skeletal key points or the 2D pixel coordinates of these points. The key contributions of the proposed framework are as follows:

1. Novel and effective methods for feature creation, feature selection, and the mapping of extracted skeleton features into a vector of given dimensions are proposed to make the proposed deep network more suitable for gait disorder classification.
2. A novel algorithm that is a combination of a 1D vision transformer and an unsupervised methodology with a modified optimisation function is designed to improve the performance and increase the classification accuracy.

## 2. Materials and methods

### 2.1. Measurement scheme

The measured process consists of the different walking exercises presented in Table 1. A patient is asked to perform these exercises in a hospital hallway, and the frames of the patient's gait are recorded.

**Table 2**
Dataset overview.

| | |
|---|---|
| Start date | 16-01-2018 |
| End date | 13-12-2020 |
| Number of patients | 37 |
| Number of sessions | 84 |
| Number of men | 23 |
| Number of women | 14 |
| Average age | 56.6 years |

### 2.2. Dataset

The dataset contains 84 successful rehabilitation exercises for 37 patients (Table 2). All patients (23 males/14 females with ages from 21–77 years) needed surgery for vestibular schwannoma (24 left side/13 right side; Koos classification: 10 grade 1/10 grade 2/6 grade 3/11 grade 4a tumours; International classification: 10 grade T0/9 grade T1/8 grade T2/9 grade 3/0 grade 4/1 grade 5 tumours). The selected dataset was designed to study specific gait disturbance associated with sudden peripheral vestibular deafferentation due to vestibular neurectomy and eventual labyrinthectomy. At the same time, only probands without preoperative vestibular dysfunction were analysed. This was based on the clinical examination as well as results of video head impulse test and video oculography including caloric test.

At the same time, the exclusion criteria were the presence of visual impairment, contralateral vestibular pathology, musculoskeletal or neurological disease that could lead to impaired stability and gait, and interfere with the course of vestibular compensation. The proposed biomedical model is studied in detail including the processes associated with the progression of vestibular compensation and its impact on postural stability. Moreover, the course of vestibular compensation can be assessed by the above mentioned instrumental neuro-otological examination techniques and thus eventual gait disturbance can be directly correlated to the degree of vestibular deficit. An undeniable advantage is the time sequence of the course of the compensation, where in an otherwise uncomplicated situation the compensation occurs within 12 weeks regardless of age.

### 2.3. Data preprocessing

The most important step in solving a classification problem is obtaining valuable data to use to train the deep neural network. Data selection in the analysis of gait disorders is based on expert knowledge of the patient's movement during the assessment. Selecting significant features before feeding the data into the network can help the network to distinguish between normal and unbalanced walking for each patient during the assessment, resulting in a higher classification accuracy. The raw data collected with the Kinect camera and robot platform consist of around 84 sessions. Each session includes three exercises, and each exercise involves a patient walking down a hallway. As shown in Fig. 1, there are 47 sessions in class 1 (light imbalance before surgery), 19 sessions in class 2 (heavy imbalance before surgery), and 18 sessions in class 3 (no imbalance before surgery).

The framework of the proposed data processing method is illustrated in Fig. 2, and it includes three steps: data preprocessing, keypoint feature detection, and feature selection before training. In the preprocessing step, we applied different image filters to obtain more detection results due to the environment in which the data were collected. The output image in the preprocessing step is used to extract the skeleton key points in combination with the Kalman filter [26,27] to determine if there are missing points in the detection results. Then, these extracted key points are used for feature generation and selection. For gait disorders classification, all the features of the body are significant. However, some features are more critical than others [28]. The

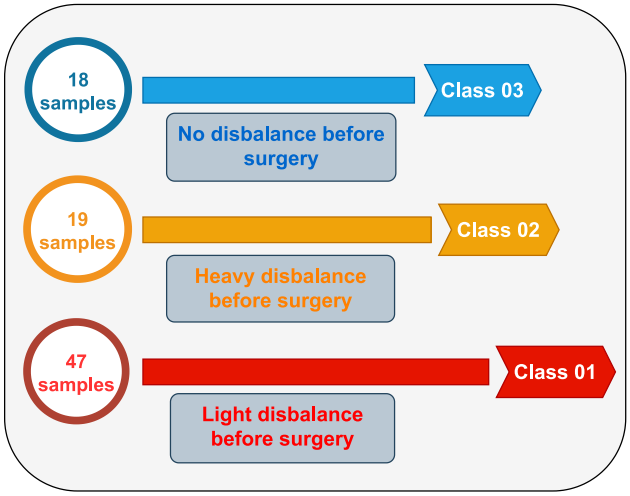


**Fig. 1.** Distribution of classes in our dataset.

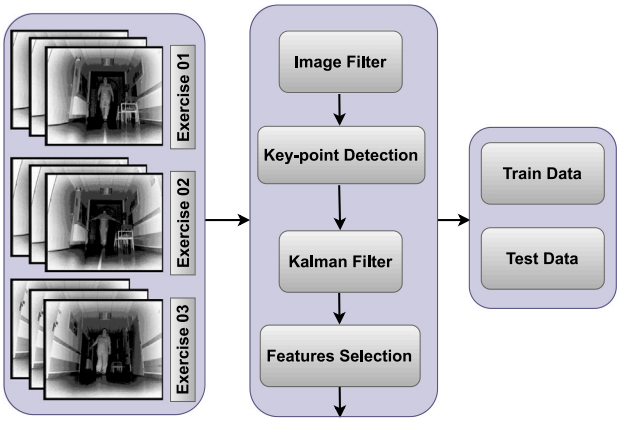


**Fig. 2.** Architecture of the proposed data processing method.

study [29] found that the swing of arms is the most essential feature that can increase the set of characteristics. Humans can find complex patterns even if they do not exist. So, we have feature creation to derive new features from existing ones. We also used feature scaling techniques to help with normalising data. These are the main concepts of feature engineering in our project. Regarding feature creation, we have applied the concept of clinicians to categorise gait disorders as they focus mainly on the patient's hands and legs during exercises. Due to these reasons, we applied four statistical measurements: $x$ and $y$ for a selected point, the distance and the angle between two symmetric key points, and the angle between two bones. All of these measurements are confirmed by clinicians to be tested with a practical machine-learning model. After feature selection, the final dataset is split into training and test datasets. Furthermore, due to our imbalanced and small dataset, we have applied the techniques recommended in [6], such as the use of optimal weights for each class in the training process, the F-score metric, and data augmentation.

For key-point detection, we applied the key-point-RCNN detector built on the ResNet-50 FPN backbone [30] in the Torchvision library. The Keypoint-RCNN acts like the Mask-RCNN, where the key points are encoded in the keypoint mask instead of the whole mask. Mask R-CNN is extended of Faster R-CNN by adding an object mask prediction in parallel with bounding box recognition [31]. Mask R-CNN is easy to generalise to other tasks, such as human pose estimation in the same framework [31]. The Mask-RCNN was trained on the MS-COCO dataset [32] and offers key points only for the person class. There are

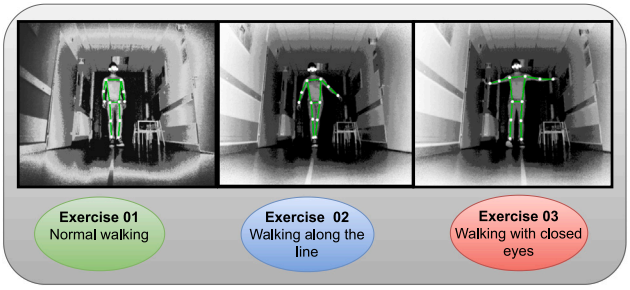


**Fig. 3.** Typical patient performing the three exercises.

**Table 3**
Points of interest (key-point locations).

| Point | Position | Point | Position |
|---|---|---|---|
| 1 | Nose | 10 | Left wrist |
| 2 | Left eye | 11 | Right wrist |
| 3 | Right eye | 12 | Left hip |
| 4 | Left ear | 13 | Right hip |
| 5 | Right ear | 14 | Left knee |
| 6 | Left shoulder | 15 | Right knee |
| 7 | Right shoulder | 16 | Left ankle |
| 8 | Left elbow | 17 | Right ankle |
| 9 | Right elbow | | |

17 key points in the output of the detector as a 2D vector (x and y) for the position of points. The results produced by the key-point detector for our dataset are shown in Fig. 3. As shown, 17 key points are used to obtain a skeleton of the patient's body.

