AIMS Bioengineering, 12(2): 283–313.
DOI:10.3934/bioeng.2025014
Received: 14 March 2025
Revised: 15 May 2025
Accepted: 28 May 2025
Published: 26 June 2025

Bioengineering

https://www.aimspress.com/journal/Bioengineering

Research article

Cardiomyopathy diagnosis model from endomyocardial biopsy specimens: Appropriate feature space and class boundary in small sample size data

Masaya Mori[1], Yuto Omae[1], Yutaka Koyama[2], Kazuyuki Hara[1], Jun Toyotani[1], Yasuo Okumura[3], and Hiroyuki Hao[2,*]

[1] College of Industrial Technology, Nihon University, 1-2-1 Izumi, Narashino, Chiba 275-8575, Japan

[2] Division of Human Pathology, Department of Pathology and Microbiology, Nihon University School of Medicine, 30-1 Oyaguchi Kamicho, Itabashi, Tokyo 173-8610, Japan

[3] Division of Cardiology, Department of Medicine, Nihon University School of Medicine, 30-1 Oyaguchi Kamicho, Itabashi, Tokyo 173-8610, Japan

* **Correspondence:** Email: hao.hiroyuki@nihon-u.ac.jp.

Abstract: As the number of patients with heart failure increases, machine learning (ML) has garnered attention in cardiomyopathy diagnosis, driven by the shortage of pathologists. However, endomyocardial biopsy specimens are often limited in sample size and require techniques such as feature extraction and dimensionality reduction. This study investigated the effectiveness of texture features in the context of feature extraction for the pathological diagnosis of cardiomyopathy. Furthermore, model designs that contributed to improving generalization performance were examined by applying feature selection (FS) and dimensional compression (DC) to several ML models. The obtained results were verified by visualizing the inter-class distribution differences and conducting statistical hypothesis testing based on texture features. Additionally, they were evaluated using predictive performance across different model designs with varying combinations of FS and DC (applied or not) and decision boundaries. The obtained results confirmed that texture features may be effective for the pathological diagnosis of cardiomyopathy. Moreover, when the ratio of features to the sample size is high, a multi-step process involving FS and DC improved the generalization performance, with the linear kernel support vector machine achieving the best results. This process was demonstrated to be potentially effective for models with reduced complexity, regardless of whether the decision boundaries were linear, curved, perpendicular, or parallel to the axes. These findings are expected to facilitate the development of an effective cardiomyopathy diagnostic model for its rapid adoption in medical practice.

Keywords: cardiomyopathy; endomyocardial biopsy; pathology image analysis; machine learning;

texture analysis; dimensionality reduction; low sample size

1. Introduction

1.1. Background

With the aging of society, heart failure is increasing worldwide. Various physiological and imaging examinations, such as echocardiography, cardiac computed tomography, and cardiac magnetic resonance imaging are used to determine the causes of heart failure. An endomyocardial biopsy is the only in vivo method for obtaining histopathological information about the myocardium [1]. It is useful for discriminating between primary cardiomyopathies, such as hypertrophic, dilated, restrictive, and arrhythmogenic right ventricular, as well as secondary cardiomyopathies, such as cardiac amyloidosis, cardiac sarcoidosis, lymphocytic myocarditis, giant cell myocarditis, and Fabry disease [2, 3]. However, a worldwide shortage of pathologists who can make such a diagnosis has emerged. Machine learning (ML) has recently been used for pathological diagnoses [4, 5]. If ML could be used to enumerate differential diseases based on image analysis and automatically calculate the likelihood of each disease, it would be useful for pathological diagnosis. It may also prevent misdiagnosis due to limited access to expert pathologists. Realizing this objective requires the development of an ML model capable of making pathological diagnoses based on histopathological information (myocardial cell diameter, myocardial cell morphology, nuclear diameter, nuclear morphology, presence and frequency of cellular infiltration, types of infiltrating cells, myocardial cell arrangement, fibrosis, and definitive diagnosis) obtained from myocardial biopsy or autopsy specimens of patients with a history of cardiac disease.

1.2. Comparison of deep and non-deep learning methods

Convolutional neural networks (CNNs) are widely used to predict disease states using medical images [6, 7]. One reason for this is that, in contrast to traditional ML methods, CNNs automate three processes that would typically require manual intervention (Figure 1: ML (non-deep learning (DL)) model process) [8]: (1) feature extraction, which quantifies the visual information inherent in the input image (such as color, brightness, patterns, textures, shapes, and object scale), (2) dimensionality reduction, which aims to improve analysis efficiency and prevent overfitting by selecting useful features for predicting and eliminating irrelevant or noisy features (this involves feature selection (FS) to select features deemed useful for recognizing the target, and dimensional compression (DC) to transform high-dimensional feature spaces into lower-dimensional ones), and (3) model building, which maximizes both the goodness of fit on training data and the generalization performance on unseen data (both performances are collectively referred to as predictive performance in this study). CNNs mainly consist of convolutional, pooling, fully connected, and output layers (Figure 1: CNNs model process). The convolutional and pooling layers extract visual features useful for recognition from the input image, which are then transformed into a low-dimensional feature vector by the fully connected layer [9,10]. This vector is fed into the output layer to yield a continuous value for regression or class probabilities for classification. Consequently, CNNs eliminate the need to design handcrafted features based on deep domain expertise, which is typically required in non-DL methods, as well as

dimensionality reduction of the feature space. Moreover, CNNs have the potential to extract features that may not be visible to humans, with diagnostic performance potentially comparable to that of expert pathologists [11, 12].

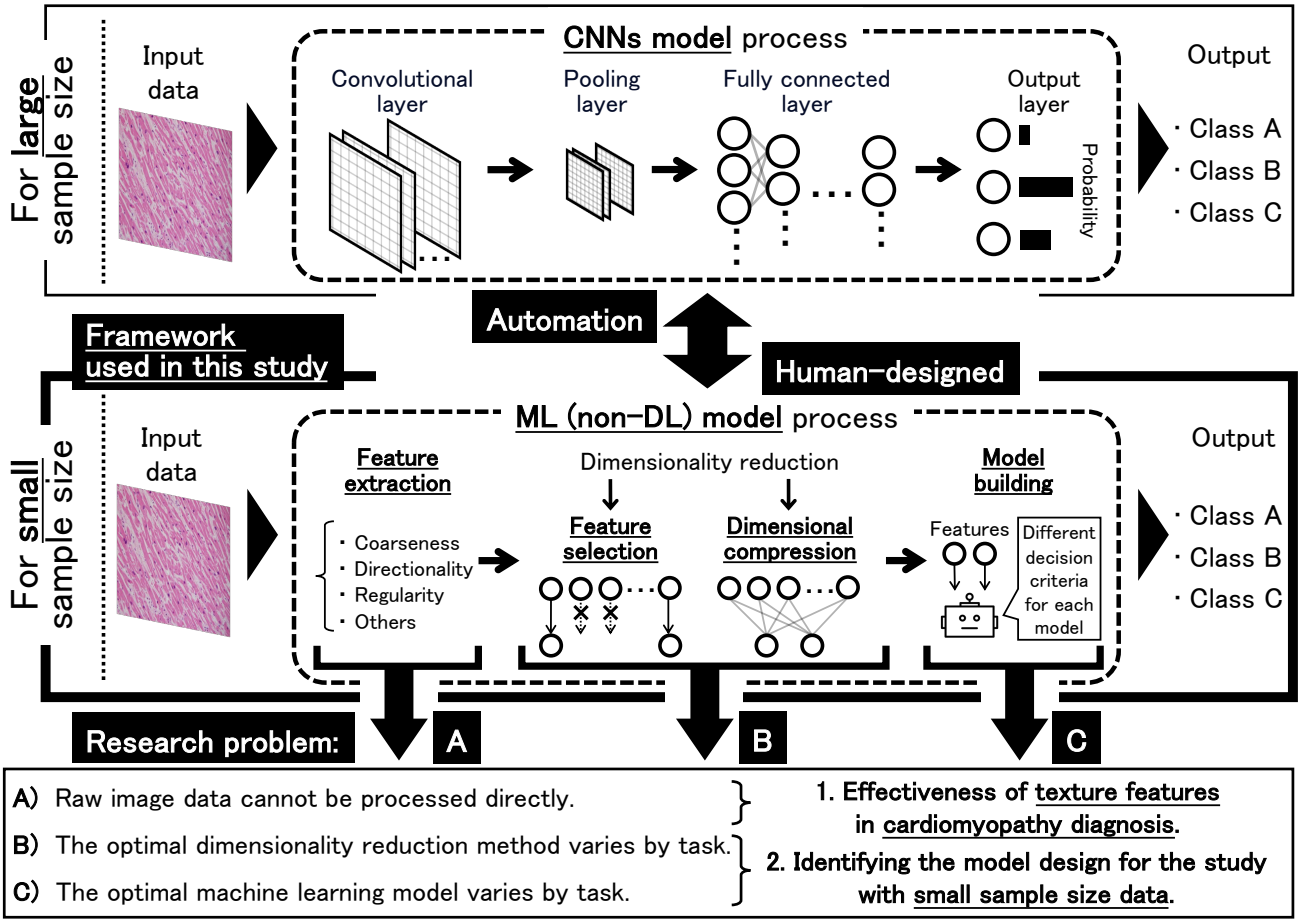


Figure 1. Visual summary of the Introduction.

However, to achieve a high predictive performance with CNNs, a vast amount of image data is required to sufficiently capture the diversity of the training data. Therefore, CNNs applied in the medical field often target X-ray, CT, and MRI images [13–16]. This is because these tests are non-invasive, are widely used in clinical settings, and allow for easier collection of high-quality images. In contrast, endomyocardial biopsy is a highly invasive procedure with associated risks, which necessitates careful examination [17]. Consequently, the frequency of these tests is relatively low, making it difficult to collect a sufficiently diverse set of histopathological images of the myocardium [18, 19]. Histopathological images of the myocardium often include small sample sizes with insufficient diversity. When constructing predictive models using small sample size data, it has been reported that the risk of overfitting increases when the ratio of features to the sample size or the complexity of ML models is high [20]. Here, model complexity refers to the extent to which a model can capture data patterns, which is determined by the structure of the algorithm and the number and values of the hyperparameters. Therefore, it is necessary to reduce the dimensionality of the dataset prior to training or adopt ML models with lower complexity. In conclusion, for the pathological

diagnosis of cardiomyopathy using small sample histopathological images of the myocardium, ML models with lower complexity that handle relatively low-dimensional data, such as non-DL models, are considered more suitable than complex CNNs that require high-dimensional input data.

Moreover, from the perspective of interpretability of the diagnostic rationale, non-DL ML models are considered more suitable. In deep image recognition models based on convolutions, methods that apply class activation mapping (CAM) are widely used to visualize the rationale behind predictions [21–23]. CAM visually emphasizes the areas within the input image that contribute significantly to the prediction results [24]. One advantage of this approach is that the reasoning behind the model can be confirmed visually, allowing even those unfamiliar with ML to understand it intuitively. This makes it easy to compare the reasoning of the model with the judgment rationale of experts in the relevant field. However, as CAM provides qualitative reasoning, it is challenging to identify the specific conditions that contribute to the prediction in non-structured images, which lack distinct shapes [25]. Therefore, in the case of cardiomyopathy pathology images, it is difficult to interpret the specific reasons for the predictions from the CAM, such as color, surface roughness, regularity, and directionality. In such cases, the application of statistical methods, as used in non-DL models for feature extraction and selection, is considered effective for interpreting the specific grounds for predictions.

1.3. Research problems

To achieve a pathological diagnosis of cardiomyopathy using a non-DL ML model, it is necessary to consider (1) the extraction of image features from histopathological images of the myocardium and (2) model design, which is robust to small sample sizes.

