

Article

# Imbalanced Ectopic Beat Classification Using a Low-Memory-Usage CNN LMUEBCNet and Correlation-Based ECG Signal Oversampling

You-Liang Xie <sup>1</sup>  and Che-Wei Lin <sup>1,2,3,4,\*</sup> 

<sup>1</sup> Department of Biomedical Engineering, College of Engineering, National Cheng Kung University, Tainan 701, Taiwan

<sup>2</sup> Medical Device Innovation Center, National Cheng Kung University, Tainan 701, Taiwan

<sup>3</sup> Institute of Gerontology, College of Medicine, National Cheng Kung University, Tainan 701, Taiwan

<sup>4</sup> Institute of Medical Informatics, College of Electrical Engineering and Computer Science, National Cheng Kung University, Tainan 701, Taiwan

\* Correspondence: lincw@mail.ncku.edu.tw

**Abstract:** *Objective:* This study presents a low-memory-usage ectopic beat classification convolutional neural network (CNN) (LMUEBCNet) and a correlation-based oversampling (Corr-OS) method for ectopic beat data augmentation. *Methods:* A LMUEBCNet classifier consists of four VGG-based convolution layers and two fully connected layers with the continuous wavelet transform (CWT) spectrogram of a QRS complex (0.712 s) segment as the input of the LMUEBCNet. A Corr-OS method augmented a synthetic beat using the top  $K$  correlation heartbeat of all mixed subjects for balancing the training set. This study validates data via a 10-fold cross-validation in the following three scenarios: training/testing with native data (CV1), training/testing with augmented data (CV2), and training with augmented data but testing with native data (CV3). *Experiments:* The PhysioNet MIT-BIH arrhythmia ECG database was used for verifying the proposed algorithm. This database consists of a total of 109,443 heartbeats categorized into five classes according to AAMI EC57: non-ectopic beats (N), supraventricular ectopic beats (S), ventricular ectopic beats (V), a fusion of ventricular and normal beats (F), and unknown beats (Q), with 90,586/2781/7236/803/8039 heartbeats, respectively. Three pre-trained CNNs: AlexNet/ResNet18/VGG19 were utilized in this study to compare the ectopic beat classification performance of the LMUEBCNet. The effectiveness of using Corr-OS data augmentation was determined by comparing (1) with/without using the Corr-OS method and (2) the Next-OS data augmentation method. Next-OS augmented the synthetic beat using the next heartbeat of one subject. *Results:* The proposed LMUEBCNet can achieve a 99.4% classification accuracy under the CV2 and CV3 cross-validation scenarios. The accuracy of the proposed LMUEBCNet is 0.4–0.5% less than the performance obtained from AlexNet/ResNet18/VGG19 under the same data augmentation and cross-validation scenario, but the parameter usage is only 10% or less than that of the AlexNet/ResNet18/VGG19 method. The proposed Corr-OS method can improve ectopic beat classification accuracy by 0.3%. *Conclusion:* This study developed a LMUEBCNet that can achieve a high ectopic beat classification accuracy with efficient parameter usage and utilized the Corr-OS method for balancing datasets to improve the classification performance.

**Keywords:** ectopic beat classification; MIT-BIH; AAMI; low memory usage; convolutional neural network (CNN); data augmentation; signal processing; image processing

**MSC:** 68T07; 92B20; 92C55; 94A12; 94A08; 68U10



**Citation:** Xie, Y.-L.; Lin, C.-W. Imbalanced Ectopic Beat Classification Using a Low-Memory-Usage CNN LMUEBCNet and Correlation-Based ECG Signal Oversampling. *Mathematics* **2023**, *11*, 1833. <https://doi.org/10.3390/math11081833>

Academic Editors: Alvaro Figueira and Francesco Renna

Received: 15 March 2023

Revised: 2 April 2023

Accepted: 4 April 2023

Published: 12 April 2023



**Copyright:** © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Ectopic beats are a type of cardiac arrhythmias, with the excessive supraventricular ectopic activity being correlated with an increased risk of stroke, and the ablation of ventricular ectopic beats has been proven to improve cardiomyopathy [1]. Arrhythmia is caused

by abnormalities of electrical impulses generation and/or conduction and ectopic beats are a type of cardiac arrhythmias [2]. To automatically recognize an ectopic beat (single heart-beat) is essential for arrhythmia diagnosis. Classification of normal/ectopic beat can be categorized into five classes: non-ectopic beats (N), supraventricular ectopic beats (S), ventricular ectopic beats (V), a fusion of ventricular and normal beats (F), and unknown beats (Q), based on the ANSI/AAMI EC57:2012 standard [3]. The ANSI/AAMI EC57 standard is published by Association for the Advancement of Medical Instrumentation (AAMI) and it provides the standard for testing and reporting the performance of the cardiac rhythm analysis algorithm. Ectopic beat classification is one of the testing items. The ECG databases such as the MIT-BIH (Massachusetts Institute of Technology—Boston's Beth Israel Hospital) arrhythmia database and the AHA (American Heart Association) ECG database are recommended by EC57 to be used to evaluate the ectopic beat classification performance.

Machine learning (ML) is good for handling multi-dimensional and multi-variety data and is thus appropriate to process the high-dimensional feature vector extracted from the ECG database to classify ectopic beats. ML classifiers such as the neural network (NN) and support vector machine (SVM) have been found to be effective for ectopic beat detection using morphological and interval-based features. When extracting different features including higher order statistics (HOSs), Gaussian mixture modeling (GMM), or wavelet packet entropy (WPE) from the RR interval and then classifying using decision trees (DTs) or random forest (RF), a 94% accuracy can be obtained in ectopic beat classification [4,5]. A combination of feature engineering techniques such as combining a discrete cosine transform (DCT)/discrete wavelet transform (DWT)/principal component analysis (PCA)/independent component analysis (ICA) to extract features and classify by various classifiers such as a  $k$ -nearest neighbor ( $k$ -NN), NN, SVM, support vector machine and radial basis function (SVM-RBF), and probabilistic neural network (PNN), can achieve accuracies of around 98% to 99% in the classification task [6–8].

Deep learning (DL) outperforms ML in many classification problems due to its ability to execute feature engineering by itself. 1D/2D/3D convolutional neural networks (CNNs) have been employed to classify ectopic beats with respect to ECG expressed in 1D/2D/3D forms [9]. In the time domain 1D signal data, Acharya et al. used a 1D-CNN model with three convolution layers and three fully connected layers, then balanced the dataset using the standard deviation and Z-score, finally achieving 94.03% in overall accuracy [10]. Wang et al. and Romdhane et al. used a self-designed 1D-CNN to classify ectopic beats within a 0.9 s to 10 s time-window and obtained an accuracy over 98% [11,12]. Yao et al. applied a gated recurrent unit (GRU) and six VGG-based local feature extraction modules (LFEM) to a 1D-CNN and achieved an overall accuracy of 99.61% in a training/test ratio of 8/2 [13]. In the 2D CNN for ectopic beat classification, Al Rahhal et al. and Xie et al. utilized the continuous wavelet transform (CWT) and STFT to generate a spectrogram, then applied the pre-trained VGG16 CNN and a self-designed 31-layer CNN classifier and achieved a high accuracy [14,15]. Zhai et al. applied a dual-beat coupling matrix with 73 by 73 pixels for generating features and a CNN that consisted of three convolution layers with two FC layers for classification, finally achieving an overall accuracy of 96.07% (50% training and 50% testing) [16]. Sellami et al. used a nine-layer CNN (one convolution with four residual modules (two convolutions)) to classify a single beat, two beats, and three beats. Next, they applied batch-weighted loss to overcome the imbalance between classes and achieved an overall accuracy of 99.5% and an average sensitivity of 94.7% [17]. For applying a 3D CNN in ectopic beat classification, Li et al. extracted and formulated the three-dimensional features including the single heartbeat segment, RRI ratios, and beat-to-beat correlations, and classified the three-dimensional features using the 3D-CNN, obtaining an overall accuracy of 91.44% (50% training and 50% testing) [18].

The low-memory-usage models such as KecNet/LiteNet are designed for the execution of the deep learning network in the portable device, with satisfactory performance but lower power consumption/memory usage [19,20]. A deeper CNN architecture generally can extract more features, but not all of them can be significantly trained [21,22], and this

might increase the memory usage/complexity/training time [23]. Lu et al. utilized KecNet, a lightweight network that can classify N/S/V/F/Q classes with a 99.31% accuracy. It is only 0.11% slightly lower in accuracy but reduces 80% of the parameter count compared to using GoogleNet [19]. LiteNet developed by He et al. reduces more than half of the parameter count compared to using AlexNet, and only decreases 0.02% of the F1-score in the N/S/V/F/Q classification task [20].

The problem of data imbalances occurred in our daily activities, especially in the physiological signals. The imbalanced dataset makes minority classes easily obtain poor results, since the model usually fits majority classes in training tasks [24–26]. More and more research has been addressing the imbalanced dataset problem using data augmentation methods or oversampling methods [27]. Data imbalance conditions can be found in many arrhythmia databases such as the MIT-BIH arrhythmia ECG database. Non-ectopic beats (N), supraventricular ectopic beats (S), ventricular ectopic beats (V), a fusion of ventricular and normal beats (F), and unknown beats (Q), are 90,586/2781/7236/803/8039, respectively, in the MIT-BIH arrhythmia ECG database. Class N accounted for more than half of the database (82.8%), while S and V are only 2.5% and 6.6%, respectively [16]. The following data augmentation methods are widely used in different studies to solve the data imbalance problem: (1) random oversampling (ROS), (2) random undersampling (RUS), (3) the synthetic minority oversampling technique (SMOTE), (4) cost-sensitive learning, (5) generative adversarial networks (GANs), and (6) augmentation with cropping images [28]. The above methods used for solving the imbalanced dataset problem, in particular ROS and RUS, might cause overfitting and underfitting in the deep learning field with an increase in the minority class and a decrease in the majority class. The GAN algorithm includes a generator network and a discriminator network to balance the dataset with augmented figures. However, GANs might consume enormous computational resources with their two convolutional architectures. Based on the above reasons, the data augmentation method based on the SMOTE becomes the best choice in this study. In the above studies for balancing databases, correlation is hardly taken into consideration [29–33]. The traditional SMOTE algorithm only uses Euclidean distance to find  $k$ -nearest samples, whereas the correlation coefficient is also an important factor in obtaining the nearest samples for generating synthetic data and avoiding some noise or outliers [34–38].