As a result of the visualisation and observation of the detection performance for our data, we decided to use a window size of 100 frames. The key points are 2D pixel coordinates that are used for feature engineering. Four statistical measurements (i.e. $x$ and $y$ for two selected points and the distance and angle between these points) were calculated from the pixel coordinates. Table 3 provides the locations of the key points. Fig. 4 illustrates a patient with the skeleton key points and the selected features used to train the proposed model labelled. The green arrows show the distance between two selected points, and the blue curves show the angle between two selected bones. These spatial features are synthesised by monitoring the patient's walking, and the following statistical measures are used [27]:

$$d_i = \sqrt{\Delta x_i^2 + \Delta y_i^2}, \tag{1}$$

$$\alpha_{1_i} = atan2(\Delta y_i, \Delta x_i), \tag{2}$$

$$\alpha_{2i} = arccos(\frac{v_{1i} \cdot v_{2i}}{|v_{1i}| \times |v_{2i}|}), \tag{3}$$

where $d_i$ is the distance between two key points, $\alpha_{1_i}$ is the angle between two symmetric key points, and $\alpha_{2i}$ is the angle between two bones.

There are 70 features in each frame, as there are 34 features for the coordinates of 17 pixels, eight distance variables (horizontal green arrows on the left skeleton) for symmetric key-point pairs, eight angle variables (angles of the horizontal green arrows on the left skeleton) for symmetric key-point pairs, eight angles (blue curves on the left skeleton) between selected bones, and 12 distance variables between pairs of key-points (vertical green arrows on the right skeleton); all of these features are shown in Fig. 4. Each data session includes the recorded frames of the three exercises, along with the class label. We applied a window size of 100 frames for each exercise to collect 300 frames (100 × 3) for each labelled session, so the sample file for each session has 300 rows (sequences) and 70 columns (features).

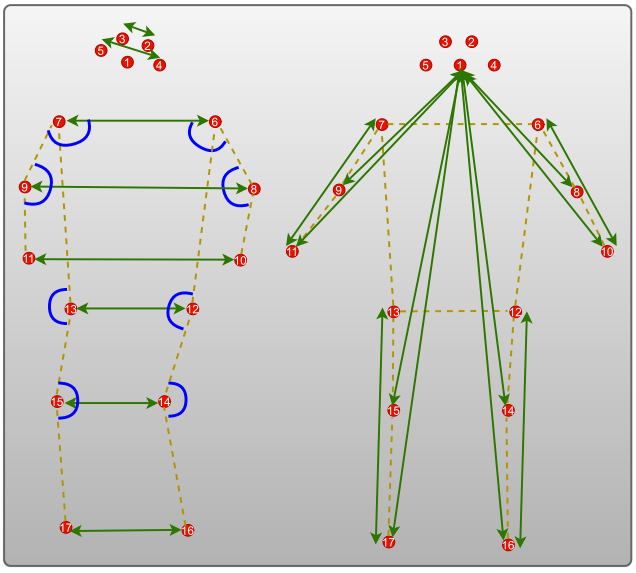

**Fig. 4.** Skeleton key-point positions and the selected features used for training.

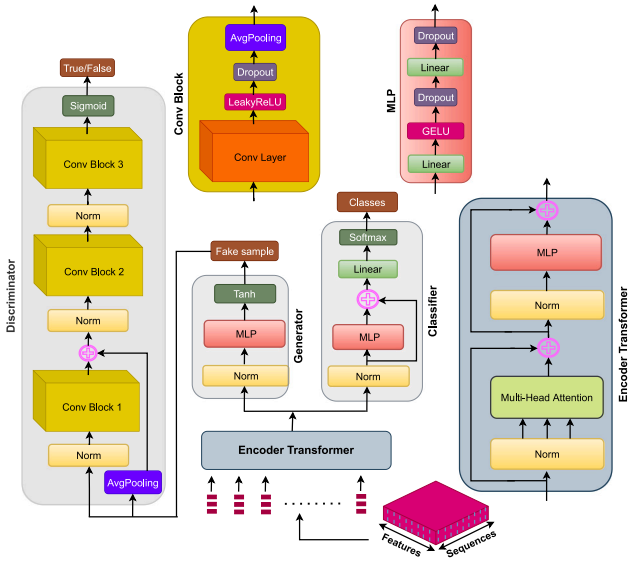


**Fig. 5.** Representation of the proposed framework.

### 2.4. Classification methodology

The structure of the proposed method is illustrated in Fig. 5. The framework includes four main module networks: the transformer, generator, classifier, and discriminator. We proposed an encoder-transformer classifier that is integrated with a generator–discriminator network (ET-GD) to evaluate the dynamic stability of patients based on the selected extracted features of each patient's skeletal movement as they perform three standard walking exercises. The reason why we applied the transformer technique instead of the common recurrent neural networks is that recurrent neural networks such as RNN and LSTM [33] evaluate the input in an ordered sequence format, but in contrast, the transformer models analyse the entire sequence of input [34]. This method helps to learn the temporal connections of the sequence and identify the patterns with long-term dependencies by using a self-attention process. These capabilities make the transformer models more suitable for long-term sequential data.

In our study, the generator–discriminator networks include a generator network, which includes encoder and decoder networks. The encoder performs feature extraction on raw data, and the decoder recreates the input data as a fake sample form feature in the bottleneck for input of the discriminator network. The classifier also uses these extracted features from the bottleneck for classification. The networks compute the loss and backpropagate through the models. By optimising the parameters of the training process, the best features are extracted in the bottleneck, where the main purpose is to get a better classification result than generating fake samples. Therefore, the model learns to improve the classification results by defining loss functions and their learning weights.

Fig. 5 illustrates the framework of the proposed ET-GD classifier. For each exercise, only 100 sequences are selected due to our observations and the performance of the key-point detector on our dataset. As each session included three exercises, the results of these three exercises are concatenated to create the input for our proposed network.

At the beginning of the proposed framework, the encoder transformer layer [35] with an attention mechanism is applied to the batch of feature sequences generated by the extracted skeletal features. For the encoder transformer, instead of using fixed-size image patches (2D) [35], we used a modified transformer for 1D feature vectors (70 features in each vector), with two layers and two heads in the encoder transformer. We remove the positional encoding [35], as we observe that this enhances the performance of the proposed model. Then, the resulting sequences of vectors are fed into the encoder transformer.

As shown in Fig. 5, the classification network is constructed with a normalisation layer, a fully connected multilayer perceptron (MLP), a linear layer, and a softmax activation function. In addition, to increase the classification accuracy, this base structure (encoder transformer plus classifier) is integrated with the generator and discriminator networks. The generator is constructed with a normalisation layer, a fully connected neural network, and a tanh activation function. The generator generates fake samples, which are fed into the discriminator as input. As can be seen from Fig. 5, there are several fully connected multilayer neural networks in the proposed method, so to prevent overfitting, the dropout technique is used.

In the discriminator network, as shown in Fig. 5, there are three convolutional blocks, three normalisation layers, an average pooling layer, and a sigmoid activation function. The discriminator network distinguishes actual samples from generated sample data. These inputs are fed into the discriminator during the training procedure, and there is a gradient penalty that is used to update both the generator and discriminator networks. If the two inputs are classified as similar, the output should be one; when they are not classified as similar, the output should be zero [36].

As can be seen from Fig. 5, we used a normalised layer in each module network of our proposed method. In each module of the proposed model, the input is first normalised using layer normalisation [37], and then it is fed into the network of that particular module. We used residual connections [38] in some parts of our network to connect the previous layer's output to the next layer's input to allow the model to transmit the practical features of the lower layer to the higher layers [39]. For any vector v, the layer normalisation is calculated as follows:

$$LayerNorm(v) = \gamma \frac{v - \mu}{\sigma} + \beta, \tag{4}$$

$$\mu = \frac{1}{d} \sum_{k=1}^{d} v_k, \tag{5}$$

$$\sigma^2 = \frac{1}{d} \sum_{k=1}^{d} (v_k - \mu)^2. \tag{6}$$

In addition, as shown in Fig. 5, the MLP blocks in the proposed module networks are composed of a two-layer feedforward network with dropout and a GELU activation function. Given a sequence of

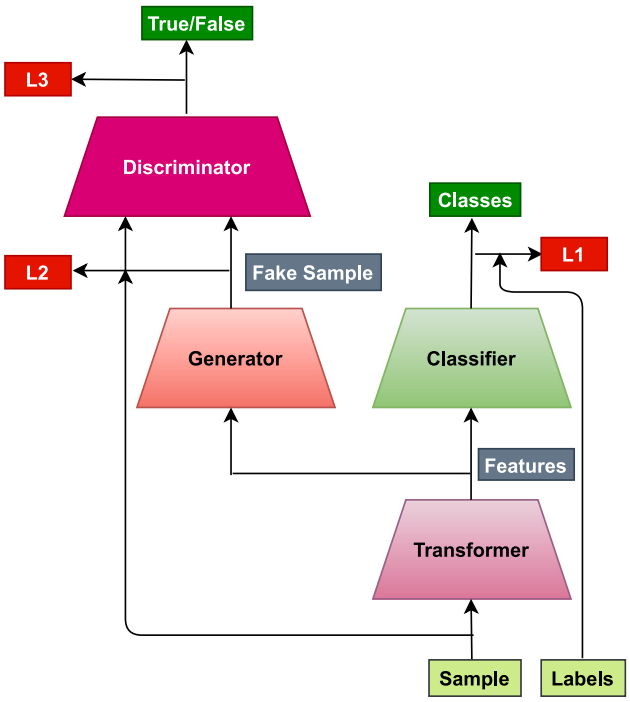


**Fig. 6.** Representation of the input, output, and loss functions in the proposed method.

vectors $v_1, \ldots, v_n$, the computation of an MLP sublayer for any $v_i$ is defined as follows:

$$MLP(v_i) = GELU(v_i \times W_1 + b_1) \times W_2 + b_2, \tag{7}$$

where $W_1$, $W_2$, $b_1$, and $b_2$ are parameters that refer to the two fully connected layers.