First, the extraction of image features from cardiomyopathy pathology images is described (Figure 1: Research problem A). Raw images contain a wide variety of visual information, such as color, shape, patterns, texture, and composition. This visual information is interrelated and numerically represented at the pixel level. Therefore, when analyzing the visual information of an entire image, raw images are interpreted as high-dimensional data, represented by the product of the height, width, and depth. These data include not only the visual information necessary for prediction (e.g., information about the animal itself in animal classification) but also irrelevant visual information (e.g., background information in animal classification). These irrelevant details increase the risk of the ML model generating incorrect decision criteria, which can reduce its generalization performance. Furthermore, in the case of high-dimensional data, the sample size required for learning is enormous, making it difficult to ensure sufficient data density. A decrease in data density is one factor that can prevent the model from appropriately learning important features for prediction. Therefore, to improve the generalization performance, it is essential to extract the relevant visual information that contributes toward solving the problem from high-dimensional data and to summarize it into a lower-dimensional feature vector [26]. In DL, a process is performed to summarize diverse visual information embedded in image data into numerical representations using several parameters across multiple layers. In contrast, non-DL models generally lack such transformation mechanisms, making feature extraction necessary to summarize the visual information of image data into meaningful features. Texture information, such as surface patterns and regularities of histopathological images of the myocardium, are generally considered useful for differentiating cardiomyopathies [27, 28]. This suggests that texture features are particularly suitable for feature extraction from histopathological images of the

myocardium. Texture features are widely employed in the construction of diagnostic models using medical images in ML, and their effectiveness has been widely reported [29, 30]. However, to the best of our knowledge, few studies have been conducted to evaluate the effectiveness of texture features from histopathological images of the myocardium for the pathological diagnosis of cardiomyopathies using ML. Thus, it is crucial to verify whether texture features from histopathological images of the myocardium are effective in predicting the pathological diagnosis of cardiomyopathies.

Next, the design of models robust to small sample sizes is discussed (Figure 1: Research problem B). Data with small sample sizes often present various challenges, such as a reduction in the data density within the feature space [31] and domain shift [32]. Collectively, these issues contribute to the risk of overfitting. Therefore, achieving high generalization performance with small sample size data requires model designs that demonstrate robust performance against these challenges. When constructing ML models using small sample size data, the use of a reduced-dimensional feature space is recommended [33]. This approach increases data density by reducing dimensionality, which is expected to enhance generalization performance. However, the degree to which the dimensionality should be reduced may vary depending on the specific task. For instance, a higher-dimensional feature space allows the incorporation of more information regarding target prediction, potentially improving the prediction performance. Conversely, the inclusion of irrelevant information may also increase, resulting in the generation of decision criteria based on meaningless data, which could cause overfitting and reduce the generalization performance. Therefore, it is essential to carefully determine an acceptable level of dimensionality.

In addition, it is important to consider not only the dimensionality of the feature space but also the complexity of the ML model (Figure 1: Research problem C). A model with controlled complexity can avoid overfitting to the training data [34]. Consequently, even with a small dataset, high generalization performance on unseen data can be expected. However, within this context, decision boundaries such as straight lines, curves, or boundaries perpendicular or parallel to the axes may exist, and the suitability of the decision boundary depends on the specific task. In general, models that use linear decision boundaries tend to exhibit simpler internal structures, which helps suppress overfitting. However, achieving high prediction performance may be difficult [34]. Conversely, models with nonlinear decision boundaries exhibit relatively complex internal structures, which can result in a higher goodness of fit, but may suffer from lower generalization performance [34]. To construct a classification model that approximates the optimal prediction performance, it is necessary to carefully determine an acceptable level of model complexity.

1.4. Objective of this study

This study investigates two aspects: the effectiveness of texture features from histopathological images of the myocardium for pathological diagnosis of cardiomyopathies, and the design of models suitable for pathological diagnosis of cardiomyopathies using small sample size data. As reported in previous research, we constructed a preliminary ML model based on a single model design and performed a simple investigation of the potential of texture features for pathological diagnosis of three types of cardiomyopathies [35]. The obtained results indicated that texture features clearly represent differences in myocardial conditions. Based on this, this study constructs a three-class classification model based on texture features and multiple model designs. Through visualization and statistical analysis of texture features, and performance evaluation of the classification models, a more rigorous

validation of the effectiveness of the texture features and the optimal model design was conducted. The clarification of these aspects will contribute to the pathological diagnosis of cardiomyopathies based on high diagnostic performance in environments where large sample sizes cannot be collected. Moreover, it is anticipated that, in the future, rapid and accurate diagnoses enable timely treatment for patients.

The remainder of this paper is organized as follows: Section 2 provides an overview of the ML methods used in this study. Section 3 describes the experimental procedures, details of the datasets, and the parameters employed in the experiments. Section 4 presents the experimental results and discusses their implications. Section 5 concludes the paper with a summary of the study, its limitations, and future prospects.

2. Machine learning (non-deep learning) methods

2.1. Texture feature extraction

In this study, the following texture analysis methods were employed: first-order statistics (FOS), gray level difference statistics (GLDS), the gray level co-occurrence matrix (GLCM), the gray level run length matrix (GLRLM), the angular distribution function based on the Fourier power spectrum (ADF), and the radial distribution function based on the Fourier power spectrum (RDF). FOS refers to statistical measures derived from a distribution in which the gray level values of image pixels are plotted on the x-axis, and the corresponding occurrence probabilities are plotted on the y-axis. GLDS represents statistical measures derived from distributions in which the gray level differences between a given pixel and its neighboring pixel, at a specified direction and distance, are plotted on the x-axis, with the occurrence probabilities of these differences plotted on the y-axis. These distributions are calculated for the horizontal, vertical, and diagonal directions, and the resulting statistics are averaged across all directions. The GLCM is a matrix that aggregates the occurrence probabilities of pairs of gray level values in pixels adjacent to a given pixel at specified distances and directions. The GLRLM is a matrix that aggregates the occurrence probabilities of consecutive runs of pixels with the same grey-level value in a specified direction. The ADF refers to the distribution derived from the power spectrum image obtained via a two-dimensional Fourier transform, where the angular directions from the center of the spectrum are plotted on the x-axis, and the intensity of the frequency components along these directions is plotted on the y-axis. The RDF is a distribution in which the distance from the center of the power spectrum image obtained via a two-dimensional Fourier transform is plotted on the x-axis, and the intensity of the frequency components at that distance is plotted on the y-axis.

Texture features were calculated from various statistical measures derived from the distributions and matrices obtained using texture analysis methods. Table 1 lists the texture analysis methods and their corresponding statistical measures. For FOS, GLDS, the ADF, and the RDF, seven statistical measures were adopted: mean, contrast, variance, skewness, kurtosis, energy, and entropy. The mean quantifies the center of mass of the probability distribution. The value decreases as the distribution shifts to the left and increases as it shifts to the right. The contrast is an indicator that quantifies the magnitude of the probability distribution, excluding the sign. Regardless of the sign, it increased as the probability distribution moved further from 0. Variance quantifies the extent to which the data deviate from the mean of the probability distribution. The value is smaller when the data are concentrated around the mean, and larger when they are spread out. Skewness quantifies the asymmetry of a probability distribution compared to a normal distribution. If the distribution is skewed to the left with a longer

right tail, the value is positive; if it is skewed to the right with a longer left tail, the value is negative. In addition, the magnitude of skewness increases as the degree of asymmetry increases, resulting in larger positive or negative values. Kurtosis is a measure that quantifies the sharpness of the peak and the heaviness of the tails of a probability distribution in comparison with a normal distribution. When the distribution peaks have heavier tails than a normal distribution, kurtosis has a positive value; when the distribution is flatter with lighter tails, it has a negative value. Furthermore, the stronger this tendency, the larger the value is in the positive or negative direction. Energy quantifies the concentration of the probability distribution. The value is higher when the distribution is concentrated at specific values, and lower when it is spread out over a wider range. Entropy quantifies the closeness of a distribution to a uniform distribution. The value increases as the distribution approaches uniformity, and decreases as the distribution concentrates around specific values.

For the GLCM, six statistical measures were adopted: contrast, correlation, joint energy, joint entropy, inverse difference moment (IDM), and inverse variance. Correlation is a measure that quantifies the linear dependence of shade changes between adjacent pixels. Strong linear dependence between shade changes in adjacent pixels (e.g., when bright pixels are surrounded by similar pixels and the shade changes in a step-like manner), the value is close to 1. However, when there is a weak linear dependence (e.g., when bright pixels are surrounded by dark pixels), the value is close to 0. Joint energy quantifies the degree of continuity of specific shade patterns between adjacent pixels. The larger the value, the more frequently a specific shade pattern appears between adjacent pixels, while the smaller the value, the more evenly distributed the various shade patterns are. Joint entropy quantifies the diversity and irregularity of shade patterns between adjacent pixels. The larger the value, the more diverse and irregular the shade patterns are between adjacent pixels, while the smaller the value, the more the pattern is biased toward a specific shade. The IDM is an index that quantifies the uniformity and smoothness of an image. The larger the value, the more uniform the image is, with pixels of equal tone values appearing successively; conversely, the smaller the value, the more uneven the image is, with pixels of high- and low-tone values appearing in succession. Inverse variance is a measure that quantifies the variation in changes in grayscale values between adjacent pixels. The larger the value, the more frequently small differences in grayscale values occur between adjacent pixels, indicating less variation. Conversely, the smaller the value, the larger the differences in grayscale values between adjacent pixels, indicating greater variation.

For the GLRLM, five statistical measures were adopted: short run emphasis (SRE), long run emphasis (LRE), gray level non-uniformity (GLN), run length non-uniformity (RLN), and run percentage (RP). SRE is an index that quantifies the fineness of an image. The higher the value, the shorter the run length in the image (the finer the image). LRE is an index that quantifies the coarseness of an image. The higher the value, the longer is the run length in the image (the coarser the image). GLN is an index that quantifies the uniformity of the distribution of grayscale values. The larger the value, the more frequently a particular gray value appears compared with other gray values (the distribution of gray values is uneven). However, the smaller the value, the more even the gray values appear (the distribution of gray values is even). RLN quantifies the evenness of the distribution of run lengths. The larger the value, the more frequently a particular run length appears compared with other run lengths (the distribution of run lengths is uneven). In contrast, the more even the run lengths (the more uniform the run length distribution), the smaller the value. RP quantifies the coarseness or fineness of a texture. The shorter the number of runs (the more detailed the image), the larger the value;

conversely, the longer the runs (the coarser the image), the smaller the value.

In this study, a Python package was developed to analyze the texture information of images and compute their statistical features. By utilizing this package, a total of 39 texture features, as shown in Table 1, were calculated. The features related to the GLCM and GLRLM were implemented using the Python package Pyradiomics (version: 3.1.0) [36].

Table 1. Texture analysis methods and their corresponding statistical measures (◯ indicates applied).

Statistical measures[1]	Texture analysis methods[2]					
	FOS	GLDS	GLCM	GLRLM	ADF	RDF
Mean	◯	◯	—	—	◯	◯
Contrast	◯	◯	◯	—	◯	◯
Variance	◯	◯	—	—	◯	◯
Skewness	◯	◯	—	—	◯	◯
Kurtosis	◯	◯	—	—	◯	◯
Energy	◯	◯	—	—	◯	◯
Entropy	◯	◯	—	—	◯	◯
Correlation	—	—	◯	—	—	—
Joint energy	—	—	◯	—	—	—
Joint entropy	—	—	◯	—	—	—
IDM	—	—	◯	—	—	—
Inverse variance	—	—	◯	—	—	—
SRE	—	—	—	◯	—	—
LRE	—	—	—	◯	—	—
GLN	—	—	—	◯	—	—
RLN	—	—	—	◯	—	—
RP	—	—	—	◯	—	—

[1] IDM stands for inverse difference moment. SRE stands for short run emphasis. LRE stands for long run emphasis. GLN stands for gray level non-uniformity. RLN stands for run length non-uniformity. RP stands for run percentage.

[2] FOS stands for first-order statistics. GLDS stands for gray level difference statistics. GLCM stands for gray level co-occurrence matrix. GLRLM stands for gray level run length matrix. ADF stands for angular distribution function based on the Fourier power spectrum. RDF stands for radial distribution function based on the Fourier power spectrum.

Because the scale of each texture feature is different, considering the importance of each feature equally, it is necessary to make the features dimensionless through standardization. For small sample size datasets, each feature is highly likely to be non-normally distributed. Furthermore, when applying standardization, robustness to outliers is necessary. Therefore, robust standardization, a standardization method that is robust to outliers and is not affected by the shape of the distribution, was adopted here. Scaling through robust standardization was performed as follows:

$$z(x) = \frac{x - \tilde{x}}{\mathrm{IQR}}, \tag{2.1}$$

where x is an element of the texture feature, $\tilde{x}$ is the median of the texture feature, and IQR is the interquartile range of the texture feature.