This paper aims to develop a low-memory-usage ectopic beat classification convolutional neural network (LMUEBCNet) which can achieve more than 99% accuracy for discriminating N/S/V/F/Q classes in the MIT-BIH arrhythmia ECG database. The parameter usage in the LMUEBCNet is expected to be less than 10% of the AlexNet/ResNet18/VGG19 method. This study proposed a data-level oversampling (OS) approach called correlation-based oversampling (Corr-OS), which is modified from the SMOTE method and considers the correlation and relevance between ECG heartbeats to deal with data imbalance conditions.

The rest of the paper is organized as follows: Section 2 presents the related works that provide an overview of recent works for ectopic beat classification using low-memory-usage CNNs and oversampling methods. Section 3 describes the methods developed in this paper. Section 4 shows the experimental results. Discussions and conclusions of this work are given in Sections 5 and 6.

## 2. Related Works

### 2.1. Low-Memory-Usage Convolutional Neural Network for Ectopic Beat Classification

Mathunjwa et al. transformed 2 s ECG recordings into recurrence plot image and classified using a self-developed CNN. Noisy and ventricular fibrillation are classified in the first stage and normal AFib/VPC/APC are classified in the second stage. Mathunjwa et al. compared the performance of their self-designed network to ResNet (18/34/50/101/152 layers)/AlexNet/VGG-16/19 and concluded that a deeper CNN is not guaranteed to achieve higher ectopic beat classification accuracy. The customized CNN for ectopic beat classification can not only obtain higher classification accuracy, but also uses smaller memory usage/parameters to do so [21]. Lu et al. developed a KecNet 1-D CNN with

a special sync-conv layer and only three convolution layers to classify N/S/V/F/Q and achieved a 99.31% accuracy. KecNet consumed only around 20% of parameters and only decreased accuracy by 0.1% when compared to using GoogLeNet [19] for ectopic beat classification. He et al. used a LiteNet 1-D CNN with a lighter inception structure and a residual structure to recognize N/S/V/F/Q, achieving an accuracy of 98.8%. LiteNet saved 60% of parameters and only decreased accuracy by 0.2% when compared with GoogLeNet [20]. LiteNet achieves higher classification accuracy than GoogLeNet in ectopic beat classification while consuming less power.

## 2.2. Oversampling Data Augmentation for Ectopic Beat Classification

Bernardo et al. utilized a self-designed 1-D CNN classifier with ROS to balance the dataset and reached an average macro F1-score of 0.98 and an 0.98 F1-score for all classes (N/S/V/F/Q) [39]. Zhang et al. applied ROS and RUS to balance the dataset of the hybrid time-frequency 2-D image with the ResNet-101 CNN classifier; they achieved average accuracies of 99.62% using ROS and 94.57% using ROS with RUS [40]. Acharya et al. balanced the dataset using the standard deviation and Z-score and generated all classes so that they had an equal number to the N class (90,592 images), achieving a significant improvement from 89.03% to 94.03% in overall accuracy using a 1D-CNN [10]. Lu et al. used 25 PQRS samples as features and extracted 200 features from a CNN (input signal images). Then, they applied several balancing methods, such as ROS, RUS, cluster centroids (CC), near miss (NM), edited nearest neighbors (ENNs), repeated edited nearest neighbors (RENNs), the neighborhood cleaning rule (NCR), and a one-sided selection (OSS). Finally, using ROS with a RF classifier obtained a highest accuracy of 99.96% [29]. Mousavi et al. used the SMOTE method to balance the dataset with a CNN auto-encoder and a bidirectional recurrent neural network (BiRNN) to classify the N/S/V/F classes and achieved an accuracy of 99.92% [30]. Ahmad et al. utilized the SMOTE method to oversample S/V/F/Q classes and obtained 30,000/20,000/20,000/10,000 samples, respectively. Next, using a AlexNet CNN and self-designed simpler CNN to extract the Gramian angular field (GAF), recurrence plot (RP), and Markov transition field matrix (MTF) features with the SVM classifier, finally achieved an accuracy of 99.7% [41]. Shaker et al. applied a GAN as the balancing method and a 1-D CNN with three inception modules and three FC layers for classification. They achieved an overall accuracy of above 98.0% and a sensitivity of over 97.7% [31].

## 3. Methods

The proposed ectopic beat classification algorithm consists of windowing processing, oversampling, feature generation, and a LMUEBCNet classifier. The flowchart of the proposed algorithm is shown in Figure 1. The windowing processing segments raw ECG signal into consecutive 0.712 s windows. The minority classes (S/V/F/Q) are augmented via the proposed correlation-based oversampling (Corr-OS) method. Corr-OS is generated by the interpolation of one ECG segment and a segment in the same class with a top  $K$  ( $K = 1\sim 5$ ) high correlation value. The time domain ECG signal is transformed into a time-frequency spectrogram (represented as figures) using the continuous wavelet transform (CWT) in feature generation. The difference between the N class and V/S class can be enhanced with the help of time-frequency transformation. The LMUEBCNet with low memory usage is composed of two convolution layers, two VGG-based convolution blocks, and two fully connected layers.

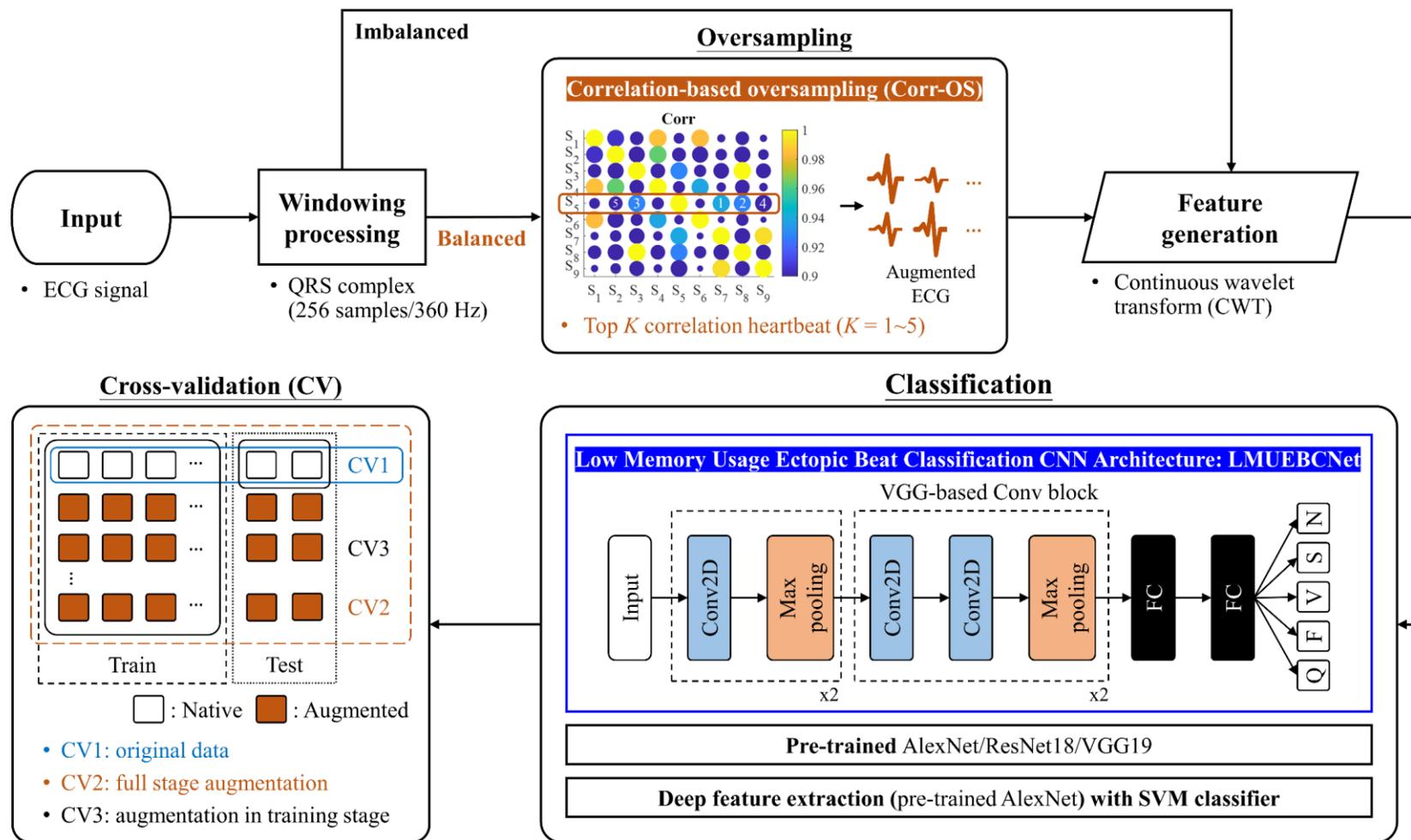


Figure 1. Flowchart of the proposed algorithm.