#### 2.4.1. Objective optimisation

Fig. 6 shows the loss functions that we have applied in the proposed framework. Our model has three tasks: classification, sample generation, and discrimination, where the main task of the model is the classification task. Other tasks are only used as auxiliary tasks due to not affecting the classification task performance by making the generator and discriminator loss function large. During the training process, the main task and auxiliary tasks are optimised at the same time. These branches have a sharing bottleneck layer, which is responsible for the best feature extraction in the network where, with this sharing, it can be achieved to a certain extent. We use this multi-task network model based on the joint loss function to adaptively share and transfer parameters between tasks during training.

Multiclass weighted cross-entropy loss [40] is applied as a loss function for multiclass classification. As we mentioned in the data processing section, we have an imbalanced dataset, and as a result, the trained model can be biased towards categories with a greater number of samples. To deal with this problem, the cross-entropy loss function is modified by assigning higher weights to categories with only a few samples and lower weights to categories with a greater number of samples. This can compensate for the misclassification of our data. The modified cross-entropy loss function can be calculated as follows:

$$L_1 = \frac{-1}{M} \sum_{k=1}^{K} \sum_{i=1}^{M} w_k y_i^k \log(y_i^k), \tag{8}$$

$$w_k = \lambda(1 - \frac{M_k}{M}), \lambda = 2.5, \tag{9}$$

where $w_k$ and $M_k$ are the weight and the number of samples for class k, respectively. We assign the highest value to classes 2 and 3 and the lowest to class 1, as class 1 has more samples.

For the generator, we applied the log-hyperbolic cosine (log-cosh) loss function [41,42] to calculate the loss between the generated samples and the actual samples. It includes the advantages of the L1 and L2 loss functions while mitigating their disadvantages [42]. Furthermore, this loss function has also been shown to improve the performance of the generator network in terms of output quality [42]. The generator loss function is as follows:

$$L_2 = \frac{1}{n} \sum_{i=1}^{n} log(cosh(\hat{y}_i - y_i)), \tag{10}$$

where $y_i$ is the actual value of the $i$th sample and $\hat{y}_i$ is the predicted value of the $i$th sample. For the generator–discriminator network's loss function, the following loss function was used [43,44]:

$$\begin{aligned} L_3 &= \min_G \max_D V(D,G) \\ &= \mathbb{E}_{x \sim p_{data}(x)}[\log D(x)] \\ &\quad + \mathbb{E}_{z \sim p_z(z)}[\log(1 - D(G(z)))]. \end{aligned} \tag{11}$$

We trained the discriminator (D) to maximise the probability of assigning the correct labels to both the training samples and the generated samples from the generator (G) [43]. At the same time, we trained the generator (G) to minimise $log(1 - D(G(z)))$ [43]. The discriminator network generates a scalar probability to determine whether a sample is real or fake [44]. Finally, we proposed a total loss function, which is a combination of all three of the above-mentioned loss functions, along with their assigned learnable weights:

$$loss = w_1 \times L_1 + w_2 \times L_2 + w_3 \times L_3. \tag{12}$$

## 3. Results

The proposed classifier was trained to classify the three classes in our dataset, which consists of one normal gait class and two unbalanced gait classes. We selected only 100 frames for each exercise in each session because of the noise caused by having 300 sequences in each session. We added zero-padding to the stored files for the exercises with less than 100 frames of key-point data to make the sequences the same length for all data. Among the 84 samples in our dataset, we selected 58 samples for the training dataset and 26 samples for the test dataset. In the training data, 34 samples are in class 1, 13 samples are in class 2, and 11 samples are in class 3, while in the test data, 13 samples are in class 1, 6 samples are in class 2, and 7 samples are in class 3. Due to our imbalanced data, the classes in the test dataset are not of the same size, as the samples were randomly selected so that the classes in the test set had the same proportions as the classes in the full dataset. We select more samples in class 1 for the test dataset to prevent the training dataset from becoming more imbalanced due to a gap between the classes' sizes. The five-fold cross-validation technique was used to validate unseen samples while preventing overfitting. In other words, the datasets were randomly shuffled and split into five folds of almost equal size. Each division was used as validation data, and the other four divisions were used as training data.

We use PyTorch [45], an open source library for deep learning, as a framework to implement the algorithms. This framework is based on the Torch library [45] and is written in Python. In this paper, the system configuration used for training included an Intel(R) Core i7-CPU @ 2.60 GHz with 16 GB of RAM and an NVIDIA GeForce GTX 1660 Ti. Furthermore, to optimise the models and accelerate the training process, we used the Adamw optimiser [46]. We set the random seed to 21 in training for a fixed initialisation of the network parameters. Finally, when the best test accuracy was achieved, the model that achieved that test accuracy was chosen; the best test accuracy was reached after 100 epochs with an optimised learning rate of $1e^{-3}$, and regularisation was added by adding a weight decay of $1e^{-5}$ to validate the performance of the proposed gait disorder classification network.

Part of our deep network framework is inspired by the encoder transformer classifier [35], so to demonstrate the improvement in the

**Table 4**
Comparison of training results (average 5 times).

| Method | Accuracy [%] | F-score [%] | Precision [%] | Recall [%] |
|---|---|---|---|---|
| Proposed model | 88.00 | 89.00 | 90.00 | 88.00 |
| Encoder transformer | 91.00 | 91.00 | 94.00 | 91.00 |

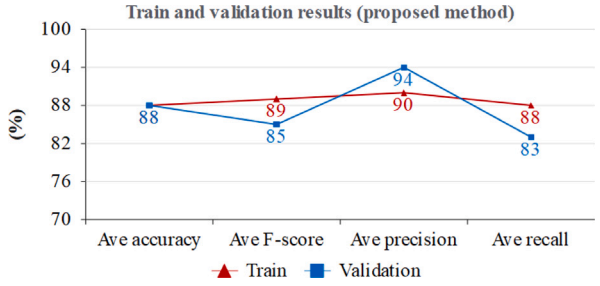


**Fig. 7.** Training and validation results of the proposed method.

**Table 5**
Comparison of validation results (average 5 times).

| Method | Accuracy [%] | F-score [%] | Precision [%] | Recall [%] |
|---|---|---|---|---|
| Proposed model | 88.00 | 85.00 | 94.00 | 83.00 |
| Encoder transformer | 77.00 | 75.00 | 78.00 | 74.00 |

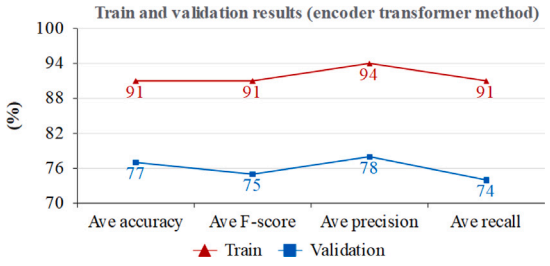


**Fig. 8.** Training and validation results of the encoder transformer method.

performance of the proposed method, we also compared it with the modified 1D encoder transformer classifier. Figs. 7 and 8 show the performance results of the proposed method and the encoder transformer on the training and validation datasets according to several metrics. Furthermore, the average precision, recall, F-score, and accuracy are compared for the training and validation processes, as described in Tables 4 and 5, respectively. As can be seen from the training results, both methods show good performance during training, but in the validation results, the highest scores are achieved with the proposed method, while the encoder transformer obtains accuracy and F-score values that are around 10% lower than those obtained by the proposed method.