2.2. Feature selection

FS is the process of selecting features that are useful for prediction and excluding features that may be noisy or not useful. This is performed using an algorithm suited to the characteristics of the dataset and the problem being addressed. In this study, the within-class variance between-class variance ratio was adopted as the FS method. In this method, the total sum of the Euclidean distances between the average vectors of all samples and the average vectors of each class were divided by the sum of the variances of each class, which was considered the final evaluation value. Features with sufficiently separated inter-class distances and small within-class variances are desirable; therefore, a higher evaluation value is preferable. This method has the advantage of easily selecting features that can clearly distinguish different classes of classification problems, because it considers the distance between classes and the variance of each class simultaneously. In addition, because the separation relationship between the classes is simple, it is possible to obtain a feature space that is not prone to overfitting, even in the case of small sample size data. Another advantage is that the selected features are easy to interpret. However, there is a drawback in that it is not possible to accurately evaluate the spatial structure seen in anomaly detection, where a specific class is concentrated at a single point in the feature space and the other classes are scattered around it. Such a spatial structure is expected to contribute to an improved prediction performance; however, this also increases the risk of overfitting, making it difficult to interpret the selected features. Therefore, these features were excluded from the selection in this study.

In this study, the within-class variance between-class variance ratio was applied individually to each texture feature. The FS evaluation value is calculated using one dimension at a time. This was done to reduce the search range for FS and reduce the calculation and time costs. For example, when evaluating a one-dimensional feature from 39 feature types, ${}_{39}C_1 = 39$ different evaluations are required. When evaluating a four-dimensional feature space, ${}_{39}C_4 = 82{,}251$ different evaluations were required. When evaluating a seven-dimensional feature space ${}_{39}C_7 = 15{,}380{,}937$ different evaluations are required. It can be observed that the higher the dimensionality of the evaluated space, the wider the search range, making it more difficult to obtain an answer that approximates the optimal solution in a realistic amount of time.

On the other hand, for a one-dimensional evaluation, the existence of feature values that can clearly classify multiple classes in one dimension is a prerequisite. However, there is a possibility that such feature values do not exist. Therefore, simplification of feature value selection was attempted by dividing the multiclass classification problem into multiple two-class classification problems. For example, in a three-class classification problem (C_a, C_b, C_c), the problem is divided into a combination of multiple two-class classification problems (C_a, C_b), (C_a, C_c), and (C_b, C_c). Thereafter, the search for useful features for the two-class classification problem was conducted for each of these, and the union of the obtained features, excluding duplicates, was extracted as a feature useful for three-class classification. That is, if n useful features are selected for each two-class classification problem, then $3n$ features are selected for the entire three-class classification problem. If there are d overlapping features among them, then $3n - d$ features are selected after excluding them.

2.3. Dimensional compression

DC is a method for converting data into a low-dimensional space while minimizing the loss of information regarding the spatial structure in a multidimensional feature space. In this study, the supervised DC method, Fisher's linear discriminant analysis (LDA), was adopted.

LDA is a method for performing a linear transformation that maximizes the between-class variance and minimizes the within-class variance based on the training data [37]. In this method, by maximizing the feature values that contribute to class identification, each class is expected to be clearly separated, even after transformation to a low-dimensional space. However, the dimensionality of the transformed space depends on the number of classes c, and the dimensionality of the transformed space is constrained to $c - 1$ or less [38]. Therefore, the number of dimensions that can be selected is only one dimension for two-class classification and only one or two dimensions for three-class classification. In this study, two-dimensional compression was adopted, which considers the interaction of feature values because it targets three-class classification.

The LDA implementation used the Python package scikit-learn (version: 1.2.2) [39].

2.4. Classification models and evaluation metrics

For small sample size data, a prediction model with limited complexity is used to avoid overfitting [40]. However, the degree of complexity depends not only on the sample size and number of features but also on the domain, such as the difficulty of solving a problem [34]. Therefore, it is not always optimal to choose a simple model, and it is believed that a certain degree of complexity is necessary at the decision boundary to achieve a high prediction performance. In this study, to verify this, a support vector machine (SVM) was adopted as a simpler model compared to other classification models. In this case, a linear kernel (LK) and radial basis function kernel (RBF) were adopted as the kernel functions. In addition, a decision tree (DT) was adopted as the model with a decision boundary that is perpendicular or horizontal to the axis based on a tree structure algorithm. This was performed to verify the appropriate complexity of the decision boundary for myocardial pathology diagnosis by comparing the evaluation values based on the linearity and nonlinearity of the decision boundaries. These were implemented using the Python package scikit-learn (version: 1.2.2) [39].

The prediction performance of each classification model is evaluated based on the macro-average F1-score. The F1-score is an evaluation metric for two-class classification that uses the harmonic mean of the precision and recall as its evaluation value. Precision is a metric that indicates the proportion of samples that are predicted to be positive or actually positive. Recall is a measure of the proportion of samples that were correctly predicted as positive among the samples that were actually positive. The F1-score enables the prediction performance of each class to be reflected equally, even in the case of unbalanced data with bias in the samples for each class in two-class classification. The macro-average F1-score is an indicator that can be extended to multiple classes, and is effective when all classes are evaluated equally. This evaluation takes values between 0 and 1, with values closer to 1 indicating better prediction performance.

2.5. Cross-validation and hyperparameter optimization

Cross-validation is a method for statistically evaluating a model's generalization performance by repeatedly constructing and evaluating it using a subset of the original dataset as validation or test

data, while the remaining data are used for training. In this study, stratified K-fold cross-validation was adopted as a cross-validation method. This method creates a subset from the original dataset, such that the sample ratio of each class is equal. For example, consider a dataset with 20 samples of disease cases, 20 samples of borderline cases, and 20 samples of normal cases, which is then divided into training data and test data (Figure 2: Step 1). The sample size ratio was set to training:test = 8:2. Five subsets were created, each containing four disease cases, four borderline cases, and four normal cases, with no overlapping samples between the subsets. The training set was then formed by combining four of these subsets (16 disease cases, 16 borderline cases, and 16 normal cases), while the remaining subset (four disease cases, four borderline cases, and four normal cases) was allocated to the test set. Finally, this subset replacement was performed five times. The generalization performance of the model was statistically evaluated based on the evaluation values of the five subsets. The same procedure applies to the division of the split training and validation set (Figure 2: Step 2).

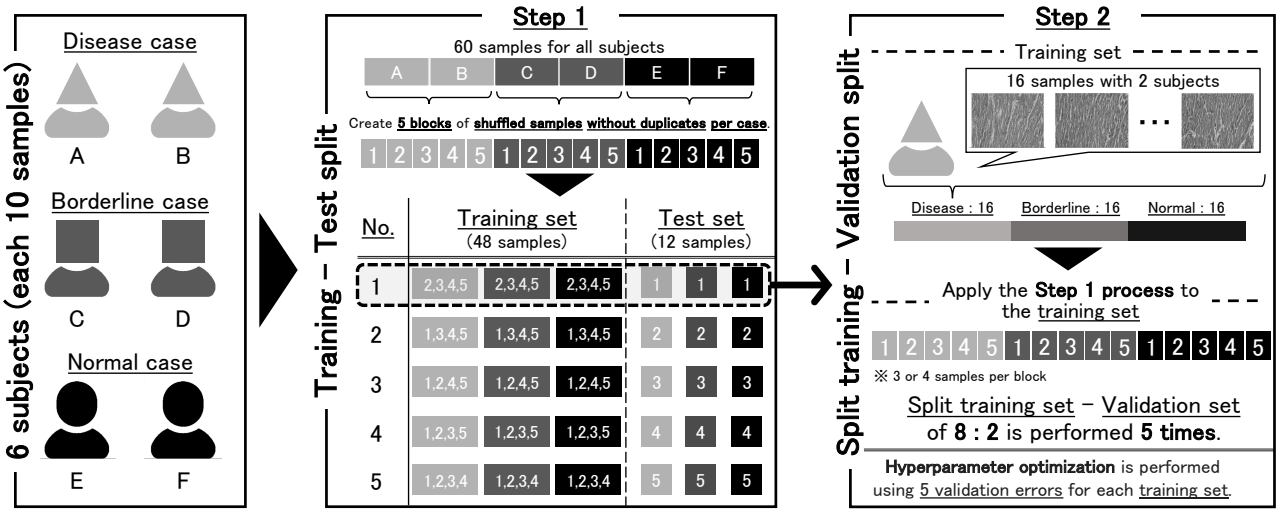


Figure 2. Overview of stratified K-fold cross-validation with a nested structure for $K = 5$.

Hyperparameter optimization based on minimizing validation error was performed during cross-validation of the training and validation sets. Hyperparameter optimization is the process of adjusting hyperparameters that influence the complexity of an ML model using an optimization method. In this study, the hyperparameters of texture analysis methods and the hyperparameters of the ML models were the targets of optimization. In this case, Bayesian optimization was adopted as the optimization method. The hyperparameters that are the target of optimization are shown in Table 2. For hyperparameters not listed here, the default values of the Python packages used in each method were adopted.

Table 2. Hyperparameters to be optimized and their search ranges.

Target[1]	Hyperparameter	Search range[2, 3]
FOS	bin width	$p_1 \in \{4, 16, 64, 256\}$
GLDS	bin width	$p_2 \in \{4, 16, 64, 256\}$
	distance	$p_3 \in \{1, 2, 3, 4\}$
GLCM	bin width	$p_4 \in \{4, 16, 64, 256\}$
	distance	$p_5 \in \{1, 2, 3, 4\}$
GLRLM	bin width	$p_6 \in \{4, 16, 64, 256\}$
ADF	infinitesimal angle	$p_7 \in \{1, 2, 3, 4\}$
RDF	infinitesimal radius	$p_8 \in \{1, 2, 3, 4\}$
FS	number of FS-selected features	$p_{c_1} \in \{2, 3, 4, \cdots, 38\}$
FS + DC	number of FS-selected features with DC	$p_{c_2} \in \{3, 4, 5, \cdots, 38\}$
SVM with LK	cost parameter	$p_{c_3} \in [0.0001, 10000]$
SVM with RBF	cost parameter	$p_{c_4} \in [0.0001, 10000]$
	γ parameter	$p_{c_5} \in [0.0001, 10000]$
DT	criterion	$p_{c_6} \in$ {"gini", "entropy", "log loss"}
	max depth	$p_{c_7} \in \{1, 2, 3, 4, 5\}$

[1] FOS stands for first-order statistics. GLDS stands for gray level difference statistics. GLCM stands for gray level co-occurrence matrix. GLRLM stands for gray level run length matrix. ADF stands for angular distribution function based on the Fourier power spectrum. RDF stands for radial distribution function based on the Fourier power spectrum. FS stands for feature selection. DC stands for dimensional compression. SVM stands for support vector machine. LK stands for linear kernel. RBF stands for radial basis function kernel. DT stands for decision tree.

[2] $\{\cdot\}$ represents each discrete value or category, and $[A, B]$ represents a continuous value of A or more and B or less.

[3] The subscript c of the parameter variable varies depending on the element of the selected model vector (Equation 3.2). For example, if the combination is FS and SVM with LK (m_1 = FS, m_2 = None, m_3 = SVM with LK), it is $c_1 = 9$ and $c_3 = 10$. If the combination is FS + DC and DT (m_1 = FS, m_2 = DC, m_3 = DT), it is $c_2 = 9$, $c_6 = 10$, $c_7 = 11$.

The bin width in Table 2 is a parameter that determines the discretization of grayscale image tone values. For example, if the bin width is set to 64, a conversion process is applied, in which 0 becomes black and 63 becomes white. The distance in Table 2 is a parameter that determines the distance between the pixels to be compared when comparing the tone values between the pixels. For example, if the distance is set to 1, a comparison is made between the pixel and the next pixel. If the distance is set to 2, then a comparison is made between pixels and two pixels away. The infinitesimal angle in Table 2 is a parameter which determines the smoothness of the angular distribution. The smaller this value, the smoother the angular distribution and the more detailed the trend that can be captured. However, its disadvantage is that it requires a longer processing time. The infinitesimal radius in Table 2 is a parameter which determines the smoothness of the radial distribution. The smaller this value, the smoother the radial distribution and the more detailed the trend that can be captured. However, its disadvantage is that it requires a longer processing time. The number of FS-selected features in Table 2 is a parameter that determines the number of features selected in the FS. The same applies to the number of FS-selected features with DC. When this number is small, fewer features are selected, which increases the likelihood of excluding features that are important for prediction. On the other hand, the

larger this value is, the more features are selected; thus, there is a greater chance of selecting features that could be noisy and unnecessary for prediction. The cost parameter in Table 2 determines the extent to which misclassification is allowed when there is a mixture of samples. The smaller the value, the more misclassification is allowed, and the larger the value, the less misclassification is allowed. The γ parameter in Table 2 is a parameter that determines the complexity of the decision boundary. The smaller the value, the simpler the decision boundary, and the larger the value, the more complex the decision boundary. The criterion in Table 2 is a hyperparameter that determines how to split nodes in a DT. "gini" is the Gini impurity, "entropy" is the information gain based on Shannon entropy, and "log loss" is the cross-entropy loss. All these metrics measure the uncertainty and/or mixture of the class distribution. The max depth in Table 2 is a hyperparameter that determines the maximum depth of the DT. The smaller this value is, the simpler the DT structure becomes, which helps reduce the risk of overfitting. However, the larger this value, the more complex the DT structure becomes, leading to a higher goodness of fit on the training data but also an increased risk of reduced generalization performance.