### 3.1. Windowing Processing

Windowing processing was applied for splitting a continuous ECG signal into ECG windows in order to easily analyze the ectopic beats. Each window segment consists of 0.712 s ECG readings which are centered with the R wave point that is extracted from each QRS complex. This study used the symbols of the MIT-BIH arrhythmia database downloaded from the PhysioBank ATM as the center and extended 0.356 s left/right (127 samples before and 128 samples after the R peak; sampling rate: 360 Hz).

### 3.2. Feature Generation: Continuous Wavelet Transform (CWT)

The continuous wavelet transform (CWT) is used to emphasize the difference between N/S/V/F/Q classes from the point of view of the time–frequency spectrum [42,43]. The computation of the CWT is expressed using Equation (1):

$$C(a, b) = \int_{-\infty}^{\infty} s(t) \frac{1}{\sqrt{a}} \psi\left(\frac{t-b}{a}\right) dt, \quad a \in \mathbb{R}^+ - \{0\}, b \in \mathbb{R}, \quad (1)$$

$$\text{Morlet wavelet: } \psi(x) = e^{-x^2} \cos\left(\pi \sqrt{\frac{2}{\ln 2}} x\right) \quad (2)$$

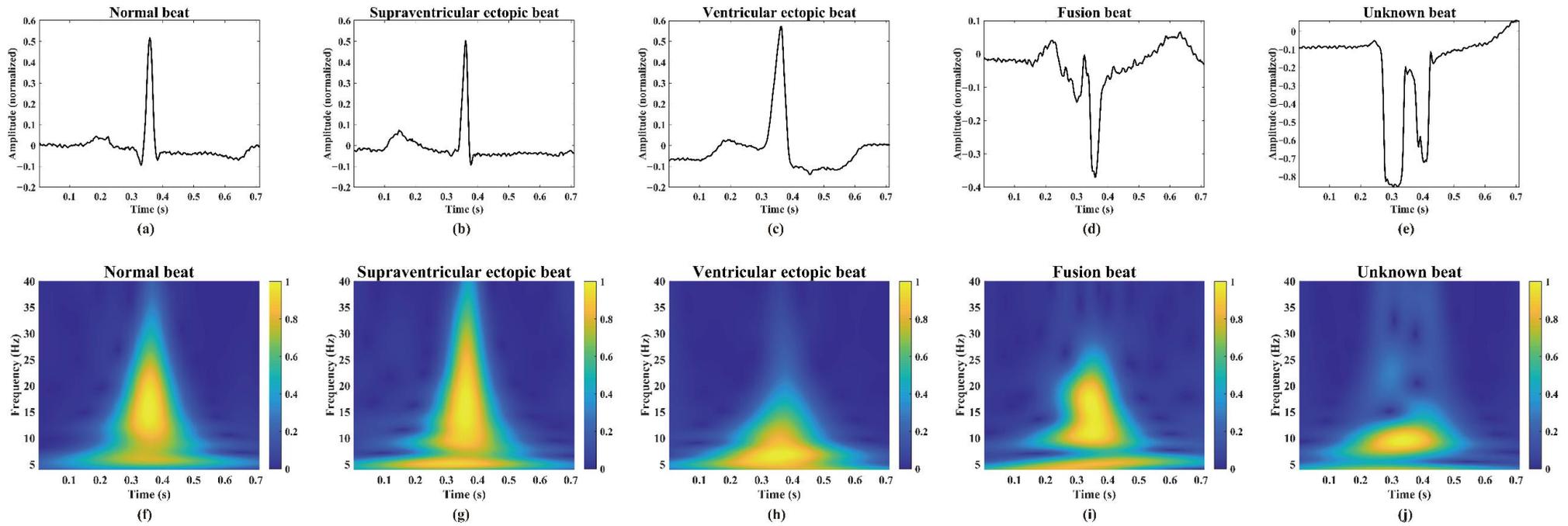
where  $s(t)$  is the signal,  $a$  is the scale,  $b$  is the translation,  $\psi(t)$  is the mother wavelet shown in Equation (2),  $\psi_{a,b}(t)$  is the scaled and translated wavelet, and  $C$  is the 2D matrix of the wavelet coefficients. The Morlet wavelet setting of  $a$  is two divided by the sampling rate (360 Hz), and  $b$  is zero in this study.

Time–frequency analysis is based on the classical Fourier analysis and assumes that signals are infinite in time or periodic, while many signals in practice are of a short duration and change substantially over their duration [4]. The CWT is non-redundant and efficient enough for accurate reconstruction by continuously varying the translation and scaling the parameters of the wavelet. The relationship between scale and frequency in the CWT was also explored as a band-pass filter. Wavelet analysis allows low-frequency information to be more accurate for long intervals and high-frequency information to be more accurate for short intervals. Figure 2 shows the original ECG signals for five classes (N/S/V/F/Q) and the time–frequency spectrogram after transformation.

### 3.3. LMUEBCNet Classifier and Performance Comparison to Existing CNNs

#### 3.3.1. Low-Memory-Usage Ectopic Beat Classification Network: LMUEBCNet

LMUEBCNet is a neural network architecture that comprises of two convolutional layers, two VGG-based convolution blocks, and two fully connected layers. The first convolutional layer uses max pooling to downsample by a stride of 2. The second convolutional layer and two VGG-based convolution blocks are capable of extracting deeper features. Each VGG-based convolution block consists of two convolutional layers with the same padding. To avoid overfitting and reduce the parameters, a dropout of 0.25 is applied after ReLU activation of each convolutional/VGG block, for a total of four times. Additionally, a max pooling with a stride of 2 is applied after each dropout layer for denoising. Finally, all the extracted features are connected using fully connected layers with a softmax activation function.



**Figure 2.** ECG signal of the QRS complex of (a) normal, (b) ventricular ectopic, (c) supraventricular ectopic, (d) fusion, and (e) unknown beats, and the CWT of the QRS complex of (f) normal, (g) ventricular ectopic, (h) supraventricular ectopic, (i) fusion, and (j) unknown beats.

LMUEBCNet’s design incorporates  $3 \times 3$  convolutional layers to achieve high performance while minimizing computational time. Unlike popular architectures such as AlexNet and VGG-16/19 [44], which employ 5 convolutional layers and 3 fully connected layers, this study proposes a new architecture with only 4 convolutional layers and 2 fully connected layers. This architecture reduces the number of parameters and model size. Table 1 displays the proposed LMUEBCNet architecture. The above approach provides a way of designing CNN architectures for embedded systems with limited memory, such as the ARM STM32F7 series, which requires only a maximum flash memory of 1 MB to 2 MB. The last convolutional layer has a maximum of  $7 \times 7 \times 12$  parameters, and the output parameters are set to  $C$  (number of classes)  $\times 16 \times 3$  (number of channels).

**Table 1.** Proposed LMUEBCNet architecture compared with VGG19 [44].

Layer	Filter Numbers		Output Size		Kernel Size	Stride		
	VGG19	LMUEBCNet	VGG19	LMUEBCNet		VGG19	LMUEBCNet	
Input			$224 \times 224 \times 3$		-	-		
1	Convolution	64	6	$224 \times 224 \times 64$	$112 \times 112 \times 6$	$3 \times 3$	1	2
	Max Pooling	64	-	$112 \times 112 \times 64$	$56 \times 56 \times 6$	$2 \times 2$	2	
2	Convolution	128	8	$112 \times 112 \times 128$	$56 \times 56 \times 8$	$3 \times 3$	1	
	Max Pooling	128	-	$56 \times 56 \times 128$	$28 \times 28 \times 8$	$2 \times 2$	2	
3	Convolution	256	12	$56 \times 56 \times 256$	$28 \times 28 \times 12$	$3 \times 3$	1	
	Max Pooling	256	12	$56 \times 56 \times 256$	$28 \times 28 \times 12$	$2 \times 2$	2	
4	Convolution	512	12	$28 \times 28 \times 512$	$14 \times 14 \times 12$	$3 \times 3$	1	
	Max Pooling	512	12	$28 \times 28 \times 512$	$14 \times 14 \times 12$	$2 \times 2$	2	
5	Convolution	512	-	$14 \times 14 \times 512$	-	$3 \times 3$	1	-
	Max Pooling	512	-	$7 \times 7 \times 512$	-	$2 \times 2$	2	-
6	FC	-	-	4096	-	-	-	-
7	FC	-	-	4096	$1 \times 1 \times (C^* \times 48)$	-	-	-
Output	FC	-	-	1000	$C^*$	-	-	-

\* C: number of the class.

### 3.3.2. Performance Comparison to Existing CNNs: Pre-Trained AlexNet/ResNet18/VGG19

This study compared the performance of the proposed LMUEBCNet with existing CNNs by following these techniques: (1) transfer learning using pre-trained AlexNet/ResNet18/VGG19 and (2) deep feature extraction with AlexNet and classification via a SVM classifier. The training process of the CNNs in this research was using the MATLAB® 2019 CNN toolbox [45,46]. The pre-trained CNN models have been trained on approximately 1.2 million images from the ImageNet Dataset [47].

AlexNet [48] is a popular CNN architecture used in computer vision, comprising of five convolutional layers with ReLU or pooling layers, two fully connected layers, and one output (fully connected) layer. The architecture’s design, featuring five convolutional layers and three fully connected layers, has demonstrated high accuracy in image classification tasks on ImageNet. Due to its success, AlexNet has become a popular choice for deep learning researchers who use CNNs and GPUs to accelerate the learning process.

The Residual net (ResNet) [49] series have 18/34/50/101/152 convolution layers in their architectures. This study chose an 18-layer ResNet (ResNet18) to compare the performance of ectopic beat classification. The degradation problem for the increased depth of the network is solved by introducing a deep residual learning framework. The original

mapping will be recast to  $F(x) + x$ , and it can be implemented by using a feedforward neural network with quick connections.