## 4. Discussion

t-SNE [47] takes a high-dimensional dataset and reduces it to a low-dimensional dataset to find clusters in the data that keep the most useful of the initial information and retain the meaning in the data [47]. The upper subfigure of Fig. 9 shows the results of t-SNE applied to the test samples. The results show that the samples of each class overlap each other. The middle subfigure shows the results of t-SNE on the features extracted in the middle layer from the proposed method. We

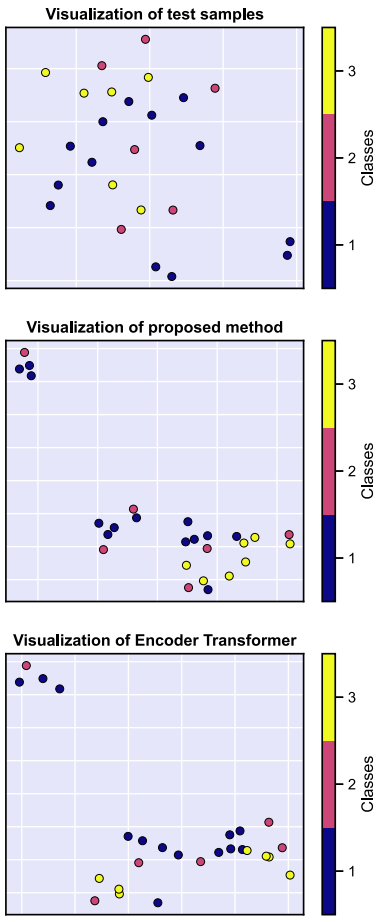


**Fig. 9.** t-SNNE results comparison.

can see significant clusters formed and some samples of different classes overlapping one another when they are more similar (classes 1 and 2, which are light and heavy disbalances), but class 3, without disbalance, has a more apparent distinction from samples from other classes. In the lower sub-figure, the encoder transformer results show significant improvement in clustering results compared to the original test samples but worse than the results from our proposed method.

As [47] mentioned, t-SNE can create inaccurate clusters in the low-dimensional representation of the data; we also applied the manifold discovery and analysis (MDA) introduced by [48], which is presented for deep learning feature visualisation. They claimed this method presents insightful visualisation of deep learning feature space in classification tasks.

Fig. 10 shows the results of MDA on the test samples, the features extracted in the mid-layer from the proposed method, and the encoder transformer for feature visualisation of the trained networks. We can observe that the two classes (class 1 and class 3, which are light and have no disbalance) have a more apparent distinction than the t-SNE result, as classes were better separated in the two dimensions. Moreover, MDA provides better-differentiated distribution than t-SNE between class 1 and class 2, which are light and heavy disbalances.

Furthermore, for more insightful visualisation of deep learning features, we apply MDA for three different layers, two from feature outputs of transformer heads (layer one and layer 2), and one from features output of bottleneck (layer 3) during five different epochs. As seen from Fig. 11, the data points are randomly distributed at the beginning of training, but by increasing epochs, the feature space becomes well clustered. In addition, the deeper layer (layer 3) shows better clustering

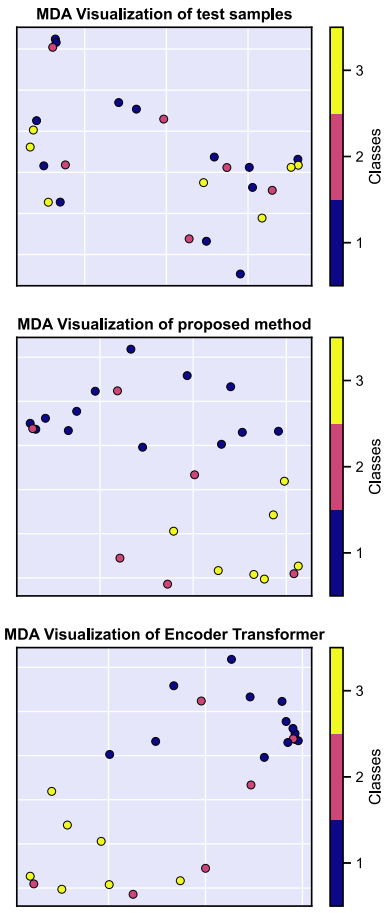


**Fig. 10.** MDA results comparison.

**Table 6**
Performance metrics of the proposed method for the test dataset.

| Metric | Class 1 [%] | Class 2 [%] | Class 3 [%] |
|---|---|---|---|
| Precision | 81.00 | 100.00 | 100.00 |
| Recall | 100.00 | 50.00 | 100.00 |
| F-score | 90.00 | 67.00 | 100.00 |

results and can extract higher-level features for better classification results.

It is evident from the performance metrics of the proposed method provided in Table 6 that it achieved an F-score of 100% for class 3, which is followed by class 1, with an F-score of 90%, and class 2, with an F-score of 67%. This means that the method performed well for class 1 and class 3, which contained samples with a light imbalance and no imbalance, but for class 2 (heavy imbalance), it had a poor classification performance. On the other hand, it has a precision of 100% for class 2, while the recall value for this class is 50%. The results show that the proposed model performed perfectly for class 1, with all metric values equal to 100%.

Table 7 shows that the encoder transformer achieved an F-score of 85% for class 1, which is followed by class 3, with an F-score of 83%, and class 2, with an F-score of 57%. This method also had a better precision for class 3, while the worst results were obtained for class 2. For class 1, all metrics were equal to 85%. For both classification methods, it is evident that class 2 is the most challenging class.

The confusion matrix obtained for the maximum success value of the proposed model is shown in Fig. 12. The maximum classification success achieved by the proposed model is around 88.46%. According

**Table 7**
Performance metrics of the encoder transformer method for the test dataset.

| Metric | Class 1 [%] | Class 2 [%] | Class 3 [%] |
|---|---|---|---|
| Precision | 85.00 | 50.00 | 100.00 |
| Recall | 85.00 | 67.00 | 71.00 |
| F-score | 85.00 | 57.00 | 83.00 |

**Table 8**
Comparison of the results of different groups.

| Group | Num. of Features | Accuracy [%] | F-score [%] | Precision [%] | Recall [%] |
|---|---|---|---|---|---|
| Group 1 | 34 | 62.00 | 46.00 | 41.00 | 54.00 |
| Group 2 | 36 | 58.00 | 54.00 | 56.00 | 53.00 |
| Group 3 | 58 | 81.00 | 74.00 | 87.00 | 73.00 |
| Group 4 | 62 | 85.00 | 81.00 | 83.00 | 81.00 |
| Group 5 | 62 | 69.00 | 64.00 | 66.00 | 68.00 |
| Group 6 | 62 | 81.00 | 78.00 | 83.00 | 76.00 |
| Proposed | 70 | 88.00 | 85.00 | 94.00 | 83.00 |

to the resulting confusion matrix, there was an error in classifying only one class. Three samples from class 2 (patients with a heavy imbalance) were classified as class 1 samples (patients with a light imbalance), which means that the proposed method incorrectly classified a few abnormal samples (heavy and light imbalance samples).

The maximum success value obtained for the encoder transformer is 76.92%, and incorrect results were obtained for all three classes. As can be seen in Fig. 13, two samples from class 2 were classified as class 1 samples and two samples from class 1 and class 3 were classified as class 2 samples. Thus, the encoder transformer had difficulty distinguishing between normal and abnormal cases.

The above results show that the proposed method can effectively classify gait disorders by using all the selected features to help the model distinguish between the different classes. We have also shown that our novel method significantly improved the performance of the encoder transformer classifier. In the following, the effect of the features on the classification results is analysed.

As mentioned in the data processing section, we generated 70 features using our dataset. To analyse the effect of each feature on the classification results, the selected features were divided into six subgroups: group 1 (34 features – only the $x$ and $y$ coordinate values of key points), group 2 (36 features – $x$ and $y$ coordinates are removed from the total features), group 3 (58 features – 12 vertical distance features are removed), group 4 (62 features – eight horizontal distance features are removed), group 5 (62 features – eight angles of symmetric key points are removed) and group 6 (62 features – eight angles between selected bones are removed). The performance results of these subgroups are shown in Table 8. As can be seen in Table 8, group 1, group 2, and group 5 have the worst results in terms of the accuracy and F-score, meaning that removing these features significantly worsened the classification results. In contrast, other groups have less of an impact on classification performance. The results show the importance of group 5, in which only the eight angles of symmetric key points were removed; removing these eight features from our data caused the classification accuracy and F-score to decrease by around 20%.