Stratified K-fold cross-validation was implemented using the Python package scikit-learn (version: 1.2.2) [39]. Bayesian optimization was implemented using the Python package optuna (version: 3.5.0) [41].

3. Experiment

3.1. Objectives and overview

This study aims to verify the effectiveness of texture features in the pathological diagnosis of cardiomyopathy and to clarify a model design suitable for diagnosis with small sample size data. These aspects are investigated using a three-class classification problem that diagnoses three patterns: two types of cardiomyopathy and the normal state. Texture features are extracted from histopathological images of the myocardium. The effectiveness of the texture features is verified by visualizing them and investigating the significance of the differences in class distribution using statistical hypothesis testing. In addition, the model design most suitable for diagnosing cardiomyopathy using small sample data is clarified by comparing the prediction performance of each model design.

The experimental procedure is shown in Algorithms 1, 2, and 3. First, the collection and processing of the dataset used in this experiment are explained. Next, additional details on some of the variables and functions used in the algorithms are provided. Finally, a step-by-step explanation of the specific experimental procedure is given.

Algorithm 1 An algorithm for evaluating the predictive performance of a model design.

Input:

D: Dataset ▷ Sec. 3.2

$\boldsymbol{m}$: Model vector, K: Number of cross-validation splits, B: Iterations in Bayesian optimization ▷ Sec. 3.3

Output:

$(s^{\text{mean}}_{\text{train}}, s^{\text{std}}_{\text{train}})$, $(s^{\text{mean}}_{\text{valid}}, s^{\text{std}}_{\text{valid}})$, $(s^{\text{mean}}_{\text{test}}, s^{\text{std}}_{\text{test}})$: Mean and standard deviation of macro-average F1-scores for each dataset ▷ Tab. 3

Dataset type: $J \leftarrow \{\text{train}, \text{valid}, \text{test}\}$

Initialize score sets: $S_j \leftarrow \emptyset$, $j \in J$

Make subsets: $D_1, D_2, \cdots, D_K \leftarrow \text{CrossVal}(D, K)$ ▷ Sec. 2.5 & 3.3

for $k \leftarrow 1$ **to** K **do**

  Make training set: $D_{\text{train}} \leftarrow D \setminus D_k$ ▷ Fig. 2

  Make test set: $D_{\text{test}} \leftarrow D_k$ ▷ Fig. 2

  $s_{\text{valid}}, \boldsymbol{p}^* \leftarrow \text{SearchBestParameters}(D_{\text{train}}, K, B, \boldsymbol{m})$ ▷ Alg. 2

  Extract features: $F_{\text{train}}, F_{\text{test}} \leftarrow \text{TextureFeatures}(D_{\text{train}}, D_{\text{test}}, \boldsymbol{p}^*)$ ▷ Sec. 2.1

  Reduce features: $F_{\text{train}}, F_{\text{test}} \leftarrow \text{DimensionalityReduction}(F_{\text{train}}, F_{\text{test}}, \boldsymbol{m}, \boldsymbol{p}^*)$ ▷ Alg. 3

  Get scores: $s_{\text{train}}, s_{\text{test}} \leftarrow \text{ClassifierScore}(F_{\text{train}}, F_{\text{test}}, \boldsymbol{m}, \boldsymbol{p}^*)$ ▷ Sec. 2.4

  Merge scores: $S_j \leftarrow S_j \cup \{s_j\}$, $j \in J$

end for

Get means of scores: $s^{\text{mean}}_j \leftarrow \text{Mean}(S_j)$, $j \in J$

Get standard deviations of scores: $s^{\text{std}}_j \leftarrow \text{Std}(S_j)$, $j \in J$

return $\{(s^{\text{mean}}_j, s^{\text{std}}_j) \mid j \in J\}$

Algorithm 2 Bayesian optimization-based hyperparameter search with cross-validation.

Input:

D_{train}: Training set, $\boldsymbol{m}$: Model vector, K: Number of cross-validation splits, B: Iterations in Bayesian optimization ▷ Sec. 3.3

Output:

$s_{\text{valid}}^{\max}$: Maximum score of validation data, $\boldsymbol{p}^*$: Optimized hyperparameter vector

Make subsets: $D_1, D_2, \cdots, D_K \leftarrow \text{CrossVal}(D_{\text{train}}, K)$ ▷ Sec. 2.5 & 3.3
for $b \leftarrow 1$ **to** B **do**
  Initialize validation score sets: $S_{\text{valid}} \leftarrow \varnothing$
  $\boldsymbol{p} \leftarrow$ Draw the hyperparameter vector from Table 2 by Bayes. opt. ▷ Sec. 2.5 & 3.3
  for $k \leftarrow 1$ **to** K **do**
    Make split training set: $D_{\text{st}} \leftarrow D_{\text{train}} \setminus D_k$ ▷ Fig. 2
    Make validation set: $D_{\text{valid}} \leftarrow D_k$ ▷ Fig. 2
    Extract features: $F_{\text{st}}, F_{\text{valid}} \leftarrow \text{TextureFeatures}(D_{\text{st}}, D_{\text{valid}}, \boldsymbol{p})$ ▷ Sec. 2.1
    Reduce features: $F_{\text{st}}, F_{\text{valid}} \leftarrow \text{DimensionalityReduction}(F_{\text{st}}, F_{\text{valid}}, \boldsymbol{m}, \boldsymbol{p})$ ▷ Alg. 3
    Get scores: $s_{\text{valid}} \leftarrow \text{ClassifierScore}(F_{\text{st}}, F_{\text{valid}}, \boldsymbol{m}, \boldsymbol{p})$ ▷ Sec. 2.4
    Merge scores: $S_{\text{valid}} \leftarrow S_{\text{valid}} \cup \{s_{\text{valid}}\}$ ▷ Sec. 2.4
  end for
  Record mean score: $s_{\text{valid},\,b}^{\text{mean}} \leftarrow \text{Mean}(S_{\text{valid}})$
  Record hyperparameter: $\boldsymbol{p}_b \leftarrow \boldsymbol{p}$
end for
Get optimal step id.: $b^* \leftarrow \arg\max_b \{s_{\text{valid},\,b}^{\text{mean}} \mid b \in \{1, 2, \cdots, B\}\}$
Get maximum validation score: $s_{\text{valid}}^{\max} \leftarrow s_{\text{valid},\,b^*}^{\text{mean}}$
Get optimal parameter: $\boldsymbol{p}^* \leftarrow \boldsymbol{p}_{b^*}$
return $s_{\text{valid}}^{\max}, \boldsymbol{p}^*$

Algorithm 3 Feature selection and/or dimensional compression, or no processing, for texture features.

Input:

F_a, F_b: Texture features of datasets a and b, $\boldsymbol{m}$: Model vector, $\boldsymbol{p}$: Hyperparameter vector ▷ Sec. 3.3

Output:

F_a, F_b: Texture features to which dimensionality reduction has not been applied or has been applied

if m_1 = FS **then** ▷ Eq. 3.3
  Select features: $F_a, F_b \leftarrow \text{FeatureSelection}(F_a, F_b, \boldsymbol{m}, \boldsymbol{p})$ ▷ Sec. 2.2
end if
if m_2 = DC **then** ▷ Eq. 3.4
  Compress features: $F_a, F_b \leftarrow \text{DimensionalCompression}(F_a, F_b)$ ▷ Sec. 2.3
end if
return F_a, F_b

3.2. Subjects and data description

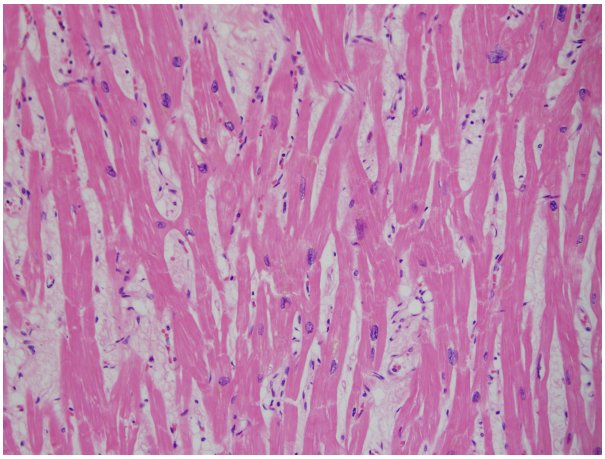
Four patients histopathologically diagnosed with hypertrophic cardiomyopathy (HCM) at autopsy from January 1, 2020, to December 31, 2022, at Nihon University Itabashi Hospital and University Hospital were enrolled. Among them, two patients with histopathologically confirmed cardiomyocyte disarray characteristics of HCM were considered disease cases, and two patients without cardiomyocyte disarray were considered borderline cases. In addition, two patients who died due to reasons other than cardiac disease and were autopsied during the abovementioned period were considered normal cases. Histological sections 4 micrometers in thickness from myocardial specimens were prepared and stained with hematoxylin-eosin. Ten histological sections were imaged using a microscope equipped with a digital camera (Digital microscopic system, DP-70, Olympus Corporation, Japan; 200× magnification) for each of the six cases. In other words, there were six subjects (two disease cases, two borderline cases, and two normal cases), and the sample size was 60. The protocol was approved by the Ethics Review Board of Nihon University Itabashi Hospital (RK-210914-18).

The histopathological images of the myocardium are shown in Figure 3. These were 256-level color images composed of $4096 \times 3086 \times 3$ pixels each. Because the similarity between surrounding pixels is high, the computational cost increases unnecessarily. Therefore, to facilitate efficient image analysis, the images were resized to $204 \times 154 \times 3$ pixels using bilinear interpolation. In general, texture features are typically extracted from grayscale images rather than color images. In this study, the color images were converted to grayscale using the Python package opencv-python (version: 4.7.0.68), following the equation

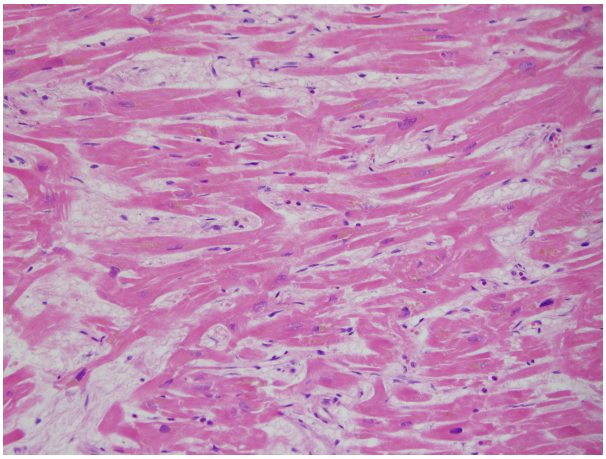
$$\boldsymbol{Y} = 0.299\boldsymbol{R} + 0.587\boldsymbol{G} + 0.114\boldsymbol{B}, \tag{3.1}$$

defined in Recommendation ITU-R BT.601. Here, $\boldsymbol{Y}$ denotes the grayscale image matrix, and $\boldsymbol{R}$, $\boldsymbol{G}$, and $\boldsymbol{B}$ represent the red, green, and blue channel matrices of the original color image, respectively.