VGG16 and VGG19, developed by the Visual Geometry Group at the University of Oxford [44], are CNN architectures based on AlexNet's five convolutional layers and three fully connected layers. VGG16 has sixteen layers, with thirteen convolutional layers and three fully connected layers, while VGG19 has nineteen layers, with sixteen convolutional layers and three fully connected layers. The deeper architectures of the VGG models results in an increased model weight and computation time. However, for a single crop and similar layers, VGG outperforms AlexNet in prediction accuracy.

Transfer learning is a useful technique where layers from a pre-trained network on a large dataset can be fine-tuned on a new dataset. Fine-tuning the network can be faster and easier than building and training a new network from scratch. The pre-trained network has already learned many image features, but fine-tuning allows it to learn features specific to the new dataset [50]. In this study, pre-trained models such as AlexNet, ResNet18, and VGG19 were utilized with input image sizes of  $227 \times 227 \times 3$  pixels for AlexNet and  $224 \times 224 \times 3$  pixels for ResNet18 and VGG19. The classification output size was set to five classes, and all training curves converged to approximately 100%.

Deep feature extraction with AlexNet and classification via a SVM classifier was also used in this study for evaluating the performance. A pre-trained network can be used as a feature extractor by taking the layer activations as features. Feature extraction is a quick, easy way to take advantage of deep learning without having to spend time and effort training a complete network. The present study used a pre-trained network, AlexNet, as a feature generator to extract the learned image features and used these features to train the support vector machine (SVM) classifier [51]. The trained model has better generalization performance and achieves more advanced classification accuracy compared with other alternatives.

### 3.4. Oversampling Method: Correlation-Based Oversampling (Corr-OS)

This study proposes a novel correlation-based oversampling (Corr-OS) method to augment new beats by identifying the top  $K$  ( $K \in \mathbb{N}$ ) correlation coefficient beats from all beats. In contrast to the traditional synthetic minority oversampling technique (SMOTE) methods that rely on  $k$ -nearest neighbors [52], the proposed method considers the importance of the original signal and identifies the most similar heartbeat based on the correlation coefficient between heartbeats [34,35]. The pseudo-code for the Corr-OS method is presented in Algorithm 1. The method involves collecting all segments of the MIT-BIH arrhythmia database records in the same class, calculating the correlation coefficient between each segment, and selecting the  $K$  highest correlation values. The proposed method then augments the artificial signals by interpolating the target segment with the segment of the highest correlation value and the target segment with the segment of the second-highest correlation value. For example, when  $K = 2$ , the augmented signal will be interpolated by the target segment with the segment of the highest correlation value and the target segment with the segment of the second-highest correlation value.

While random oversampling (ROS) and random undersampling (RUS) are commonly used to address binary class data imbalance problems, ROS can lead to overfitting. In multi-class datasets, the synthetic minority oversampling technique (SMOTE) is widely used to generate artificial samples through interpolating the minority samples and reducing overfitting [27,28]. However, most SMOTE methods use Euclidean distance to search for the  $k$ -nearest neighbors without considering the signal's correlation [32,33,53]. In this study, the proposed Next-OS method augments the synthetic data by selecting the beat and the next beat with the minimum Euclidean distance from one subject. The pseudo-code for the Next-OS method is presented in Appendix A Algorithm A1. Although the morphology of heartbeats differs among healthy individuals, the baseline of adjacent heartbeats has low variation, making it suitable for use in the proposed method. The proposed method finds all heartbeats with the same ectopic type in one record, except the segments where

the R peak is too close to the beginning or end of the records. If there is no other heartbeat with the same ectopic type next to the target heartbeat, the method goes back to the first heartbeat of the record. Some augmented beats may be missing in the data that are too close to the end in the Next-OS method.

**Algorithm 1. Pseudo-code of Corr-OS()**

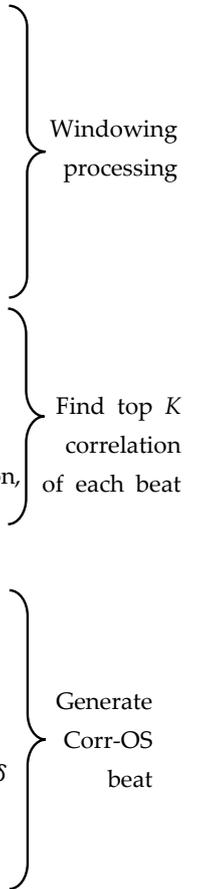
```

1: Inputs: Symbol (downloaded from PhysioBank ATM) array of the subject:  $S$ 
           30-min ECG Lead II signal of the subject:  $ecg$ 
           Total number (beats) of the class:  $n$ 
           Total number (beats) of the majority class:  $N_{maj}$ 
           Total number (beats) of the minority class:  $N_{minor}$ 
           Dilation of the CWT:  $a = 0.1$ 
           Translation of the CWT:  $b = 0$ 

2: Method:
   for  $h = 1$  to  $nS$  (total number of subjects)
     for  $i = 1$  to  $n$  (total number (beats) of the record)
        $S\_R =$  sampling rate = 360 Hz in this study
        $L =$  data length
       if  $(S[i] < S\_R \times 0.35)$  or  $(S[i] > L - S\_R \times 0.36)$ 
         continue
       else
          $T[h][i] = ecg[S[i] - (S\_R \times 0.35):S[i] + (S\_R \times 0.36)]$ 
       end if
     end
     for  $i = 1$  to  $n$ 
       for  $j = 1$  to  $n$ 
          $Local\_Corr[j] =$  Correlation Coefficient of  $T[i]$  and  $T[j]$ 
       end
       for  $k = 1$  to  $K$  ( $K \in \mathbb{N}$ : define by user)
          $Corr[i][k] = k + 1$  order of  $Local\_Corr$  (the maximum is self-correlation,
          $Local\_Corr[1] = 1$ )
       end
     end
     for  $i = 1$  to  $n$ 
        $L =$  round to the nearest integer of  $(N_{maj}/N_{minor}) - 1$ 
       for  $j = 1$  to  $K$  ( $K \in \mathbb{N}$ : defined by user)
         for  $k = 1$  to  $L/K$ 
            $\delta =$  random numbers where  $\delta \in [0, 1]$ 
            $NewT[(j - 1) \left(\frac{L}{2}\right) + k][256] = T[i][256] + (Corr[i][j] - T[i][256]) \times \delta$ 
         end
       end
     end
   end

3: Output: Array for Corr-OS beats:  $NewT$ 
           CWT matrix:  $C$ 

```



3.5. Cross-Validation (CV)

This study utilized the following three cross-validation methods for validating original (native) data and augmented data. The schematic diagram of three CV methods is shown in Figure 1 in the left bottom block (CV part). To obtain more reliable and steady predictions and reduce overfitting in deep learning, Zheng et al. applied various augmentation processes in different stages: augmentation in training stage, augmentation in testing stage, full stage data augmentation, and no data augmentation [54].

- CV1—original data: training/testing with native data; it can represent the baseline of the classification performance.
- CV2—full stage augmentation: training/testing with augmented data; it can represent the overall classification performance for all augmented data.
- CV3—augmentation in training stage: training with augmented data but testing with native data; it can represent the real-world case classification performance. It was only using augmented data for training that can avoid training similar images to cause overfitting. Santos et al. proposed a method that utilizes cross-validation during oversampling rather than  $k$ -fold cross-validation (randomly separate) after oversampling [55]. The testing data only kept the original data subset, and the oversampling data were not used in the training set. The present study regarded every single heartbeat as different data numbers (not subject-related).

In  $k$ -fold cross-validation, the original samples are randomly divided into  $k$  equal-sized subsamples [56]. One of the  $k$  subsamples is then selected as the verification data, while the remaining  $k - 1$  subsamples are used for training. This process is repeated  $k$  times, with each subsample used exactly once as the verification data. The results are then averaged to produce a single estimate. This method offers an advantage over repeated random subsampling, as all observations are used for training and verification, and each observation is used only once for verification purposes. Typically, 10-fold cross-validation is used when  $k$  is an unfixed parameter. For example, if  $k$  equals 10, all the data are divided into ten folders, and the first folder is used for testing while the remaining data are used for training. The process is then repeated ten times, with each folder used once for testing until all the data have been tested. The final result is the mean of every calculation.

## 4. Experiments and Results

### 4.1. MIT-BIH Arrhythmia Database

PhysioNet provides free web access to large collections of recorded physiological signals and related open-source software. In this study, the MIT-BIH arrhythmia database and the INCART 12-lead arrhythmia database from PhysioNet were utilized for training and testing the proposed DL algorithm [57]. The MIT-BIH arrhythmia database comprises over 4000 long-term Holter recordings obtained by the Beth Israel Hospital Arrhythmia Laboratory between 1975 and 1979 [58]. The database includes 23 records chosen randomly from this set and 25 records selected from the same set. Each of the 48 records is slightly over 30 min long, with the upper signal being a modified limb lead II (MLII) obtained by placing electrodes on the chest in most records. The records also contain annotations transcribed from paper chart recordings.

The present study utilized the Association for the Advancement of Medical Instrumentation (AAMI) classes, which classify fifteen types of MIT-BIH heartbeats into five classes: N (non-ectopic beats), S (supraventricular ectopic beats), V (ventricular ectopic beats), F (fusion beats), and Q (unknown beats), as shown in Table 2. The MIT-BIH arrhythmia database records have a sampling rate of 360 Hz, and all 48 records are approximately 30 min long. The records contain symbols that classify heartbeats into different arrhythmia types based on the occurrence time of the R peak. The proposed method chose 127 samples before and 128 samples after the R peak from these symbols. The N class includes 90,586 beats, the S class includes 2781 beats, the V class includes 7235 beats, the F class includes 803 beats, and the Q class includes 8039 beats.