The number of model parameters, MACs, and FLOPs of the presented methods is shown in Fig. 14. The proposed framework has 161.72K parameters and 56.77M MACs and achieves an accuracy of 88.46%. On the contrary, the encoder transformer achieved an accuracy of 76.92% with 158.69K parameters and 35.32M MACs. The results show how much the classification results are improved when the proposed unsupervised methodology is used to modify the encoder transformer.

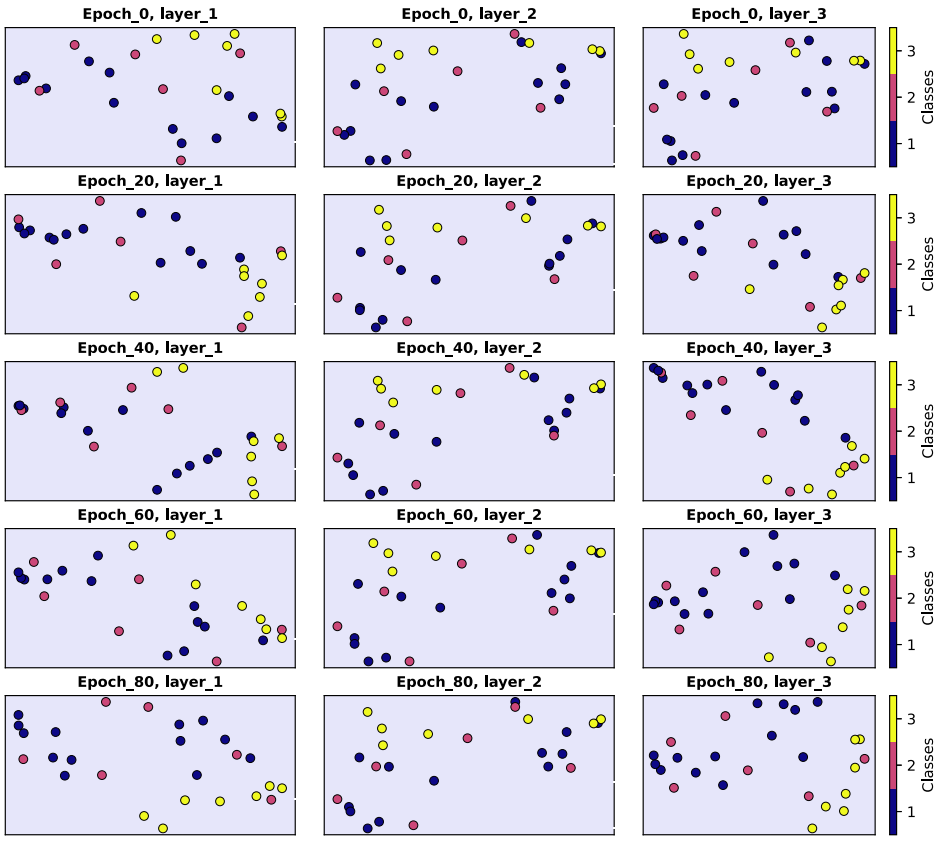


**Fig. 11.** MDA results with the progression of epochs.

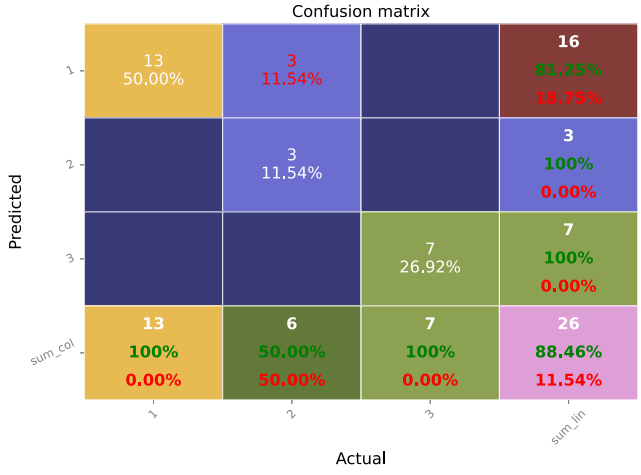


**Fig. 12.** Metrics of the efficiency of the proposed model.

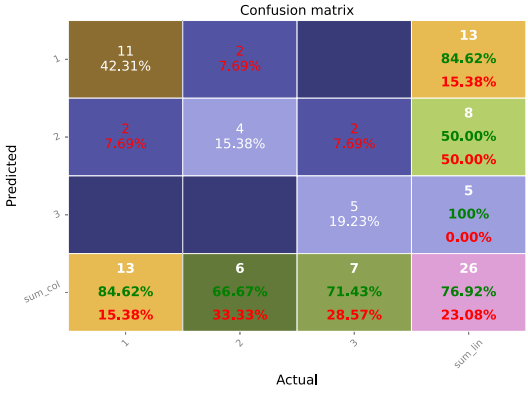


**Fig. 13.** Metrics of the efficiency of the encoder transformer.

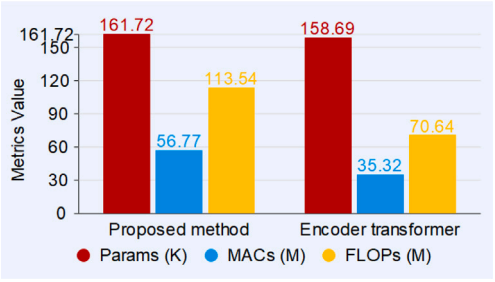


**Fig. 14.** Metrics of model efficiency.

### 4.1. Comparison

We compared the performance of the proposed method with different state-of-the-art methods. To benchmark, we apply the proposed model against state-of-the-art tabular data analysis methods such as genoMap [49] and deepInsight [50] input. In addition, we compare the performance of the model against the genoNet [49] with genoMap tabular input and pretrained ResNet152 [51] as a base network (and a header with a fully connected layer for classification) with deepInsight tabular input. For genoMap, the pixel size configured as 50 × 50 pixels was trained as the best in terms of memory size and training speed,