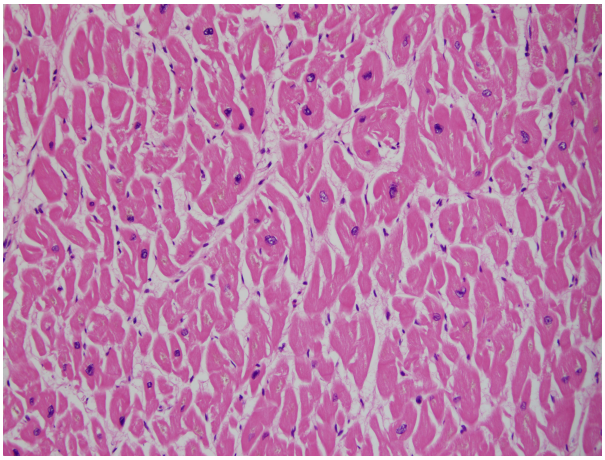
The dataset D used in this experiment is a set of vectors where the elements are pairs of converted grayscale images and their corresponding myocardial states (class labels).



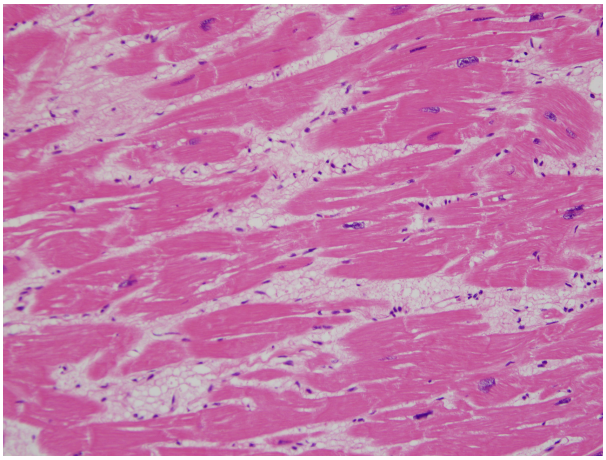
(a) Disease case: Subject A



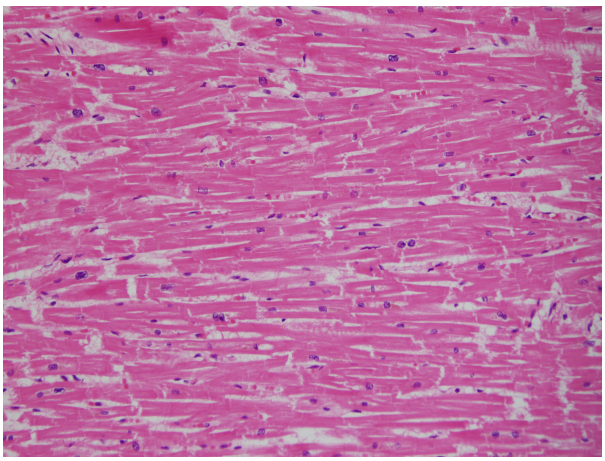
(b) Disease case: Subject B



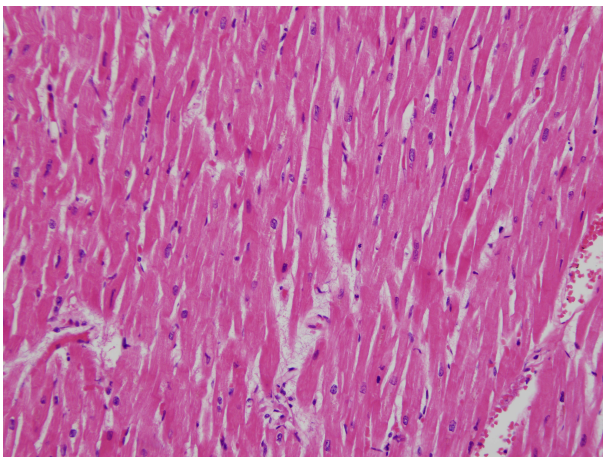
(c) Borderline case: Subject C



(d) Borderline case: Subject D



(e) Normal case: Subject E



(f) Normal case: Subject F

Figure 3. Histopathological images of the myocardium from disease cases, borderline cases, and normal cases in six subjects.

3.3. Explanation of the variables and functions used in the algorithms

First, the model vector $\boldsymbol{m}$ that represents the design of each model is defined. In this study, the model vector is defined as

$$\boldsymbol{m} = \begin{bmatrix} m_1 & m_2 & m_3 \end{bmatrix}^\top \in M. \tag{3.2}$$

However,

$$m_1 \in M_1 := \{\text{FS}, \text{AF}\}, \tag{3.3}$$
$$m_2 \in M_2 := \{\text{DC}, \text{None}\}, \tag{3.4}$$
$$m_3 \in M_3 := \{\text{SVM with LK}, \text{SVM with RBF}, \text{DT}\}, \tag{3.5}$$
$$M = M_1 \times M_2 \times M_3. \tag{3.6}$$

For example, $\boldsymbol{m} = \begin{bmatrix} \text{FS} & \text{None} & \text{SVM with RBF} \end{bmatrix}^\top$ means that FS was performed, no DC was applied, and SVM using RBF was used. In addition, $\boldsymbol{m} = \begin{bmatrix} \text{AF} & \text{DC} & \text{DT} \end{bmatrix}^\top$ means that a DT was used for features to which DC was applied across all features. The number of patterns that $\boldsymbol{m}$ can be used is $|M| = |M_1| \times |M_2| \times |M_3| = 2 \times 2 \times 3 = 12$. Therefore, there are 12 model vectors $\boldsymbol{m}_1, \boldsymbol{m}_2, \cdots, \boldsymbol{m}_{12} \in M$.

Next, we provide additional details on the CrossVal function, which is a dataset partitioning function based on stratified K-fold cross-validation. Let D be a dataset comprising pairs of input images and class labels. This dataset D is divided into K subsets of equal size using the function

$$D_1, D_2, \cdots, D_K = \text{CrossVal}(D, K). \tag{3.7}$$

Here,

$$|D_1| = |D_2| = \cdots = |D_K|, \; D = \bigcup_{k=1}^{K} D_k, \tag{3.8}$$

is assumed. That is, all subsets D_k have the same number of data samples, and there is no data overlap among the different subsets. In this case, the class-label distribution remained uniform across all subsets (Sections 2.5). The number of splits, denoted as K, used for the stratified K-fold cross-validation represents the number of data blocks used to divide the dataset D into training, validation, and test sets (Figure 2). In this experiment, the value of K was set to five times for both the training–test and the split training–validation splits, as shown in Figure 2. Therefore, the training set D_{train} and test set D_{test} were divided at a ratio of 8:2 (Algorithm 1: Lines 5–6). In addition, the split training set D_{st} and validation set D_{valid} were divided at a ratio of 8:2 (Algorithm 2: Lines 6–7). In other words, the training and test sets were evaluated five times, and the validation set was evaluated 25 times.

Finally, the hyperparameter vector $\boldsymbol{p}$, which corresponds to the texture features, FS, and the classification model in Table 2, is defined. It is represented by

$$\boldsymbol{p} = \begin{bmatrix} p_1 & p_2 & p_3 & \cdots \end{bmatrix}^\top. \tag{3.9}$$

In Table 2, the hyperparameters of the texture features, bin width for FOS, GLDS, GLCM, and GLRLM (p_1, p_2, p_4, p_6), were set to four gradation values (4, 16, 64, and 256), which were powers of four in

each case. In addition, for the pixel distances of GLDS and GLCM (p_3, p_5), four values from one to four pixels were adopted for each. Furthermore, for the infinitesimal angle of ADF (p_7) and the infinitesimal radius of RDF (p_8), four values from one to four were used, with angles in degrees and radii in pixels. For the number of FS-selected features (p_{c_1}), a hyperparameter for FS, 37 values from 2 to 38 were used. In this study, 39 features are extracted regardless of the image or sample size of the histopathological images of the myocardium (Table 1). Therefore, the number of dimensions to be selected ranged from 2 to 38, which is the minimum number of dimensions in which the interaction of the features can be considered. However, when DC is applied, the number of dimensions is two, as described in Section 2.3. Thus, in the number of FS-selected features with DC (p_{c_2}), the number of dimensions for FS is set to three or more. The cost parameter (p_{c_3}, p_{c_4}) and γ parameter (p_{c_5}), which are hyperparameters of the classification model, were adopted in the range of 1×10^{-4} to 1×10^{4}. For criterion (p_{c_6}), three values of gini, entropy, and log loss were used, and for max depth (p_{c_7}), five values from one to five were used.

The variables p_1 to p_8, which serve as hyperparameters for texture feature extraction, are fixed and independent of the model design. In contrast, the hyperparameters $p_{c_1}, p_{c_2}, \ldots, p_{c_7}$ for FS and classification models are determined by the model vector $\boldsymbol{m}$ and vary depending on its elements. For example, if $\boldsymbol{m} = \begin{bmatrix}\text{AF} & \text{None} & \text{SVM with LK}\end{bmatrix}^\top$, then p_{c_3} is adopted as the hyperparameter of the classification model, resulting in $c_3 = 9$. That is, the hyperparameter $\boldsymbol{p}$ is nine-dimensional. For $\boldsymbol{m} = \begin{bmatrix}\text{FS} & \text{DC} & \text{DT}\end{bmatrix}^\top$, p_{c_2} is used as the hyperparameter for FS, and p_{c_6} and p_{c_7} are used as the hyperparameters of the classification model, resulting in $c_2 = 9$, $c_6 = 10$, and $c_7 = 11$. In other words, the hyperparameter $\boldsymbol{p}$ is 11-dimensional.

The hyperparameter vector thus defined is then optimized using Bayesian optimization. In this experiment, the number of iterations in the Bayesian optimization B was set as 300.

3.4. Experimental procedure using the Algorithms

In this experiment, the mean and standard deviation s_j^{mean}, s_j^{std}, $j \in \{\text{train, valid, test}\}$, of the macro-average F1-score for each dataset were calculated using dataset D, model vector $\boldsymbol{m}$, number of cross-validation splits K, and iterations in Bayesian optimization B as the inputs. First, set J is defined to represent each dataset, and an empty set S_j, $j \in J$, is defined to store the output evaluation values. Dataset D is partitioned into subsets $D_1, D_2, \cdots, D_K$ based on the stratified K-fold cross-validation (Equation 3.7). One block of K partitions is then assigned to the test set D_{test} and the remaining $K-1$ blocks are assigned to the training set D_{train}. Then, the SearchBestParameters function is executed to search for hyperparameters suitable for the dataset and model design (Algorithm 1: Lines 1–7).

In the SearchBestParameters function in Algorithm 2, given input D_{train}, K, B, and $\boldsymbol{m}$, it returns the score $S_{\text{valid}}^{\max}$ and the hyperparameter $\boldsymbol{p}^*$ when the average generalization performance on the validation set was the highest, using a combined cross-validation and hyperparameter optimization method. Here, the training set D_{train} is first partitioned into the subsets $D_1, D_2, \cdots, D_K$ based on stratified K-fold cross-validation (Equation 3.7). Subsequently, an empty set S_{valid} is declared to store the validation score obtained in one optimization run. Furthermore, a hyperparameter vector $\boldsymbol{p}$ is constructed from the search range listed in Table 2 using Bayesian optimization according to Equation 3.9 (Algorithm 2: Lines 1–4). Next, in the K subsets, one block is assigned to the validation set D_{valid} and the remaining $K-1$ blocks were assigned to the split training set D_{st}. Texture feature extraction is then performed on D_{st} and D_{valid} based on hyperparameter vector $\boldsymbol{p}$. In the texture feature extraction, 39 features were

first extracted, and then robust standardization was applied. The median and interquartile range were calculated using F_{st} and robust standardization was applied to F_{st} and F_{valid} based on these values (Algorithm 2: Lines 6–8).

For dimensionality reduction, FS, DC, and both are applied or not, depending on the model design (Algorithm 2: Line 9). Therefore, in the DimensionalityReduction function in Algorithm 3, the dimensionality reduction method was selected based on whether the model vector $\boldsymbol{m}$ contains FS and/or DC parameters. If DC is applied, the transformation parameters are determined according to the sample distribution of the input data. Thus, F_{st} is compressed first, followed by F_{valid} using the transformation parameters calculated from F_{st}. If the model vector $\boldsymbol{m}$ does not contain FS and DC, dimensionality reduction is not applied and the feature inputs to the DimensionalityReduction function are returned as they are (Algorithm 3: Lines 1–7).