**Table 2.** Sample numbers in the MIT-BIH arrhythmia database and class description using the AAMI standard.

AAMI Class	MIT-BIH Symbol	MIT-BIH Heartbeat Types	Numbers	(%)
N (Non-ectopic beats)	N	Normal beat	75,015	68.543
	L	Left bundle branch block beat	8071	7.375
	R	Right bundle branch block beat	7255	6.629
	j	Nodal (junctional) escape beat	229	0.209
	e	Atrial escape beat	16	0.015
S (Supraventricular ectopic beats)	A	Atrial premature beat	2546	2.326
	a	Aberrated atrial premature beat	150	0.137
	J	Nodal (junctional) premature beat	83	0.076
	S	Supraventricular premature beat	2	0.002
V (Ventricular ectopic beats)	V	Premature ventricular contraction beat	7129	6.514
	E	Ventricular escape beat	106	0.097
F (Fusion beats)	F	Fusion of ventricular and normal beat	802	0.733
Q (Unknown beats)	Q	Unclassifiable beat	33	0.030
	/	Paced beat	7024	6.418
	f	Fusion of paced and normal beats	982	0.897
Total:			109,443	100%

In this study, thirty-three augmentations were performed for every two beats in the S class, twelve augmentations were performed for every two beats in the V class, one hundred ten and one hundred thirteen augmentations were performed for every two beats in the F class, and eleven augmentations were performed for every two beats in the Q class. The total numbers for each minority class were brought closer to that of the majority class, N. The data numbers of the Next-OS and Corr-OS algorithms are shown in Table 3, representing the total number of heartbeats from all records that were randomly separated into ten folds using K-fold CV (K = 10).

**Table 3.** Data numbers in the dataset used in the experiments.

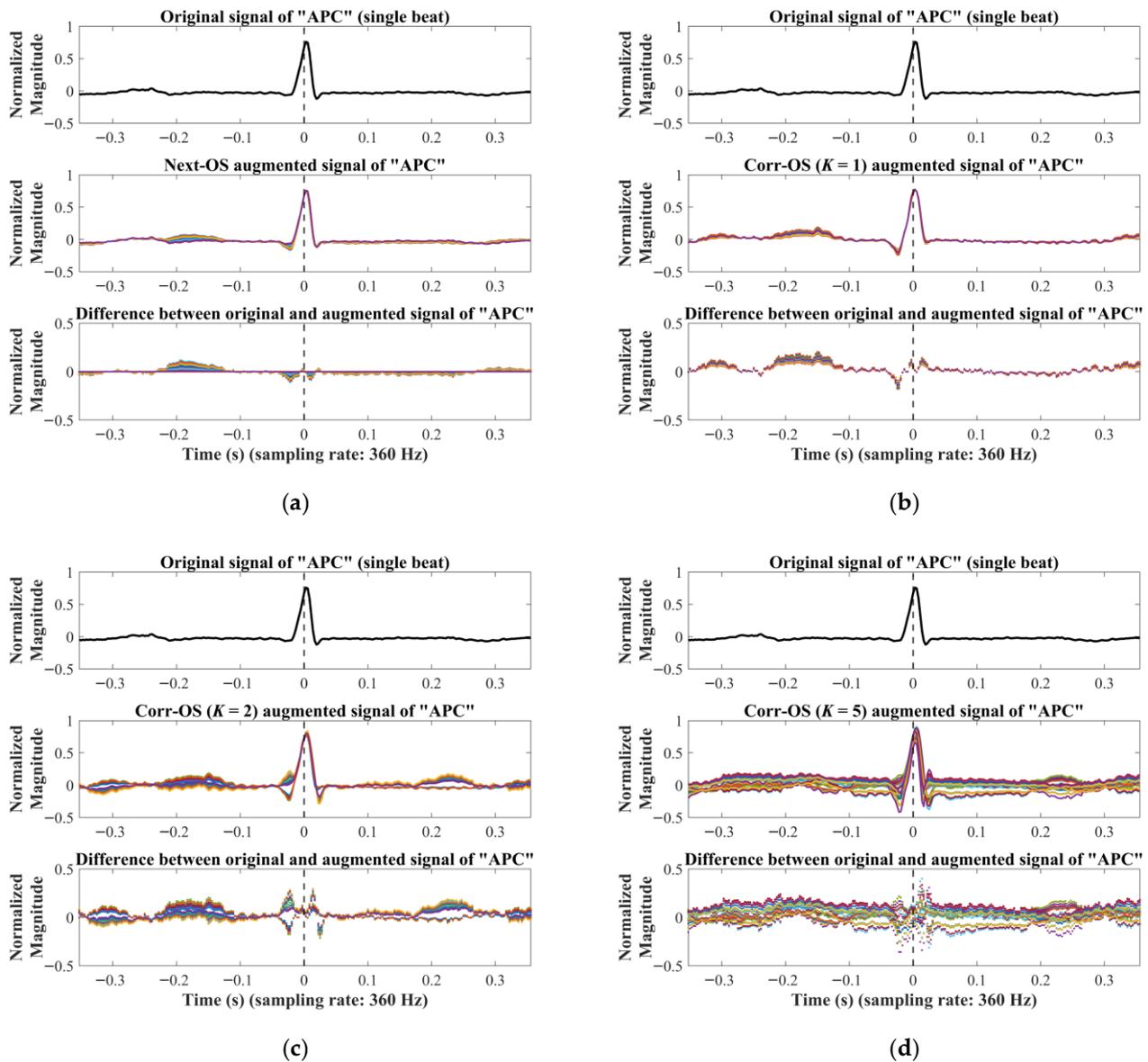
Class	Original Data Number	Ratio (Majority/Class)	Total Number after Oversampling	
			Next-OS	Corr-OS
N (majority)	90,586	1	90,586	90,586
S	2781	32.6	91,773	91,773
V	7236	12.5	87,013	94,068
F	803	112.8	88,802	90,739
Q	8039	11.3	88,439	88,429

#### 4.2. Results

##### 4.2.1. Calculating Error between Native/Augmented Segments Using Differences Plot

The augmented beats can be generated by the raw data of different beats. Note that the amplitude in Figures 3 and A6 are normalized amplitudes. The differences ( $Difference = Signal_{oversampling} - Signal_{original}$ ) between the original signal and the augmented signal of the “A” symbol for the two oversampling methods are shown in Figure 3. The differences between Next-OS (Figure 3a) and Corr-OS (Figure 3b) (K = 1) are narrow, and Corr-OS keeps the baseline of the ECG compared with Next-OS. Using the top two (Figure 3c) and top five (Figure 3d) correlated beats of the Corr-OS method can generate more diverse signals than Next-OS (Figure 3a) and Corr-OS (Figure 3b) with K = 1. The first column of Figure 3a shows the raw ECG segment. The last two columns represented the augmented segments with different colors and the differences from the raw segment. The difference became larger when the K of Corr-OS increased. We then augmented different level variation segments for evaluating the performance of Next-OS and Corr-OS (K = 1~5). As shown in

Figure A6, the Corr-OS beats can be generated from the raw data of the beat and the next beat. Note that the amplitude in Figure A6 is the normalized amplitude. The difference between the original signal and the Corr-OS augmented signal of the F class (fusion beat) and the Q class (unknown beat) might be larger than that for the S class and the V class. This is because fusion beats and unknown beats might comprise a mixture of multiple diseases.



**Figure 3.** Differences between native and augmented ECG segments (symbol A, atrial premature beats) generated by the (a) Next-OS (next beat) and Corr-OS (top  $K$  correlation beats) algorithms with (b)  $K = 1$ , (c)  $K = 2$ , and (d)  $K = 5$  (figure adapted from [59]).

#### 4.2.2. Classification Performance between LMUEBCNet with Existed Models Using Corr-OS/Next-OS Methods under CV1/CV2/CV3

All the classification results were shown in Table 4 (CV1 and CV2) and Table 5 (CV3). The sensitivity and precision of each class are shown in Tables A1–A3. The total accuracy of classifying five AAMI classes using deep feature extraction with AlexNet and SVM classifier under CV1 (native dataset) achieved 99.4%.

**Table 4.** Classification performance between LMUEBCNet with existing models using native/Corr-OS/Next-OS dataset under CV1/CV2.