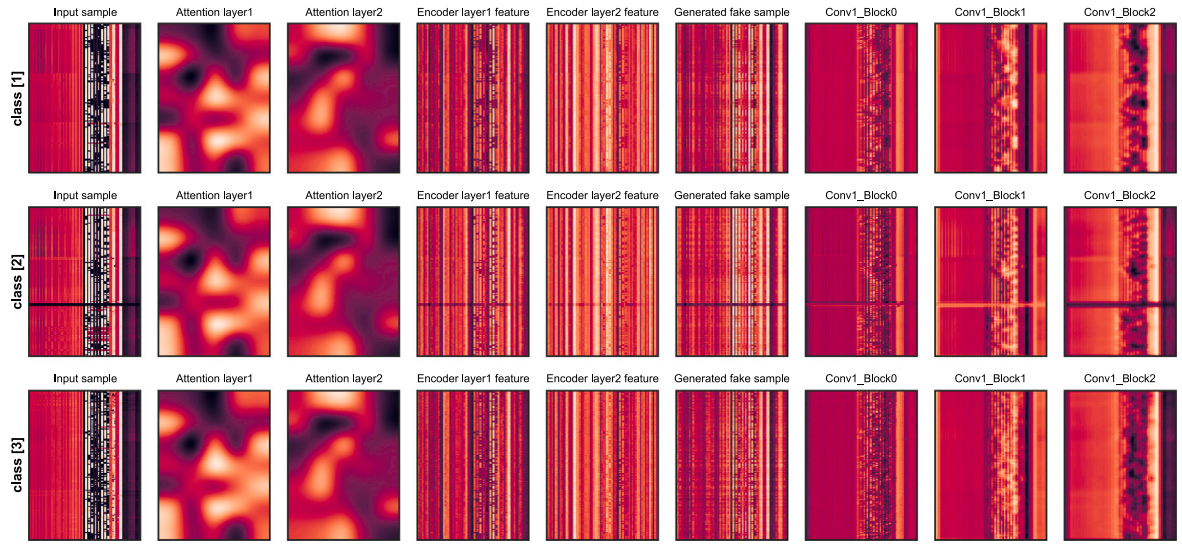


**Fig. 15.** Result of proposed method with proposed input.

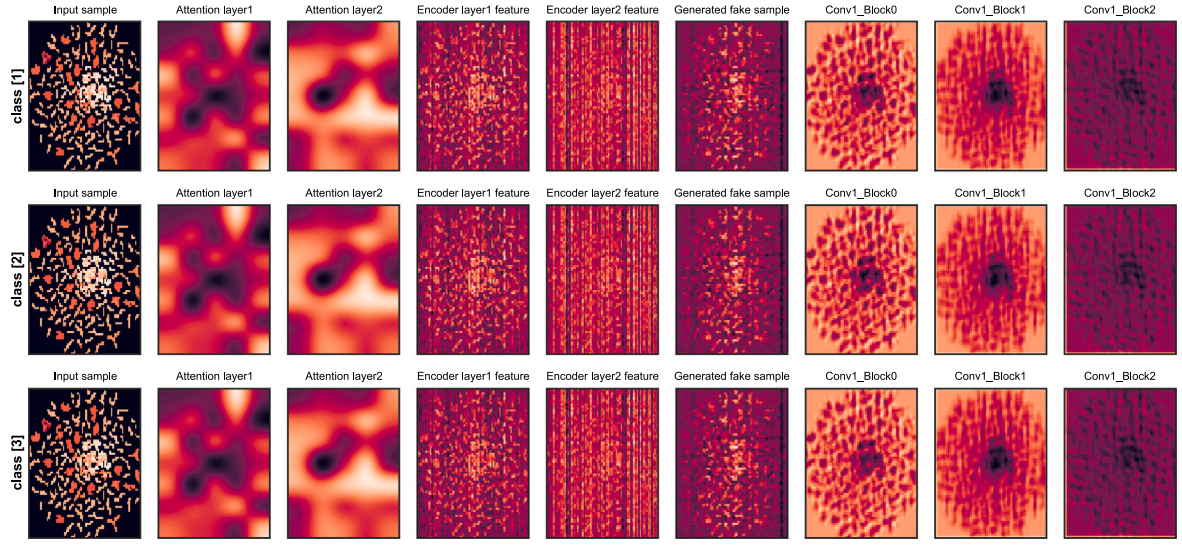


**Fig. 16.** Result of proposed method with deepInsight input.

while 38 × 38 pixels were trained with genoNet to achieve a better result compared to 50 × 50 pixels. For the deepInsight image, it has configured 224 × 224 and 70 × 70 pixels for ResNet152 and the proposed method, respectively.

Fig. 15 shows three samples of three classes from the proposed test dataset along with the results of the layers of the proposed method. As the image sample is made from features in our dataset, it looks complex to distinguish between them by glancing. However, from the last layer, it is clear that the results of the classes are different, and the results of classes 1 and 2 (who have gait disorders) are more similar to the results of class 3 (who have gait with no disbalance). We also can see from attention layers that the network concentrates on regions with the most representative class information. As all three classes are the features of patients' movement, generally, the attention map of different classes shows a similar map. The class' sample results show some attention on specific regions at layer 1, where it improved in the next attention layer. In other words, the first attention layer mainly focused on unrelated information, and then the next attention layer attempted to obtain the discriminative region. The last three layers in each row show the output of three different CNN layers. As shown in Fig. 15, it learns to identify the discriminative characteristics corresponding to each class when the CNN network becomes deeper.

Fig. 16 shows the above-mentioned samples generated with the deepInsight method along with the output of the layers of the proposed method. It also looks complex to distinguish between them in sample images, but from the interlayers; results are not different; they are more similar in each layer's results. We also can see from attention layers that the network concentrates on most regions of images that have unrelated information about the classes. The output of three different CNN layers also shows a similar pattern among different classes.

Fig. 17 shows the samples as mentioned above generated with the genoMap method along with the outputs of the layers of the proposed method. The samples generated by this method also look complex in distinguishing between them. However, from the last CNN layer, it is clear that the results of the classes are different, and the results of classes 1 and 3 are more similar than those of class 2. In other words, the method of these generated samples classifies class 1(who gait with light disbalance) as similar to class 3 (who gait with no disbalance). The output of class 2 (who gait with heavy disbalance) showed different patterns. The output of the sample shows some attention to specific regions in layer 1, where it progressed to the next attention layer.

Table 9 shows the result of the proposed network with the proposed test samples, genoMap test samples, and deepInsight test samples.

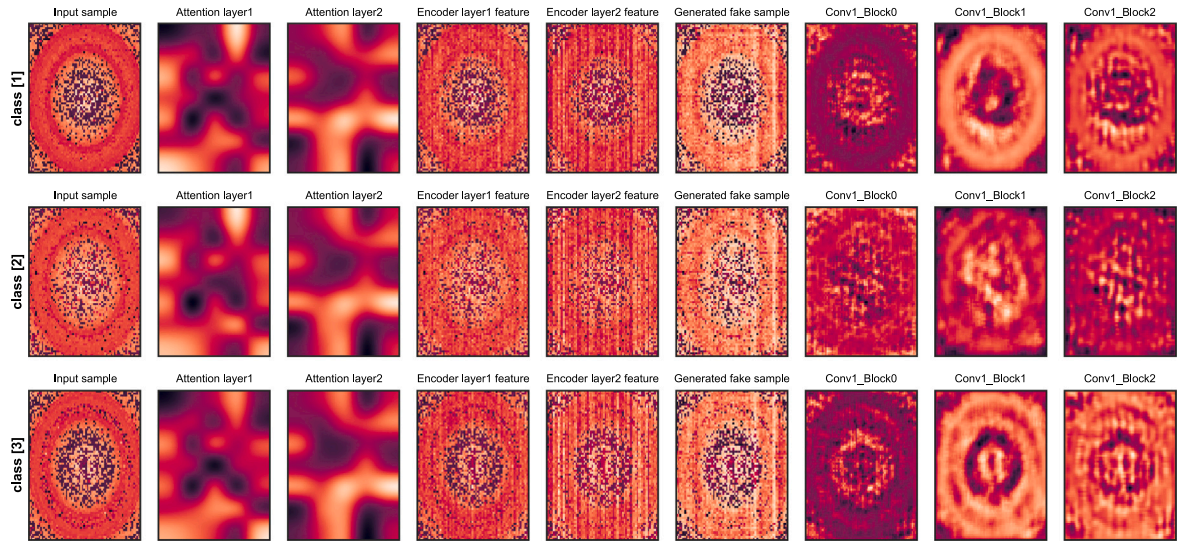


**Fig. 17.** Result of proposed method with genomap input.

Additionally, the results of genoMap by genoNet and deepInsight sample by ResNet152 were also evaluated for better conclusions of the results of these two state-of-the-art methods. As can be seen from the table mentioned, the maximum Fscore value achieved by the proposed method and the proposed test sample, where its accuracy and Fscore are around 88% and 85%, respectively. The highest precision and recall values are also among the proposed method and the proposed test sample. Second-best metrics were achieved with the proposed method and the genoMap test sample with 65%, 55%, 76%, and 54%, respectively. The genoMap sample with genoNet shows better metrics than the deepInsight sample with the proposed method or ResNet152. The related confusion matrix of the above-aforementioned methods is shown in Table 10. The result shows that the proposed method with the proposed test sample correctly predicts class 1 with 13 correct predictions from 13 samples, three correct predictions for class 2 among 6, and a 100% correct prediction for class 3. Overall, the proposed method was effective among class 1 and class 3, among others. It only shows poor performance for class 2, with three mistakes among six samples. It is misclassified as class 1, where both class 1 and class 2 are among imbalanced gait samples. In other words, the proposed method effectively distinguished between normal and imbalanced gait samples: the metrics results of the genoMap sample drop significantly when we apply genoNet instead of the proposed method. The second-best result is from the genoMap sample with the proposed method, which also correctly predicts class 1 with 13 correct predictions from 13 samples, two correct predictions for class 2 among 6, and two correct predictions for class 3 among 7. The incorrect results had mostly been misclassified with class 1. The deepInsight sample with ResNet152 results is the worst case, with zero correct predictions in classes 2 and 3. In other words, it shows the highest misclassification results where all classes are predicted as class 1. From the results, it is reasonable to conclude that the features generated from the joints play an essential role in classifying gait disorders in our assessment. As we see from the result, state-of-the-art methods to generate images from features significantly reduced the classification results, either with the proposed method or state-of-the-art deep network methods. In other words, the analysis of gait disorders must consider the significant temporal patterns between subjects walking, which influenced the classification of gait disorders.