Based on the feature vectors F_{st} and F_{valid} returned by the DimensionalityReduction function, $\boldsymbol{m}$ indicating the classification model to be applied, and its hyperparameter $\boldsymbol{p}$, the evaluation value s_{valid} for the validation set is calculated. Subsequently, K instances of s_{valid} are merged into S_{valid}. The mean values $s^{\mathrm{mean}}_{\mathrm{valid},\,b}$ and their hyperparameter $\boldsymbol{p}_b$ are stored as the numerical value at the bth Bayesian optimization iteration (Algorithm 2: Lines 10–14). When the procedure up to this point is performed B times, hyperparameter b^* is extracted when $s^{\mathrm{mean}}_{\mathrm{valid},\,b}$ is the largest. A validation score $s^{\mathrm{mean}}_{\mathrm{valid},\,b^*}$ is adopted as the maximum score $S^{\mathrm{max}}_{\mathrm{valid}}$. Furthermore, the hyperparameter $\boldsymbol{p}_{b^*}$ at the b^*-th iteration is adopted as the optimized hyperparameter $\boldsymbol{p}^*$ (Algorithm 2: Lines 16–19).

Extraction of texture features and dimensionality reduction and calculation of the classification scores for the training set D_{train} and test set D_{test} in Algorithm 1 were performed using the optimized hyperparameters $\boldsymbol{p}^*$. This procedure follows lines 8–11 of Algorithm 2. Consequently, sets of evaluation metrics, S_{train}, S_{valid}, and S_{test} were obtained for the training, validation, and test sets, respectively. Subsequently, the mean and standard deviation of these metrics were returned as the prediction performance for a given model design $\boldsymbol{m}$ (Algorithm 1: Lines 8–15).

4. Results and discussion

4.1. Effectiveness of texture features for myocardial pathology diagnosis

First, the effectiveness of texture features in diagnosing myocardial pathology was evaluated for the disease, borderline, and normal case categories. Thus, the set of optimal hyperparameters $\boldsymbol{p}^*$ ($|M| \times K = 60$), employed in each cross-validation for all model designs, M, was collected. The most frequently adopted values for each element were aggregated, and texture features were extracted using the reconstructed hyperparameter vector. Statistical hypothesis testing was conducted to examine whether texture features exhibited distinct class distributions in one or more diseased, borderline, or normal cases. The statistical hypothesis testing methods employed were the Kruskal-Wallis test, a nonparametric test designed for three or more groups, and the Benjamini-Hochberg method, a multiple-testing correction. $\alpha = 0.05$ was set as the significance level. Box plots of these texture features and their adjusted p-values are shown in Figure 4.

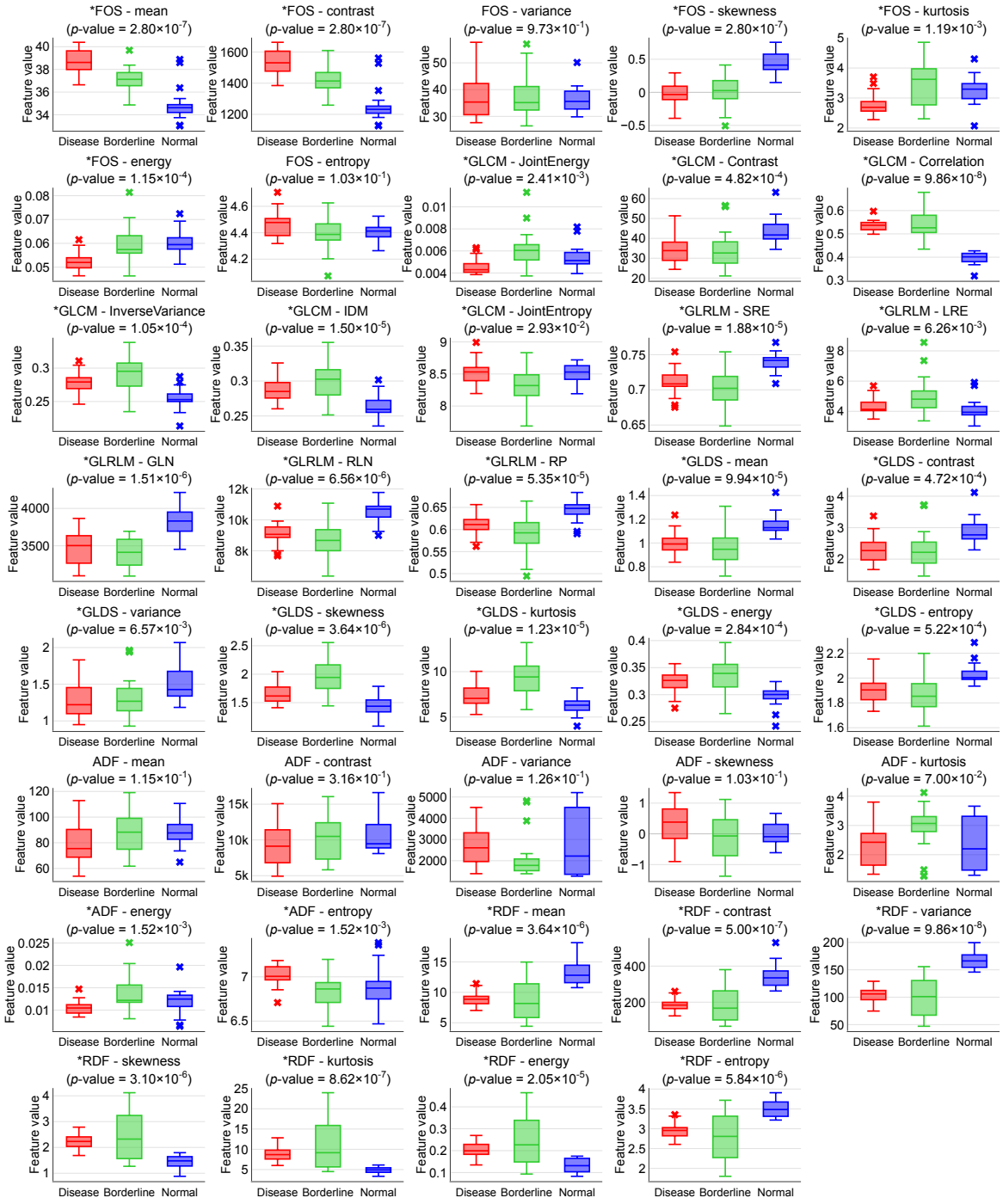


Figure 4. Box plots for disease, borderline, and normal cases across all texture features, and adjusted p-values for the three groups from statistical hypothesis testing (features with significant differences are marked with an asterisk beside the variable name).

Here, it was qualitatively confirmed that there exist many features in which one class is clearly distinguishable from the others. For example, this includes the mean of the RDF, which quantifies the periodicity of an image, and the IDM of the GLCM, which quantifies image uniformity and smoothness. The mean of the RDF was higher in the normal cases compared to the disease and

borderline cases, indicating that normal cases exhibit stronger high-frequency components and more fine-scale repetition than the others. In contrast, the IDM of the GLCM was lower in the normal cases than in the disease and borderline cases, suggesting that normal cases tend to exhibit sharper intensity transitions between adjacent pixels, indicating more abrupt tonal changes rather than gradual variations. Among the extracted features, the mean of FOS showed relatively clear separation among the three classes. This metric represents the centroid of the gray level distribution in an image, and distinct trends were observed across the classes. Specifically, in disease cases where the median of the box plot was the highest, the image shows a stronger prevalence of whites compared to the other classes. The borderline case showed a slightly lower median, whereas the normal case, with the lowest median, demonstrated a stronger presence of black across the entire image. These trends are also illustrated in Figure 3. In normal cases, cardiomyocytes are arranged in parallel and orderly fashion, with minimal collagen or fibrous tissue present in the myocardium [42]. In contrast, hypertrophic cardiomyopathy exhibits cardiomyocyte disarray and diffuse fibrosis throughout the myocardium [43]. In this study, disease cases were defined by the presence of cardiomyocyte disarray, and borderline cases by its absence. Accordingly, the observed trends—the highest mean of the RDF and lowest IDM of the GLCM in normal cases, and a decreasing mean of FOS from disease to borderline to normal—were consistent with established histopathological findings, suggesting the potential clinical relevance of the extracted texture features.

More than half of the features exhibited statistically significant differences, with significance confirmed for 32 of the 39 ($\approx$ 82.1%) features, excluding the variance and entropy of FOS and the mean, contrast, variance, skewness, and kurtosis of the ADF. This suggests that texture features have the potential to extract distinct trends in the three myocardial conditions. However, for many statistical metrics of the ADF, no significant differences were observed in the class distributions. Therefore, it is suggested that considering periodicity with respect to angular directionality may not be effective for the diagnosis of myocardial pathology. This may be due to a lack of uniformity in the fiber direction of the myocardial cells during image acquisition (Figure 3). Thus, although the cost of data preparation is high, setting rules for the fiber orientation during image acquisition could potentially yield significant differences in the statistical metrics related to the ADF.

4.2. Performance comparison of different model designs

Next, the prediction performance of each model design M is compared to identify the model design that is most suitable for the pathological diagnosis of cardiomyopathy using a small sample size. The number of spatial dimensions after dimensionality reduction and the macro-average F1-score for each dataset for each model design are summarized in Table 3.

Initially, we focused on overall evaluation values: the minimum was 0.983 and the maximum was 1.000 for the training set, the minimum was 0.722 and the maximum was 0.906 for the validation set, and the minimum was 0.742 and the maximum was 0.949 for the test set. From these results, it was confirmed that in all model designs, good overall prediction performance was achieved. However, when focusing on the DC method, it was observed that in all the models, the fitting performance of the training set was 1.000, while the generalization performance on the validation set ranged from 0.722 to 0.789, and the generalization performance on the test set ranged from 0.742 to 0.796, indicating a tendency toward overfitting. This suggests that the LDA, being a supervised dimensionality-reduction method, may have resulted in overfitting during DC. This was not attributed to features that could be

noise and did not contribute to the classification, as indicated by the results of the statistical hypothesis tests, but a rather high spatial dimensionality relative to the sample size. A known limitation of classical LDA is that, as the number of features increases relative to the sample size, it becomes difficult to reliably estimate the within-class covariance, which has been reported as a cause of overfitting [44]. In this experiment, cross-validation was applied to the dataset; therefore, the split training set comprised 38 samples. In contrast, the number of features was 39, resulting in high-dimensional, low-sample size data (sample size $\ll$ features) ($38/39 \approx 0.97$). Therefore, it is possible that the distance calculation between samples did not function properly. To address this issue, combining unsupervised dimensionality reduction methods, such as principal component analysis [45] or applying a regularized LDA, which has been extended to high-dimensional low-sample size data [46], is considered necessary.

Table 3. Macro-average F1-score (mean ± 1 standard deviation) for each dataset and classification model obtained using the dimensionality reduction algorithm, and the average feature dimension after dimensionality reduction.

Classification models[1]	Dimensionality reduction[2]	Feature dimension[3]	Macro-average F1-score[4]		
			Training set	Validation set	Test set
SVM with LK	AF	39.0	.992 ± .017	.887 ± .117	.916 ± .053
	FS	$39.0 \xrightarrow{FS} 32.4$	.992 ± .017	.880 ± .113	.867 ± .040
	DC	$39.0 \xrightarrow{DC} 2.0$	**1.00 ± .000**	.789 ± .135	.796 ± .036
	FS + DC	$39.0 \xrightarrow{FS} 18.4 \xrightarrow{DC} 2.0$	.996 ± .008	.895 ± .104	**.949 ± .067**
SVM with RBF	AF	39.0	.992 ± .017	**.906 ± .076**	.898 ± .084
	FS	$39.0 \xrightarrow{FS} 34.0$	.996 ± .008	.900 ± .131	.882 ± .041
	DC	$39.0 \xrightarrow{DC} 2.0$	**1.00 ± .000**	.750 ± .186	.768 ± .108
	FS + DC	$39.0 \xrightarrow{FS} 18.2 \xrightarrow{DC} 2.0$	.992 ± .010	.882 ± .089	.932 ± .034
Decision tree	AF	39.0	.996 ± .008	.806 ± .123	.774 ± .088
	FS	$39.0 \xrightarrow{FS} 36.2$	.983 ± .024	.811 ± .141	.754 ± .090
	DC	$39.0 \xrightarrow{DC} 2.0$	**1.00 ± .000**	.722 ± .183	.742 ± .106
	FS + DC	$39.0 \xrightarrow{FS} 18.2 \xrightarrow{DC} 2.0$	.996 ± .008	.782 ± .119	.811 ± .108

[1] SVM stands for support vector machine. LK stands for linear kernel. RBF stands for radius basis function kernel.
[2] AF stands for all features. FS stands for feature selection. DC stands for dimensional compression.
[3] This presents the average number of dimensions after dimensionality reduction across five-fold cross-validation. The leftmost column lists the number of dimensions at input, while the columns to the right display the dimensions after dimensionality reduction (FS or/and DC).
[4] The underlined values indicate the highest classification performance within each classification model for each data set. **The bold values** indicate that the classification performance is the highest among all classification models for each data set.