Data Augmentation	Classifier	F1-Score					Total Acc.	F1-Score		Memory Usage (MB)	
		N	S	V	F	Q		Macro	Weighted		
Native (CV1 *)	SVM	99.9	94.1	96.7	87.0	98.9	99.4	95.3	94.5	N/A	
	AlexNet	100.0	95.3	97.3	88.8	99.3	99.6	96.1	95.6	45.1	
	ResNet18	100.0	94.2	96.8	87.1	98.9	99.5	95.4	94.6	39.8	
	VGG19	100.0	95.9	97.5	89.3	99.2	99.6	96.4	96.1	496.0	
	LMUEBCNet	99.9	90.1	94.8	81.7	98.1	99.1	92.9	90.8	1.1	
Augmented (CV2 *)	Next-OS	AlexNet	99.6	99.9	99.6	99.8	99.8	99.7	99.7	99.7	45.7
		ResNet18	99.6	99.8	99.6	99.8	99.8	99.7	99.7	99.7	40.2
		VGG19	99.6	99.9	99.6	99.8	99.9	99.8	99.8	99.8	496.0
	Corr-OS (K = 1)	AlexNet	100.0	99.9	99.8	99.8	99.9	99.9	99.9	99.9	47.3
		ResNet18	100.0	99.9	99.9	99.9	99.9	99.9	99.9	99.9	41.8
		VGG19	100.0	99.9	99.8	99.9	99.9	99.9	99.9	99.9	499.3
		LMUEBCNet	99.7	99.4	98.9	99.4	99.7	99.4	99.4	99.4	2.6
	Corr-OS (K = 2)	AlexNet	100.0	99.8	99.6	99.8	99.8	99.8	99.8	99.8	-
		ResNet18	100.0	99.8	99.7	99.9	99.8	99.9	99.9	99.9	-
		VGG19	100.0	99.8	99.6	99.8	99.9	99.8	99.8	99.8	-
		LMUEBCNet	99.5	98.2	97.6	98.6	99.4	98.7	98.7	98.7	-
	Corr-OS (K = 3)	AlexNet	100.0	99.7	99.4	99.7	99.8	99.7	99.7	99.7	-
		ResNet18	100.0	99.7	99.6	99.8	99.9	99.8	99.8	99.8	-
		VGG19	100.0	99.7	99.4	99.7	99.8	99.7	99.7	99.7	-
		LMUEBCNet	99.4	98.3	97.4	98.4	99.4	98.6	98.6	98.6	-
Corr-OS (K = 4)	AlexNet	100.0	99.6	99.3	99.7	99.8	99.7	99.7	99.7	-	
	ResNet18	100.0	99.8	99.7	99.9	99.8	99.8	99.8	99.8	-	
	VGG19	100.0	99.7	99.4	99.7	99.8	99.7	99.7	99.7	-	
	LMUEBCNet	99.7	98.2	97.5	98.3	99.4	98.6	98.6	98.6	-	
Corr-OS (K = 5)	AlexNet	100.0	99.5	99.1	99.6	99.7	99.6	99.6	99.6	-	
	ResNet18	100.0	99.5	99.2	99.7	99.7	99.6	99.6	99.6	-	
	VGG19	100.0	99.5	99.0	99.5	99.7	99.6	99.6	99.6	-	
	LMUEBCNet	99.7	98.0	97.1	98.1	99.3	98.4	98.5	98.5	-	

\* CV1: original data; CV2: full stage augmentation.

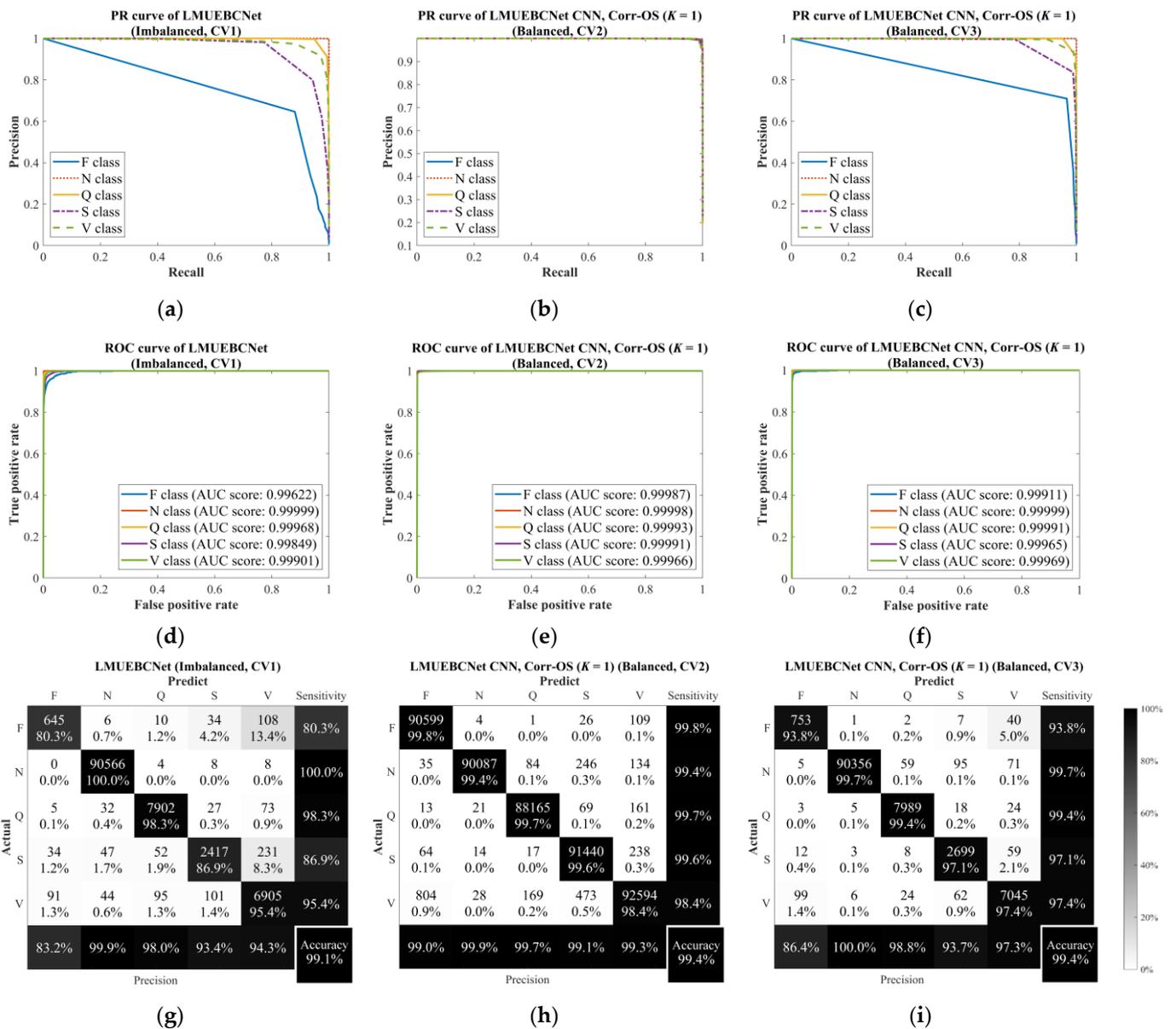
The LMUEBCNet (memory usage: 1.1 MB) under CV1 (native) can achieve a 92.9% macro F1-score, with the F1-scores for each class being 99.9%, 90.1%, 94.8%, 81.7%, and 98.1%, respectively. After data augmentation, the LMUEBCNet under Corr-OS (K = 1) in CV2 (full stage) can achieve a 99.4% F1-score with a 6.5% improvement in comparison to native dataset, and the F1-scores for each class were 99.7%, 99.4%, 99.3%, 99.4%, and 99.7%, respectively. The LMUEBCNet under Corr-OS (K = 5) in CV3 (training stage) can achieve a 96.1% macro F1-score with a 3.2% improvement in comparison to the native dataset, and the F1-scores for each class were 99.9%, 95.8%, 97.6%, 87.9%, and 99.3%, respectively. The PR curve of the F class using the LMUEBCNet can be improved after applying Corr-OS, as shown in Figure 4a–c. The ROC curve of the LMUEBCNet achieved an AUC of almost 1 for all classes, as shown in Figure 4d–f. The confusion matrix of the LMUEBCNet using different cross-validation methods (CV1, CV2, and CV3) is shown in Figures 4d–f and A1.

**Table 5.** Classification performance between LMUEBCNet with existing models using Corr-OS/Next-OS dataset under CV3.

Data Augmentation	Classifier	N	S	F1-Score			Total Acc.	F1-Score		Memory Usage (MB)	
				V	F	Q		Macro	Weighted		
Next-OS	AlexNet	99.2	97.0	92.7	84.2	98.9	98.6	94.4	96.5	45.7	
	ResNet18	99.5	95.8	96.3	80.7	99.5	99.0	94.4	95.8	40.2	
	VGG19	99.5	95.9	95.7	85.5	99.5	99.0	95.2	95.9	496.0	
Corr-OS	(K = 1)	AlexNet	100.0	98.5	98.9	94.6	99.6	99.8	98.3	98.5	47.1
		ResNet18	100.0	98.3	98.9	94.6	99.6	99.8	98.3	98.3	41.7
		VGG19	100.0	98.4	99.0	94.7	99.6	99.8	98.4	98.5	499.3
		LMUEBCNet	99.9	95.3	97.3	89.9	99.1	99.4	96.3	95.6	2.4
	(K = 2)	AlexNet	100.0	98.7	99.0	95.4	99.7	99.8	98.6	98.7	-
		ResNet18	100.0	98.5	99.1	95.8	99.6	99.8	98.6	98.6	-
		VGG19	100.0	98.8	99.2	95.7	99.7	99.9	98.7	98.8	-
		LMUEBCNet	99.8	93.9	97.0	88.6	99.1	99.3	95.7	94.4	-
	(K = 3)	AlexNet	100.0	98.9	99.2	96.0	99.6	99.8	98.7	98.9	-
		ResNet18	100.0	98.5	99.0	96.1	99.6	99.8	98.6	98.5	-
		VGG19	100.0	98.9	99.2	95.7	99.7	99.8	98.7	98.9	-
		LMUEBCNet	99.7	93.3	97.0	87.7	98.7	99.2	95.3	93.8	-
	(K = 4)	AlexNet	100.0	98.6	99.1	95.7	99.6	99.8	98.6	98.7	-
		ResNet18	100.0	96.2	97.0	90.8	99.4	99.6	96.7	96.4	-
		VGG19	100.0	99.0	99.1	95.4	99.7	99.8	98.6	99.0	-
LMUEBCNet		99.9	96.0	97.4	88.1	99.2	99.5	96.1	96.1	-	
(K = 5)	AlexNet	100.0	98.8	99.1	95.6	99.7	99.8	98.6	98.8	-	
	ResNet18	100.0	98.9	99.2	96.5	99.7	99.9	98.9	98.9	-	
	VGG19	100.0	99.0	99.2	95.0	99.7	99.9	98.6	98.9	-	
	LMUEBCNet	99.9	95.8	97.6	87.9	99.3	99.5	96.1	96.0	-	

AlexNet (memory usage: 45.1 MB) under CV1 (native) can achieve a 96.1% F1-score, with the F1-scores for each class being 100.0%, 95.3%, 97.3%, 88.8%, and 99.3%, respectively. After data augmentation, AlexNet under Corr-OS (K = 1) in CV2 (full stage) can achieve a 99.9% F1-score with a 3.8% improvement in comparison to the native dataset, and the F1-scores for each class were 100.0%, 99.9%, 99.8%, 99.8%, and 99.9%, respectively. AlexNet under Next-OS in CV2 (full stage) can achieve a 99.7% F1-score, with the F1-scores for each class being 99.6%, 99.8%, 99.6%, 99.8%, and 99.8%, respectively. AlexNet under Corr-OS (K = 5) in CV3 (training stage) can achieve a 98.6% macro F1-score, with a 2.5% improvement in comparison to the native dataset, and the F1-scores for each class were 100.0%, 98.8%, 99.1%, 95.6%, and 99.7%, respectively. AlexNet under Next-OS in CV3 (training stage) can achieve a 94.4% F1-score, with the F1-scores for each class being 99.2%, 97.0%, 92.7%, 84.2%, and 98.9%, respectively.