**Table 9**
Metrics comparison.

| Methods | Accuracy | *F*-score | Precision | Recall |
|---|---|---|---|---|
| proposed sample with proposed method | 0.88 | 0.85 | 0.94 | 0.83 |
| genoMap sample with proposed method | 0.65 | 0.55 | 0.76 | 0.54 |
| deepInsight sample with proposed method | 0.38 | 0.24 | 0.21 | 0.29 |
| genoMap sample with genoNet | 0.54 | 0.32 | 0.51 | 0.39 |
| deepInsight sample with ResNet152 | 0.50 | 0.22 | 0.17 | 0.33 |

**Table 10**
Comparison of confusion matrix.

| Methods | | | | |
|---|---|---|---|---|
| | | **1** | **2** | **3** |
| proposed sample with proposed method | **1** | 13 | 3 | 0 |
| | **2** | 3 | 2 | 1 |
| | **3** | 5 | 0 | 2 |
| | | **1** | **2** | **3** |
| genoMap sample with proposed method | **1** | 13 | 0 | 0 |
| | **2** | 3 | 2 | 1 |
| | **3** | 5 | 0 | 2 |
| | | **1** | **2** | **3** |
| deepInsight sample with proposed method | **1** | 9 | 5 | 6 |
| | **2** | 4 | 1 | 1 |
| | **3** | 0 | 0 | 0 |
| | | **1** | **2** | **3** |
| genoMap sample with genoNet | **1** | 13 | 5 | 7 |
| | **2** | 0 | 1 | 0 |
| | **3** | 0 | 0 | 0 |
| | | **1** | **2** | **3** |
| deepInsight sample with ResNet152 | **1** | 13 | 6 | 7 |
| | **2** | 0 | 0 | 0 |
| | **3** | 0 | 0 | 0 |

## 5. Conclusions

The proposed framework modified the encoder transformer classifier to classify the gait dysfunction of sequences based on generated and selected features from skeletal key points. Due to our small and imbalanced dataset, we applied the data augmentation technique and used optimal weights for each class in the loss function in the training stage. In our data processing method, we generated practical features to improve the performance of the proposed method. Through this feature generation and by designing a novel methodology, we achieved an accuracy of 88%, which is greater than the accuracy of the encoder transformer classifier by about 10%. Our method performs well in distinguishing between balanced and imbalanced movements. The results have shown that the proposed method performs well in the challenging case of heavy imbalanced movement (class 2). This model can be used in a classification system to help clinical physicians make faster and better decisions. It could also be helpful for patients; for example, it

could recommend that they visit a doctor if the application recognises gait dysfunction in their activities.

The following are some possible limitations. In the proposed project, we applied some feature creation from the concept that clinicians get from patient performance during the defined exercise. Due to this, this feature generation is specific to this problem, and it might only work in similar problems. The model is helpful for time series problems needing longer temporal or long-term dependency identification. The proposed model includes supervised and unsupervised methods, where the unsupervised technique learns from the data rather than labels. Hence, it needs considerable clean data to generate valuable results. If the dataset is too noisy or not large enough, it may affect the generated results. In addition, the generated samples are not perfect, and it is hard to decide how much loss is tolerable in the generated output, and it needs a process of trial and error. In transformer architecture, the attention structures can help detect the sample's most significant parts. However, clarifying how the model makes its classification decisions can be challenging. Transformers are subject to focus on frequent patterns rather than on proper understanding and may have difficulties in processing rare events or long-term trends in data. In addition, they depend on a fixed-size input, and it can be challenging to handle variable-length inputs efficiently.

### 5.1. Future work

We have achieved promising results; however, due to our small and imbalanced dataset, there are several gaps and limitations in this study that can be improved. We will also accumulate more data with a new application and in different environments by partnering with rehabilitation hospitals. In addition, we intend to add more practical features and analyse them in detail from a physician's perspective. We are also interested in designing a novel method based on a graph network to study the performance of this network for gait disorder analysis. Another future research direction worth investigating is analysing 3D data and comparing models trained on 3D data with those trained on 2D data.

## 6. Abbreviations

The following abbreviations are used in this manuscript:

| | |
|---|---|
| Adam | Adaptive Moment Estimation with Decoupled Weight Decay |
| BiLSTM | Bidirectional Long Short-Term Memory |
| CT | Computed Tomography |
| CNNs | Convolutional Neural Networks |
| DNNs | Deep Neural Networks |
| ET-GD | Encoder Transformer Integrated with Generator-Discriminator |
| FLOPs | Floating-Point Operations |
| GANs | Generative Adversarial Networks |
| GCNN | Graph Convolutional Neural Network |
| HAR | Human Activity Recognition |
| IT | Information Technology |
| KNN | k-Nearest Neighbour |
| LSTM | Long Short-Term Memory |
| MACs | Multiply-Accumulate Operations |
| MLP | Multilayer Perceptron |
| NMR | Nuclear Magnetic Resonance |
| PCA | Principal Component Analysis |
| RCNN | Region-based Convolutional Neural Networks |
| SVM | Support Vector Machine |
| GELU | Gaussian Error Linear Unit |
| ResNet | Residual Network |
| RNNs | Recurrent Neural Networks |
| ViT | Vision Transformer |

## CRediT authorship contribution statement

**Mohsen Shayestegan:** Writing – review & editing, Writing – original draft, Validation, Software, Methodology, Investigation. **Jan Kohout:** Writing – review & editing, Writing – original draft, Validation, Software, Methodology. **Kateřina Trnková:** Writing – review & editing, Writing – original draft, Methodology, Investigation, Formal analysis. **Martin Chovanec:** Writing – review & editing, Writing – original draft, Methodology, Investigation, Funding acquisition, Formal analysis. **Jan Mareš:** Writing – review & editing, Writing – original draft, Validation, Supervision, Software, Methodology, Investigation, Funding acquisition.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Acknowledgements

The work of J.K. and J.M. was funded by the Ministry of Education, Youth, and Sports through the grant 'Development of Advanced Computational Algorithms for Evaluating Post-Surgery Rehabilitation' (number LTAIN19007). The work of M.Ch. and K.T. was supported by the research project of Charles University Cooperation - Surgical Disciplines, 3rd Faculty of Medicine. This support is gratefully acknowledged.

### Declarations

This research was approved by the Ethics Committee of the University Hospital Královské Vinohrady Prague (EK-VP/4310120), where the measurements were made. Each patient signed an informed consent form that described the research conditions.

## References

[1] Y. Li, X. He, M. Alsinan, H. Kwak, H. Hoteit, Rapid NMR T2 Extraction from Micro-CT Images Using Machine Learning, in: Abu Dhabi International Petroleum Exhibition and Conference, vol. Day 2 Tue, November 01, 2022, 2022, http://dx.doi.org/10.2118/211095-MS, D022S158R002. arXiv:https://onepetro.org/SPEADIP/proceedings-pdf/22ADIP/2-22ADIP/D022S158R002/3040236/spe-211095-ms.pdf.

[2] A.H. Routt, N. Yang, N.Z. Piety, M. Lu, S.S. Shevkoplyas, Deep ensemble learning enables highly accurate classification of stored red blood cell morphology, Sci. Rep. 13 (1) (2023).

[3] M.A. Patel, M.J. Knauer, M. Nicholson, M. Daley, L.R. Van Nynatten, G. Cepinskas, D.D. Fraser, Organ and cell-specific biomarkers of long-COVID identified with targeted proteomics and machine learning, Mol. Med. 29 (1) (2023).

[4] A. Bawa, K. Banitsas, M. Abbod, A review on the use of microsoft kinect for gait abnormality and postural disorder assessment, J. Healthc. Eng. 2021 (2021).

[5] A. Naji Hussain, S.A. Abboud, B.A.B. Jumaa, M.N. Abdullah, Impact of feature reduction techniques on classification accuracy of machine learning techniques in leg rehabilitation, Measurement: Sensors 25 (2023).

[6] M. Shayestegan, J. Kohout, K. Štícha, J. Mareš, Advanced analysis of 3D kinect data: Supervised classification of facial nerve function via parallel convolutional neural networks, Appl. Sci. (Switzerland) 12 (2022) http://dx.doi.org/10.3390/app12125902.

[7] Ö.F. İnce, I.F. Ince, M.E. Yıldırım, J.S. Park, J.K. Song, B.W. Yoon, Human activity recognition with analysis of angles between skeletal joints using a RGB-depth sensor, ETRI J. 42 (2020) 78–89, http://dx.doi.org/10.4218/etrij.2018-0577.

[8] B. Açış, S. Güney, Classification of human movements by using Kinect sensor, Biomed. Signal Process. Control 81 (2023) http://dx.doi.org/10.1016/j.bspc.2022.104417.

[9] Y. Guo, F. Deligianni, X. Gu, G.-Z. Yang, 3D canonical pose estimation and abnormal gait recognition with a single RGB-D camera, IEEE Robot. Autom. Lett. 4 (2019) 3617–3624, URL http://eprints.gla.ac.uk/208062/.

[10] K. Jun, D.W. Lee, K. Lee, S. Lee, M.S. Kim, Feature extraction using an RNN autoencoder for skeleton-based abnormal gait recognition, IEEE Access 8 (2020) 19196–19207, http://dx.doi.org/10.1109/ACCESS.2020.2967845.