Next, focusing on evaluation values other than the DC, SVM consistently exhibited better generalization performance than DT across all dimensionality reduction methods. Within each model, a comparison of the generalization performance on the validation set revealed the following: AF, FS, and FS + DC exhibited similar performances based on their standard deviations. However, on the test set, FS + DC achieved the highest generalization performance among all models, followed by AF in second place and FS in third place.

Generally, when the number of features is large relative to the sample size, overfitting is more likely to occur, which often results in a lower generalization performance [33]. Therefore, in this experiment, AF was expected to exhibit a relatively lower generalization performance. However, as illustrated in Figure 4 and described in Section 4.1, 32 of the 39 ($\approx$ 82.1%) features were considered useful for classification, indicating that the classes were likely separated in a high-dimensional space. In addition, this study employed a nested structure for cross-validation and hyperparameter optimization, where the training and test sets as well as the split training and validation sets were hierarchically separated. Nested cross-validation has been reported to be more robust against overfitting in small sample size datasets compared with conventional cross-validation [20]. These findings indicate that the extraction of highly predictive features, along with the use of cross-validation and hyperparameter optimization methods, which are robust under small sample size conditions, enabled AF to achieve high generalization performance while avoiding overfitting, despite the large number of features relative to the sample size.

Focusing on the feature dimensions in FS, the number of dimensions after FS was 32.4 for SVM with LK, 34.0 for SVM with RBF, and 36.2 for the DT, indicating that only features that could act as noise without contributing to the classification were eliminated. However, cases in which FS outperformed AF in terms of generalization performance were rarely observed. Thus, it can be inferred that when most features are useful, the spatial dimensionality was not notably reduced through FS. The removal of features that can act as noise without contributing to the classification has minimal impact on improving the generalization performance.

However, focusing on the feature dimensions of FS+DC, it was confirmed that after FS, the number of dimensions was approximately 18 for all models. As described in Section 4.1 and shown in Figure 4, it was recognized that 32 of the 39 features were useful for classification. This indicates that not only features that may become noise and contribute little to classification, but also features that were important for classification, have been excluded. In contrast, when focusing on the evaluation values of the test set, it was confirmed that FS+DC had the highest generalization performance across all the models. The number of dimensions was approximately 18, which can be interpreted as the dimension where LDA can avoid overfitting and demonstrate its true discriminative power when the number of features was slightly larger than the sample size ($38/18 \approx 2.11$). Furthermore, the fact that the number of dimensions after applying FS is similar across all models indicates that this is not model-dependent but rather specific to the dataset. Thus, we demonstrated that the multi-step dimensionality-reduction process is effective when the number of features exceeded the sample size. Moreover, this dimensionality-reduction method may be effective regardless of the decision boundary, particularly when the complexity of the model is low.

Table 4 lists the features selected through FS and their proportions based on five-fold cross-validation of the training and test sets for $\begin{bmatrix}\text{FS} & \text{DC} & \text{SVM with LK}\end{bmatrix}^{\top}$, which exhibited the highest generalization performance for the test set. By reviewing Table 4, it was confirmed that many features, which showed significant differences as presented in Figure 4, were selected in more than three out of the five cross-validation runs. Although the GLCM exhibited significant differences across all statistics, it was confirmed that this was not selected for any of the five cross-validation runs. In contrast, although the entropy of FOS and the mean, variance, skewness, and kurtosis of the ADF did not show significant differences in Figure 4, they were selected at least once during the cross-validation. This can be attributed to the stability of the FS algorithm. Kalousis et al. [47] defined

the stability of an FS method as "the robustness of the feature preferences it produces to differences in training sets drawn from the same generating distribution." Dernoncourt et al. [48] reported that in the case of small sample size data, the stability of the FS method decreases, resulting in a higher likelihood of selecting irrelevant features. In addition, they demonstrated that even with appropriate FS methods, the probability of selecting optimal features remains significantly low. Therefore, to achieve higher generalization performance and more reliable factor analysis, it is essential to address the stability of the FS methods. One possible approach would be to utilize domain knowledge [49]. This approach involves preserving potentially interesting relationships while excluding FS configurations that contradict domain knowledge. Thus, irrelevant features may be excluded, potentially improving the stability of FS.

Table 4. Percentage of features selected through feature selection and dimensional compression using an SVM with a linear kernel, based on five-fold stratified cross-validation (training–test split) performed for feature selection.

Statistical measures[1]	Texture analysis methods[2, 3]					
	FOS	GLDS	GLCM	GLRLM	ADF	RDF
Mean	***1.0**	*0.0	—	—	0.2	***0.8**
Contrast	***1.0**	*0.0	*0.0	—	0.0	***1.0**
Variance	0.0	*0.0	—	—	0.2	***1.0**
Skewness	***1.0**	***0.8**	—	—	0.2	***0.8**
Kurtosis	***1.0**	***0.8**	—	—	**0.8**	***1.0**
Energy	***0.8**	*0.2	—	—	***1.0**	*0.2
Entropy	0.2	*0.0	—	—	***0.8**	***0.6**
Correlation	—	—	*0.0	—	—	—
Joint energy	—	—	*0.0	—	—	—
Joint entropy	—	—	*0.0	—	—	—
IDM	—	—	*0.0	—	—	—
Inverse variance	—	—	*0.0	—	—	—
SRE	—	—	—	***0.6**	—	—
LRE	—	—	—	*0.2	—	—
GLN	—	—	—	***0.8**	—	—
RLN	—	—	—	***0.8**	—	—
RP	—	—	—	***0.6**	—	—

[1] IDM stands for inverse difference moment. SRE stands for short run emphasis. LRE stands for long run emphasis. GLN stands for gray level non-uniformity. RLN stands for run length non-uniformity. RP stands for run percentage.

[2] FOS stands for first-order statistics. GLDS stands for gray level difference statistics. GLCM stands for gray level co-occurrence matrix. GLRLM stands for gray level run length matrix. ADF stands for angular distribution function based on the Fourier power spectrum. RDF stands for radial distribution function based on the Fourier power spectrum.

[3] **Bold text** indicates that a feature was selected in more than three of the iterations during the five-fold stratified cross-validation. The asterisk (*) indicates that the feature is a characteristic for which a significant difference was confirmed through statistical hypothesis testing in Section 4.1.

5. Conclusions

The goal of this study was to develop a pathological diagnostic model for cardiomyopathy with high generalization performance, even in environments where collecting diverse and large sample sizes of endomyocardial biopsy specimens is difficult. To achieve this, the effectiveness of texture features in the pathological diagnosis of cardiomyopathy was examined, and a model design suitable for small sample size data was assessed.

Regarding texture features, although several features clearly distinguished all three classes, many features were observed to distinctly separate one class from the others, as demonstrated qualitatively. Additionally, the Kruskal-Wallis test and Benjamini-Hochberg method indicated that 32 out of the 39 texture features were useful for classification. Regarding the model design, whether using classification models or dimensionality reduction methods, good predictive performance was achieved for the pathological diagnosis of the three myocardial states. However, when applying DC alone, a discrepancy of more than 0.2 points between the training and test set evaluation values was observed, indicating an overfitting tendency. In addition, when applying FS alone, only a single case of improvement in the generalization performance was observed on the validation or test sets compared to models without dimensionality reduction. Conversely, the combination of FS and DC demonstrated the best generalization performance for the test set, outperforming both models without dimensionality reduction and those with single-dimensionality reduction methods. These trends were consistent across all the classification models used in this study.

These experimental results suggest that even when the ratio of features to sample size is high, most features remain useful for classification, and the application of nested cross-validation and hyperparameter optimization, which are robust to small sample size data, may mitigate overfitting and enable high predictive performance. Moreover, if most features are useful and FS does not substantially reduce the feature space, applying FS does not contribute significantly to improving generalization performance. Furthermore, when the ratio of features to sample size is high, a multistage dimensionality reduction process may prove effective. This process is likely to be effective, regardless of whether the decision boundary is linear, curvilinear, or vertical/horizontal to the axes, provided that the complexity of the model is controlled. When these conditions were satisfied, the generalization performance of the SVM with LK was the highest, achieving a macro-average F1-score of 0.949 in the test set.

These findings are expected to help solve various issues related to the shortage of pathology experts in cardiology by developing a pathology diagnostic model for cardiomyopathy with a high generalization performance. Furthermore, the model design may be applicable to other diseases with insufficient data sizes, potentially contributing to the rapid adoption of ML models in medical practice.

This study has three main limitations. First, generalizability is constrained by the small sample size and the fact that data were collected from a limited number of institutions. In this study, a model design capable of achieving high predictive performance on small-sample data was evaluated, and texture features were comprehensively assessed. However, these results and interpretations rely on a total of 60 image samples collected from six subjects (ten images per subject). In light of this, the observed predictive performance and insights into texture features may be specific to the limited dataset used and do not guarantee comparable performance on unseen data. Moreover, as all images were acquired using the same equipment, magnification, and staining conditions, and color variation due to acquisition

settings was considered minimal, no stain normalization was applied. Nonetheless, in histopathological image analysis, stain normalization methods such as Reinhard, Macenko, or Vahadane are typically employed to suppress color variation [50]. Accordingly, to improve generalizability, it is essential to include data from subjects with varied attributes (e.g., age and sex) and to apply stain-normalization methods in order to ensure both sufficient diversity and consistency within the dataset. Second, it does not consider individual differences inherent in medical data. In general, medical data contain several unobservable individual differences, such as lifestyle factors, genetic predisposition, environmental influences, psychological aspects, and socioeconomic conditions, which are known to degrade the generalization performance [51]. Therefore, it is essential to consider the effects of these individual differences when applying dimensionality reduction or hyperparameter optimization. However, in this study, only two subjects were included per case, making it impossible to account for individual differences. Thus, it is necessary to investigate the extent to which texture features and model designs evaluated in this study are effective for unseen individual differences. Third, the need to explore methods more specifically tailored to small sample size data should be considered. Although this study achieved a high generalization performance with a macro-average F1-score of 0.949 on the test set, certain aspects remain unaddressed. Specifically, the stability of the FS methods has not yet been examined [47, 48], and the application of dimensionality reduction methods tailored to high-dimensional low-sample size data have not yet been explored [45, 46]. Addressing these aspects is indispensable for achieving further improvements in generalization performance and effectiveness. Therefore, these issues remain important subjects for future research.

Acknowledgments

This work was supported in part by the Japan Society for the Promotion of Science (JSPS) Grants-in-Aid for Scientific Research (C) (Grant Nos. 23K11310 and 21K04535), and by Nihon University Research Grant for 2023 [23-14].

Conflict of 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.

Author contributions

Masaya Mori: Formal analysis, Investigation, Methodology, Software, Visualization, Writing – original draft. Yuto Omae: Formal analysis, Funding acquisition, Investigation, Methodology, Supervision, Validation, Writing – review & editing. Yutaka Koyama: Conceptualization, Data curation, Investigation, Resources, Writing – review & editing. Kazuyuki Hara: Methodology, Validation, Writing – review & editing. Jun Toyotani: Methodology, Project administration, Supervision, Validation, Writing – review & editing. Yasuo Okumura: Conceptualization, Data curation, Investigation, Resources, Writing – review & editing. Hiroyuki Hao: Conceptualization, Data curation, Investigation, Project administration, Resources, Supervision, Writing – review & editing.