ResNet18 (memory usage: 39.8 MB) under CV1 (native) can achieve a 95.4% macro F1-score, with the F1-scores for each class being 100.0%, 94.2%, 96.8%, 87.1%, and 98.9%, respectively. After data augmentation, ResNet18 under Corr-OS (K = 1) in CV2 (full stage) can achieve a 99.9% F1-score with a 4.5% improvement in comparison to the native dataset, and the F1-scores for each class were 100.0%, 99.9%, 99.9%, 99.9%, and 99.9%, respectively. ResNet18 under Next-OS in CV2 (full stage) can achieve a 99.7% F1-score, with the F1-scores for each class being 99.6%, 99.9%, 99.6%, 99.8%, and 99.9%, respectively. ResNet18 under Corr-OS (K = 5) in CV3 (training stage) can achieve a 98.6% macro F1-score with a 3.2% improvement in comparison to the native dataset, and the F1-scores for each class were 100.0%, 98.9%, 99.2%, 96.5%, and 99.7%, respectively. ResNet18 under Next-OS in CV3 (training stage) can achieve a 94.4% F1-score, with the F1-scores for each class being 99.5%, 95.8%, 96.3%, 80.7%, and 99.5%, respectively.



**Figure 4.** ROC curves and AUC scores of LMUEBCNet in (a) CV1, (b) CV2 of Corr-OS ( $K = 1$ ), and (c) CV3 of Corr-OS ( $K = 1$ ); PR curves of LMUEBCNet in (d) CV1, (e) CV2 of Corr-OS ( $K = 1$ ), and (f) CV3 of Corr-OS ( $K = 1$ ); confusion matrix of LMUEBCNet in (g) CV1, (h) CV2 of Corr-OS ( $K = 1$ ), and (i) CV2 of Corr-OS ( $K = 1$ ).

VGG19 (memory usage: 496.0 MB) under CV1 (native) can achieve a 96.4% macro F1-score, with the F1-scores for each class being 100.0%, 95.9%, 97.5%, 89.3%, and 99.2%, respectively. After data augmentation, VGG19 under Corr-OS ( $K = 1$ ) in CV2 (full stage) can achieve a 99.9% F1-score with a 3.5% improvement in comparison to the native dataset, and the F1-scores for each class were 100.0%, 99.9%, 99.8%, 99.9%, and 99.9%, respectively. VGG19 under Next-OS in CV2 (full stage) can achieve a 99.8% F1-score, with the F1-scores for each class being 99.6%, 99.9%, 99.6%, 99.8%, and 99.9%, respectively. VGG19 under Corr-OS ( $K = 5$ ) in CV3 (training stage) can achieve a 98.6% macro F1-score with a 2.2% improvement in comparison to the native dataset, and the F1-scores for each class were 100.0%, 99.0%, 99.2%, 95.0%, and 99.7%, respectively. VGG19 under Next-OS in CV3 (training stage) can achieve a 95.2% F1-score, with the F1-scores for each class being 99.5%, 95.9%, 95.7%, 85.5%, and 99.5%, respectively.

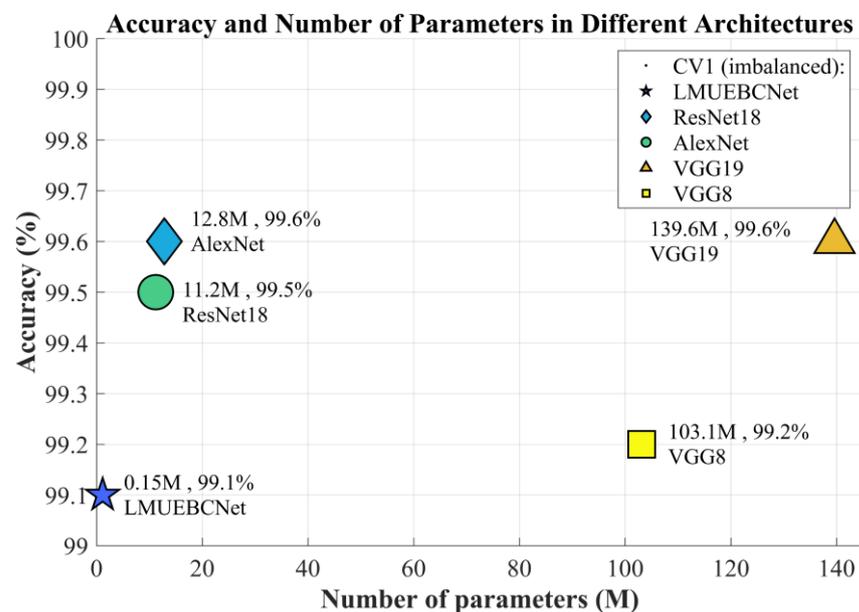
The confusion matrices of all classifiers are shown in Figures A1–A5. The AUC scores of AlexNet, ResNet18, and VGG19 for the imbalanced dataset are all about 0.99. The AUC scores of AlexNet, ResNet18, VGG19, and LMUEBCNet for the balanced dataset almost achieved a score of 1.

**5. Discussion**

The discussion/comparison based on the above classification results and existing literature can be described in the following five parts: (1) performance and memory/parameter usage of the LMUEBCNet vs. existing CNNs; (2) improvement using Corr-OS data augmentation method; (3) implications for different cross-validation methods; (4) classification performance compared with the existing literature; (5) limitations.

*5.1. Accuracy/F1-Score Performance and Memory Usage of LMUEBCNet vs. Existing CNNs*

The LMUEBCNet, with only 1% of VGG19’s parameters, achieved an overall accuracy of 99.1%, which is close to that of VGG19, as shown in Figure 5. Compared to other VGG-like architectures, such as VGG8 [38], the LMUEBCNet can save a significant amount of memory usage while maintaining high performance. Deeper CNNs, such as VGG19 and ResNet18, may slightly improve or maintain a similar performance to AlexNet, but they significantly increase the memory usage of the saved model. AlexNet, with 12.8M parameters, achieved a total accuracy of 99.5% and a weighted F1-score of 94.6%. ResNet18, with 11.2M parameters, achieved a total accuracy of 99.5% and a weighted F1-score of 94.6%. VGG19, with over 100M parameters, achieved a total accuracy of 99.6% and a weighted F1-score of 96.4%. The classification performance of machine learning (ML) using deep feature extraction with AlexNet and the SVM classifier (total accuracy of 99.4% and weighted F1-score of 94.5%) is slightly lower than that of deep learning (DL) using the AlexNet CNN (total accuracy of 99.6% and weighted F1-score of 95.6%).



**Figure 5.** Total accuracy and the number of parameters in different CNN architectures.

The total generation time for oversampling the ECG signals from the original imbalanced dataset to the balanced dataset was only about 100 s. However, considerable effort is required for processing CWT spectrograms, which is still lower than the time required for algorithmic augmentation methods such as the GAN. The proposed LMUEBCNet only requires 1 MB of memory, while AlexNet/ResNet18/VGG19 need to reserve 40 to 500 MB of memory. The LMUEBCNet can save 50% to 70% of training time due to its low memory usage while maintaining a high accuracy of over 99.0%, whereas using VGG19 requires

6 h for training in CV2/CV3. All the training processes were performed on the TWCC supercomputer with NVIDIA Tesla V100 32GB GPU. To avoid possible overfitting, this study set the max epochs in training options and stopped before the loss began to increase.

5.2. Improvement in Accuracy/F1-Score/Sensitivity Using Corr-OS/Next-OS Methods

The LMUEBCNet achieved an accuracy of 99.4% using Corr-OS ( $K = 1$ ) in CV2/CV3, with a 0.3% improvement in comparison to CV1 (native). Balancing the dataset using oversampling methods improved both the sensitivity and accuracy in CV2 and CV3. Next-OS and Corr-OS improved the average sensitivity in CV2 and CV3, resulting in a more balanced performance in all classes, rather than obtaining a result where the majority class is significantly higher than others. After balancing the dataset, the AUC scores of the five classes almost reached 1, and the PR curve of the minority classes, shown in Figure 4a–f, improved after applying oversampling methods (Next-OS and Corr-OS). In the VGG19 CNN, the total accuracy increased from 99.6% to 99.8% (Next-OS) and 99.9% (Corr-OS,  $K = 1$ ), with sensitivity for the S/V/F/Q classes improving by about 5.0%/1.7%/13.7%/0.5%, respectively, in CV2. However, the sensitivity of the N class decreased in the Next-OS dataset. Using VGG19 with Next-OS in CV3 only improved the sensitivity of each class but resulted in poor precision, with the weighted F1-score decreasing to 95.9% from 96.1% (native), and the F1-score of the V/F classes decreasing by 1.8%/3.8%, respectively.