[11] M.N.I. Shuzan, M.E. Chowdhury, M.B.I. Reaz, A. Khandakar, F.F. Abir, M.A.A. Faisal, S.H.M. Ali, A.A.A. Bakar, M.H. Chowdhury, Z.B. Mahbub, M.M. Uddin, M. Alhatou, Machine learning-based classification of healthy and impaired gaits using 3D-GRF signals, Biomed. Signal Process. Control 81 (2023) http://dx.doi.org/10.1016/j.bspc.2022.104448.

[12] X. Gu, Y. Guo, F. Deligianni, B. Lo, G.-Z. Yang, Cross-subject and cross-modal transfer for generalized abnormal gait pattern recognition, IEEE Trans. Neural Netw. Learn. Syst. (2020) URL https://guxiao0822.github.io/GAGR.

[13] M. Woollacott, A. Shumway-Cook, Attention and the control of posture and gait: A review of an emerging area of research, Gait Posture 16 (1) (2002) 1–14, Cited By :1797.

[14] L. di Biase, L. Raiano, M.L. Caminiti, P.M. Pecoraro, V. Di Lazzaro, Parkinson's disease wearable gait analysis: Kinematic and dynamic markers for diagnosis, Sensors 22 (22) (2022) Cited By :1.

[15] T.M. Steffen, T.A. Hacker, L. Mollinger, Age- and gender-related test performance in community-dwelling elderly people: Six-Minute Walk Test, Berg Balance Scale, Timed Up & Go Test, and gait speeds, Phys. Ther. 82 (2) (2002) 128–137, Cited By :1426.

[16] M.K. Graham, J.P. Staab, C.M. Lohse, D.L. McCaslin, A comparison of dizziness handicap inventory scores by categories of vestibular diagnoses, Otol. Neurotol. 42 (1) (2021) 129–136, Cited By :11.

[17] I. Tien, S.D. Glaser, M.J. Aminoff, Characterization of gait abnormalities in Parkinson's disease using a wireless inertial sensor system, in: 2010 Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBC'10, 2010, pp. 3353–3356.

[18] K.M. Monica, R. Parvathi, Efficient gait analysis using deep learning techniques, Comput. Mater. Contin. 74 (3) (2023) 6229–6249.

[19] P. Jatesiktat, G.M. Lim, C.W.K. Kuah, D. Anopas, W.T. Ang, Autonomous modeling of repetitive movement for rehabilitation exercise monitoring, BMC Med. Inf. Decis. Mak. 22 (1) (2022).

[20] Kolarova, Randomized controlled trial of robot-assisted gait training versus therapist-assisted treadmill gait training as add-on therapy in early subacute stroke patients: The GAITFAST study protocol, Brain Sci. 12 (12) (2022).

[21] Z. Dai, Z. Yang, Y. Yang, J. Carbonell, Q.V. Le, R. Salakhutdinov, Transformer-xl: Attentive language models beyond a fixed-length context, 2019, arXiv preprint arXiv:1901.02860.

[22] Y. Tay, D. Bahri, D. Metzler, D.-C. Juan, Z. Zhao, C. Zheng, Synthesizer: Rethinking self-attention for transformer models, in: International Conference on Machine Learning, PMLR, 2021, pp. 10183–10192.

[23] S. Vandenhende, B. De Brabandere, D. Neven, L. Van Gool, A three-player GAN: generating hard samples to improve classification networks, in: 2019 16th International Conference on Machine Vision Applications, MVA, IEEE, 2019, pp. 1–6.

[24] J.T. Springenberg, Unsupervised and semi-supervised learning with categorical generative adversarial networks, 2015, arXiv preprint arXiv:1511.06390.

[25] C. Li, T. Xu, J. Zhu, B. Zhang, Triple generative adversarial nets, Adv. Neural Inf. Process. Syst. 30 (2017).

[26] F. Auger, M. Hilairet, J.M. Guerrero, E. Monmasson, T. Orlowska-Kowalska, S. Katsura, Industrial applications of the Kalman filter: A review, IEEE Trans. Ind. Electron. 60 (2013) 5458–5471, http://dx.doi.org/10.1109/TIE.2012.2236994.

[27] V. Dentamaro, D. Impedovo, G. Pirlo, Gait analysis for early neurodegenerative diseases classification through the kinematic theory of rapid human movements, IEEE Access 8 (2020) 193966–193980, http://dx.doi.org/10.1109/ACCESS.2020.3032202.

[28] G.V. Veres, L. Gordon, J.N. Carter, M.S. Nixon, What image information is important in silhouette-based gait recognition? in: Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004, Vol. 2, CVPR 2004, IEEE, 2004, p. II.

[29] I. Birch, M. Birch, N. Asgeirsdottir, The identification of individuals by observational gait analysis using closed circuit television footage: Comparing the ability and confidence of experienced and non-experienced analysts, Sci. Justice 60 (1) (2020) 79–85.

[30] J.W. Johnson, Adapting mask-rcnn for automatic nucleus segmentation, 2018, arXiv preprint arXiv:1805.00500.

[31] K. He, G. Gkioxari, P. Dollár, R. Girshick, Mask r-cnn, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 2961–2969.

[32] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C.L. Zitnick, Microsoft coco: Common objects in context, in: Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13, Springer, 2014, pp. 740–755.

[33] M. Shayestegan, T. Zalabsky, J. Mareš, Triple parallel LSTM networks for classifying the gait disorders using kinect camera and robot platform during the clinical examination, in: 2023 3rd International Conference on Electrical, Computer, Communications and Mechatronics Engineering, ICECCME, IEEE, 2023, pp. 1–6.

[34] M. Shayestegan, J. Kohout, K. Trnková, M. Chovanec, J. Mareš, Motion tracking in diagnosis: Gait disorders classification with a dual-head attentional transformer-LSTM, Int. J. Comput. Intell. Syst. 16 (1) (2023) 98.

[35] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al., An image is worth 16x16 words: Transformers for image recognition at scale, 2020, arXiv preprint arXiv:2010.11929.

[36] S. Yu, H. Chen, E.B. Garcia Reyes, N. Poh, Gaitgan: Invariant gait feature extraction using generative adversarial networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2017, pp. 30–37.

[37] J.L. Ba, J.R. Kiros, G.E. Hinton, Layer normalization, 2016, arXiv preprint arXiv:1607.06450.

[38] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.

[39] R. Xiong, Y. Yang, D. He, K. Zheng, S. Zheng, C. Xing, H. Zhang, Y. Lan, L. Wang, T. Liu, On layer normalization in the transformer architecture, in: International Conference on Machine Learning, PMLR, 2020, pp. 10524–10533.

[40] Z. Zhang, M. Sabuncu, Generalized cross entropy loss for training deep neural networks with noisy labels, Adv. Neural Inf. Process. Syst. 31 (2018).

[41] R.A. Saleh, A. Saleh, et al., Statistical properties of the log-cosh loss function used in machine learning, 2022, arXiv preprint arXiv:2208.04564.

[42] P. Chen, G. Chen, S. Zhang, Log hyperbolic cosine loss improves variational auto-encoder, 2018, openreview.net.

[43] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial networks, Commun. ACM 63 (11) (2020) 139–144.

[44] D. Yoo, N. Kim, S. Park, A.S. Paek, I.S. Kweon, Pixel-level domain transfer, in: Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, the Netherlands, October 11-14, 2016, Proceedings, Part VIII 14, Springer, 2016, pp. 517–532.

[45] A. Paszke, S. Gross, F. Massa, A. Lerer, J.P. Bradbury, An imperative style, high-performance deep learning library, Adv. Neural Inf. Process. Syst. 32 (2019).

[46] I. Loshchilov, F. Hutter, Decoupled weight decay regularization, 2017, arXiv preprint arXiv:1711.05101.

[47] M.T. Islam, L. Xing, A data-driven dimensionality-reduction algorithm for the exploration of patterns in biomedical data, Nat. Biomed. Eng. 5 (6) (2021) 624–635.

[48] M.T. Islam, Z. Zhou, H. Ren, M.B. Khuzani, D. Kapp, J. Zou, L. Tian, J.C. Liao, L. Xing, Revealing hidden patterns in deep neural network feature space continuum via manifold learning, Nature Commun. 14 (1) (2023) 8506.

[49] M.T. Islam, L. Xing, Cartography of genomic interactions enables deep analysis of single-cell expression data, Nature Commun. 14 (1) (2023) 679.

[50] A. Sharma, E. Vans, D. Shigemizu, K.A. Boroevich, T. Tsunoda, DeepInsight: A methodology to transform a non-image data to an image for convolution neural network architecture, Sci. Rep. 9 (1) (2019) 11399.

[51] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.