References

1. Ishibashi-Ueda H, Matsuyama TA, Ohta-Ogo K, Ikeda Y (2017) Significance and value of endomyocardial biopsy based on our own experience. *Circ J* 81: 417–426. https://doi.org/10.1253/circj.CJ-16-0927
2. Leone O, Veinot JP, Angelini A, Baandrup UT, Basso C, Berry G, et al. (2012) 2011 consensus statement on endomyocardial biopsy from the association for european cardiovascular pathology and the society for cardiovascular pathology. *Cardiovasc Pathol* 21: 245–274. https://doi.org/10.1016/j.carpath.2011.10.001
3. Cooper LT, Baughman KL, Feldman AM, Frustaci A, Jessup M, Kuhl U, et al. (2007) The role of endomyocardial biopsy in the management of cardiovascular disease: a scientific statement from the american heart association, the american college of cardiology, and the european society of cardiology. *Circulation* 116: 2216–2233. https://doi.org/10.1161/CIRCULATIONAHA.107.186093
4. Nirschl JJ, Janowczyk A, Peyster EG, Frank R, Margulies KB, Feldman MD, et al. (2018) A deep-learning classifier identifies patients with clinical heart failure using whole-slide images of H&E tissue. *PloS one* 13: e0192726. https://doi.org/10.1371/journal.pone.0192726
5. Pallua JD, Brunner A, Zelger B, Schirmer M, Haybaeck J (2020) The future of pathology is digital. *Pathol-Res Pract* 216: 153040. https://doi.org/10.1016/j.prp.2020.153040
6. Cai L, Gao J, Zhao D (2020) A review of the application of deep learning in medical image classification and segmentation. *Ann TransL Med* 8: 713. https://doi.org/10.21037/atm.2020.02.44
7. Li M, Zhang Y (2023) Medical image analysis using deep learning algorithms. *Front Public Health* 11: 1273253. https://doi.org/10.3389/fpubh.2023.1273253
8. Liu YH (2018) Feature extraction and image recognition with convolutional neural networks, in: *Journal of Physics: Conference Series*, IOP Publishing, 1087: 062032. https://doi.org/10.1088/1742-6596/1087/6/062032
9. Ergun H, Akyuz YC, Sert M, Liu J (2016) Early and late level fusion of deep convolutional neural networks for visual concept recognition. *Int J Semant Comput* 10: 379–397. https://doi.org/10.1142/S1793351X16400158
10. Lokesh S, Priya A, Sakhare DT, Devi RM, Sahu DN, Reddy PCS (2016) CNN based deep learning methods for precise analysis of cardiac arrhythmias. *Int J Health Sci* 6: 10808–10819. https://doi.org/10.53730/ijhs.v6nS1.7596
11. Han SS, Park GH, Lim W, Kim MS, Na JI, Park I, et al. (2018) Deep neural networks show an equivalent and often superior performance to dermatologists in onychomycosis diagnosis: Automatic construction of onychomycosis datasets by region-based convolutional deep neural network. *PloS one* 13: e0191493. https://doi.org/10.1371/journal.pone.0191493
12. Nagpal K, Foote D, Liu Y, Chen PHC, Wulczyn E, Tan F, et al. (2019) Development and validation of a deep learning algorithm for improving gleason scoring of prostate cancer. *NPJ Digit Med* 2: 48. https://doi.org/10.1038/s41746-019-0112-2
13. Khalid H, Hussain M, Al Ghamdi MA, Khalid T, Khalid K, Khan MA, et al. (2020) A comparative systematic literature review on knee bone reports from mri, x-rays and

ct scans using deep learning and machine learning methodologies. *Diagnostics* 10: 518. https://doi.org/10.3390/diagnostics10080518

14. Wu JX, Pai CC, Kan CD, Chen PY, Chen WL, Lin CH (2022) Chest x-ray image analysis with combining 2d and 1d convolutional neural network based classifier for rapid cardiomegaly screening. *IEEE Access* 10: 47824–47836. https://doi.org/10.1109/ACCESS.2022.3171811
15. Sharifrazi D, Alizadehsani R, Joloudari JH, Band SS, Hussain S, Sani ZA, et al. (2022) CNN-KCL: automatic myocarditis diagnosis using convolutional neural network combined with k-means clustering. *MBE* 19: 2381–2402. https://doi.org/10.3934/MBE.2022110
16. Aromiwura AA, Settle T, Umer M, Joshi J, Shotwell M, Mattumpuram J, et al. (2023) Artificial intelligence in cardiac computed tomography. *Prog Cardiovasc Dis* 81: 54–77. https://doi.org/10.1016/j.pcad.2023.09.001
17. From AM, Maleszewski JJ, Rihal CS (2011) in: *Current status of endomyocardial biopsy*, Mayo Clinic Proceedings, Elsevier, 86: 1095–1102. https://doi.org/10.4065/mcp.2011.0296
18. Tong L, Hoffman R, Deshpande SR, Wang MD (2017) Predicting heart rejection using histopathological whole-slide imaging and deep neural network with dropout, in *2017 IEEE EMBS International Conference on Biomedical & Health Informatics (BHI)*, IEEE, 2017: 1–4. https://doi.org/10.1109/BHI.2017.7897190
19. Dooley AE, Tong L, Deshpande SR, Wang MD (2018) Prediction of heart transplant rejection using histopathological whole-slide imaging, in *2018 IEEE EMBS International Conference on Biomedical & Health Informatics (BHI)*, IEEE, 2018: 251–254. https://doi.org/10.1109/BHI.2018.8333416
20. Vabalas A, Gowen E, Poliakoff E, Casson AJ (2019) Machine learning algorithm validation with a limited sample size, *PloS one* 14: e0224365. https://doi.org/10.1371/journal.pone.0224365
21. Porumb M, Iadanza E, Massaro S, Pecchia L (2020) A convolutional neural network approach to detect congestive heart failure. *Biomed Signal Process Control* 55: 101597. https://doi.org/10.1016/j.bspc.2019.101597
22. Yildiz A, Zan H, Said S (2021) Classification and analysis of epileptic eeg recordings using convolutional neural network and class activation mapping. *Biomed signal process control* 68: 102720. https://doi.org/10.1016/j.bspc.2021.102720
23. Pikulkaew K (2023) Enhancing brain tumor detection with gradient-weighted class activation mapping and deep learning techniques, in: *2023 20th International Joint Conference on Computer Science and Software Engineering (JCSSE)*, IEEE, 2023: 339–344. https://doi.org/10.1109/JCSSE58229.2023.10202020
24. Zhou B, Khosla A, Lapedriza A, Oliva A, Torralba A (2016) Learning deep features for discriminative localization, in: *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016: 2921–2929. https://doi.org/10.1109/CVPR.2016.319
25. Amato D, Calderaro S, Lo Bosco G, Rizzo R, Vella F (2024) Explainable histopathology image classification with self-organizing maps: a granular computing perspective. *Cogn Comput* 16: 2999-3019. https://doi.org/10.1007/s12559-024-10312-1

26. Kumar KK, Chaduvula K, Markapudi B (2020) A detailed survey on feature extraction techniques in image processing for medical image analysis. *Eur J Mol Clin Med* 7: 2275–2284.

27. Shirani J, Pick R, Roberts WC, Maron BJ (2000) Morphology and significance of the left ventricular collagen network in young patients with hypertrophic cardiomyopathy and sudden cardiac death. *J Am Coll Cardiol* 35: 36–44. https://doi.org/10.1016/S0735-1097(99)00492-1

28. Cianci V, Forzese E, Sapienza D, Cardia L, Cianci A, Germanà A, et al. (2000) Morphological and genetic aspects for post-mortem diagnosis of hypertrophic cardiomyopathy: a systematic review. *Int J Mol Sci* 25: 1275. https://doi.org/10.3390/ijms25021275

29. Castellano G, Bonilha L, Li LM, Cendes F (2004) Texture analysis of medical images. *Clin radiol* 59: 1061–1069. https://doi.org/10.1016/j.crad.2004.07.008

30. Chowdhary CL, Acharjya DP (2020) Segmentation and feature extraction in medical imaging: a systematic review. *Procedia Comput Sci* 167: 26–36. https://doi.org/10.1016/j.procs.2020.03.179

31. Phan JH, Quo CF, Cheng C, Wang MD (2012) Multiscale integration of-omic, imaging, and clinical data in biomedical informatics. *IEEE Rev Biomed Eng* 5: 74–87. https://doi.org/10.1109/RBME.2012.2212427

32. Guan H, Liu M (2021) Domain adaptation for medical image analysis: a survey. *IEEE T Bio-Med Eng* 69: 1173–1185. https://doi.org/10.1109/TBME.2021.3117407

33. Mori M, Flores RG, Suzuki Y, Nukazawa K, Hiraoka T, Nonaka H (2022) Prediction of microcystis occurrences and analysis using machine learning in high-dimension, low-sample-size and imbalanced water quality data. *Harmful Algae* 117: 102273. https://doi.org/10.1016/j.hal.2022.102273

34. Aliferis C, Simon G (2024) Overfitting, underfitting and general model overconfidence and underperformance pitfalls and best practices in machine learning and ai, in: *Artificial Intelligence and Machine Learning in Health Care and Medical Sciences: Best Practices and Pitfalls*, Springer, 2024: 477–524. https://doi.org/10.1007/978-3-031-39355-6_10

35. Mori M, Omae Y, Koyama Y, et al. (2025) Potential of low-dimensionalized texture features for diagnostic support of cardiomyopathy using endomyocardial biopsy specimens, in: *Springer Proceedings in Mathematics & Statistics*, In press.

36. Van Griethuysen JJ, Fedorov A, Parmar C, Hosny A, Aucoin N, Narayan V, et al. (2017) Computational radiomics system to decode the radiographic phenotype. *Cancer Res* 77: e104–e107. https://doi.org/10.1158/0008-5472.CAN-17-0339

37. Fisher RA (1936) The use of multiple measurements in taxonomic problems, *Annals Eugenics* 7: 179–188. https://doi.org/10.1111/j.1469-1809.1936.tb02137.x

38. Brown MT, Wicker LR (2000) Discriminant analysis, in: *Handbook of applied multivariate statistics and mathematical modeling*, Elsevier, 2000: 209–235. https://doi.org/10.1016/B978-012691360-6/50009-4

39. Buitinck L, Louppe G, Blondel M, Pedregosa F, Mueller A, Grisel O, et al. (2013) API design for machine learning software: experiences from the scikit-learn project. arxiv preprint arxiv: 1309.0238. https://doi.org/10.48550/arXiv.1309.0238

40. Heikonen S, Yli-Heikkilä M, Heino M (2023) Modeling the drivers of eutrophication in finland with a machine learning approach. *Ecosphere* 14: e4522. https://doi.org/10.1002/ecs2.4522
41. Akiba T, Sano S, Yanase T, Ohta T, Koyama M (2019) Optuna: A next-generation hyperparameter optimization framework, in: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. https://doi.org/10.1145/3292500.3330701
42. Kong P, Christia P, Frangogiannis NG (2014) The pathogenesis of cardiac fibrosis. *Cell Mol Life Sci* 71: 549–574. https://doi.org/10.1007/s00018-013-1349-6
43. Kraft T, Montag J, Radocaj A, Brenner B (2016) Hypertrophic cardiomyopathy: cell-to-cell imbalance in gene expression and contraction force as trigger for disease phenotype development. *Circ Res* 119: 992–995. https://doi.org/10.1161/CIRCRESAHA.116.309804
44. Qiao Z, Zhou L, Huang JZ (2008) Effective linear discriminant analysis for high dimensional, low sample size data, in: *Proceeding of the world congress on engineering*, Citeseer, 2008: 2–4.
45. Yang S, Xiong H, Xu K, Wang L, Bian J, Sun Z (2021) Improving covariance-regularized discriminant analysis for ehr-based predictive analytics of diseases. *Appl Intell* 51: 377–395. https://doi.org/10.1007/s10489-020-01810-4
46. Friedman JH (1989) Regularized discriminant analysis. *J Am Stat Assoc* 84: 165–175. https://doi.org/10.1080/01621459.1989.10478752
47. Kalousis A, Prados J, Hilario M (2007) Stability of feature selection algorithms: a study on high-dimensional spaces. *Knowl Inf Syst* 12: 95–116. https://doi.org/10.1007/s10115-006-0040-8
48. Dernoncourt D, Hanczar B, ZuckerJD (2014) Analysis of feature selection stability on high dimension and small sample data. *Comput Stat Data An* 71: 681–693. https://doi.org/10.1016/j.csda.2013.07.012
49. Grove W (2013) Using domain knowledge to systematically guide feature selection, in: *Twenty-Third International Joint Conference on Artificial Intelligence*, Citeseer, 2013.
50. Vaishnani K, Gohel B, Hati A (2021) Impact of stain normalisation technique on deep learning based nuclei segmentation in histopathological image, in: *2023 International Conference on Advances in Intelligent Computing and Applications (AICAPS)*, IEEE, 2023: 1–4. https://doi.org/10.1109/AICAPS57044.2023.10074363
51. Ding C, Yao T, Wu C, Ni J (2024) Deep learning for personalized electrocardiogram diagnosis: A review. arxiv preprint arxiv:2409.07975. https://doi.org/10.48550/arXiv.2409.07975

AIMS Press