The improvement can be observed in standard deviations (Stds.) of 10-fold cross-validation under CV1/CV2/CV3, as shown in Figure 6. The Std. of the pre-trained AlexNet/ResNet18/VGG19 CNN under the native dataset ranged between 0.06% to 0.11%, whereas the Std. of the augmented datasets (Next-OS and Corr-OS) was less than 0.05% in CV2, and the Std. was about 0.04% and 0.05% in CV3. The Std. of the proposed low-memory-usage LMUEBCNet CNN was lower than 0.3% in CV1/CV2/CV3.

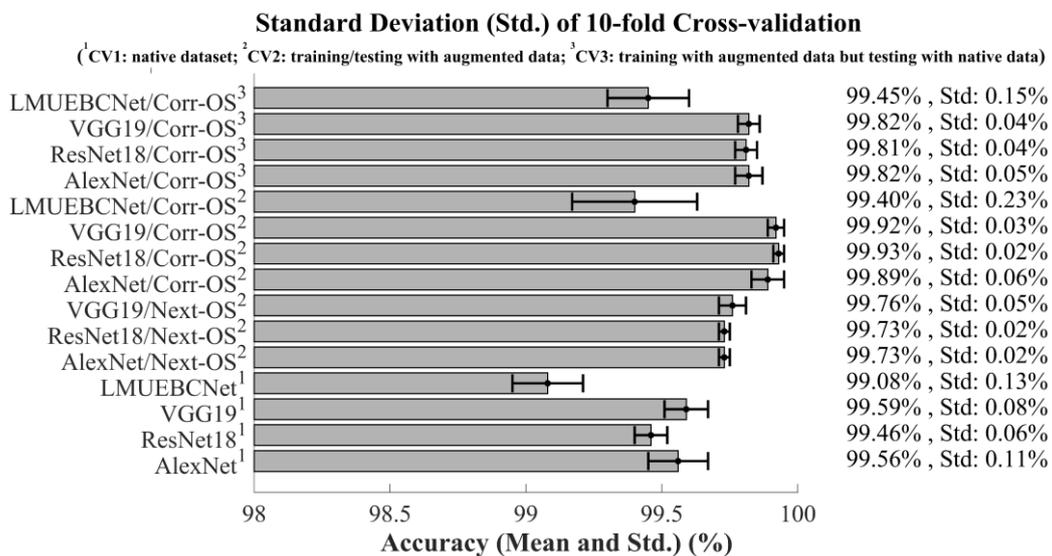


Figure 6. Standard deviation and total accuracy of LMUEBCNet/AlexNet/ResNet18/VGG19 using native/Next-OS/Corr-OS ( $K = 1$ ) dataset in CV1/CV2/CV3.

5.3. Implications for Different Cross-Validation Methods (CV1/CV2/CV3)

CV1 was used as a baseline to validate the native dataset. The baseline accuracy/F1-score of the LMUEBCNet was 99.1%/92.9%. Cross-validation using CV2 provides an overall performance evaluation of all augmented data. However, training/testing with augmented data may result in over-optimism. The LMUEBCNet achieved a higher accuracy/F1-score up to 99.4%/99.4% in CV2. CV3 better fits the actual use scenario, and the results are slightly lower than CV2. The accuracy/F1-score of the LMUEBCNet in CV3 are 99.4%/96.3%,

respectively. Training with augmented data in CV3 and testing with native data helps evaluate the performance of classification results, ensuring that the testing images are never considered in the training set. The PR curve shown in Figure 4b confirmed that CV2 is more optimistic, as the sensitivity, precision, and F1-score were always higher than CV3 for all CNN classifiers. In CV2, high similarity ECG images may appear in both the training/testing set, leading to over-optimism in 10-fold CV. Different from overfitting, Figure 3 shows that the augmented ECGs are not the same as the original ECG signal. Furthermore, a max epoch setting was used in this study to avoid overfitting during the learning task.

5.4. Classification Accuracy/F1-Score Performance Compared with the Existing Literature

Compared to previous studies using feature generation and machine learning or deep learning, Table 6 shows that the proposed LMUEBCNet with Corr-OS ( $K = 1$ ) in CV2/CV3 achieved higher F1-scores than other methods. The VGG19 CNN achieved a 99.5% accuracy (native) in CV1 and a 99.9% accuracy (Corr-OS,  $K = 5$ ) in CV3 for classifying N, S, V, F, and Q. Compared to Elhaj’s study [60], the VGG19 CNN in both CV2/CV3 showed higher total accuracy, F1-scores, and sensitivity for the N and V classes. The LSTM is commonly used in non-CNN deep learning algorithms. Ronald et al. proposed a RNN-based algorithm and achieved nearly a 100% accuracy with the ECG-ID and MIT-BIH arrhythmia database by training and testing on only 9/18 segments per subject [61]. Darmawahyuni et al. utilized the LSTM with forward and backward pass weight to learn the balanced weights of the majority and minority classes with an imbalance ratio of five times [62]. In contrast, the present study randomly separated all segments into ten folds, losing the time correlation and thus making LSTM not the best choice. The proposed method using the oversampling (Next-OS and Corr-OS) algorithm to balance the dataset achieved higher sensitivity (around 100%) for a balanced dataset than in the other literature. Overall, the results presented in this study outperformed the literature for N/S/V/Q classes in CV3 with a fairer cross-validation method.

Table 6. Comparison with the existing literature (10-fold cross-validation).

Work	Features and Balanced Methods		Classifier	Sensitivity (%)				Total Acc. (%)	Macro F1-Score	Memory Usage		
				N	S	V	F				Q	
[59]	PCA + DWT + HOS + ICA		SVM-RBF	98.9	100	98.9	100	100	98.9	-	-	
[10]	Std., Z-score	Native	Nine-layer CNN	88.4	85.3	92.7	88.2	95.5	89.1	69.3	-	
		Augmented		91.5	90.6	94.2	96.1	97.8	94.0	94.1	-	
[16]	Signal images (batch-weighted loss)		Nine-layer CNN	99.9	90.8	99.1	90.2	93.3	99.5	96.9	-	
[63]	LSTM-based auto-encoder		SVM	99.8	77.9	97.1	32.0	73.1	98.6	82.9	-	
[40]	Hybrid time–frequency diagram, ROS + RUS		ResNet-101	99.5	90.2	98.7	85.0	99.5	98.5	96.0	-	
Proposed method	CWT	Native	VGG19 CNN	100.0	95.0	98.1	86.2	99.4	99.6	96.4	496.0 MB	
			LMUEBCNet CNN	100.0	86.9	95.4	80.3	98.3	99.1	92.9	1.1 MB	
		Next-OS (CV2 *)	VGG19 CNN	99.6	99.8	99.6	99.9	99.8	99.8	99.8	99.8	496.0 MB
		Corr-OS ( $K = 1$ ) (CV2 *)	ResNet18 CNN	100.0	99.9	99.9	99.9	99.9	99.9	99.9	99.9	41.8 MB
			LMUEBCNet CNN	99.3	99.6	98.5	99.8	99.4	99.4	99.4	99.4	2.6 MB
		Corr-OS, ( $K = 1$ ) (CV3 *)	AlexNet CNN	100	98.3	99.1	94.8	99.6	99.8	98.3	98.3	47.0 MB
			ResNet18 CNN	100	98.5	98.9	93.8	99.6	99.8	98.3	98.3	41.6 MB
			VGG19 CNN	100	98.3	99.0	94.6	99.7	99.8	98.4	98.4	499.0 MB
LMUEBCNet CNN	99.7	97.1	97.4	93.8	99.4	99.4	96.3	96.3	2.4 MB			

\* CV3: augmentation in training stage.

### 5.5. Limitation

The first limitation of this study is the lack of a subject-level cross-validation (CV) method. While leave-one-subject-out (LOSO) CV is recommended for real-world ECG detection, it was not feasible in this study due to time constraints. The second limitation is that the proposed LMUEBCNet still cannot outperform pre-trained models (AlexNet/ResNet18/VGG19) with limited parameters. However, despite these limitations, this study provides valuable evaluation results using 10-fold CV on the MIT-BIH arrhythmia database. The results show that the proposed low memory usage LMUEBCNet CNN with the Corr-OS oversampling method outperforms the existing literature in terms of accuracy, F1-score, and sensitivity for the N/S/V/Q classes in both CV2 and CV3.

## 6. Conclusions and Future Works

The proposed LMUEBCNet algorithm achieves high ectopic beat classification accuracy with efficient parameter usage and utilizes Corr-OS to balance the dataset, resulting in improved classification performance. It requires lower computational effort and is therefore more feasible for implementation on resource-constrained mobile devices such as embedded systems. The LMUEBCNet achieved an accuracy of 99.4% using Corr-OS ( $K = 1$ ) in both CV2 and CV3, which is better than most previous studies using the MIT-BIH arrhythmia database. VGG19 with larger parameters under Corr-OS ( $K = 1$ ) in CV3 achieved better results for ventricular ectopic beat (V) and supraventricular ectopic beat (S) detection, with a sensitivity of 99.0% and 98.3%, respectively.

Due to the lack of medical resources, human power, and poor internet connectivity in resource-scarce areas, edge computing devices are a suitable solution. The proposed LMUEBCNet algorithm requires only about 1 MB of memory and is easily portable into embedded systems with limited memory size. The future work is to convert the LMUEBCNet model into an embedded system for more convenient applications.

**Author Contributions:** Conceptualization, Y.-L.X. and C.-W.L.; methodology, Y.-L.X. and C.-W.L.; software, Y.-L.X.; validation, Y.-L.X.; investigation, Y.-L.X. and C.-W.L.; resources, C.-W.L.; writing—original draft preparation, Y.-L.X.; writing—review and editing, C.-W.L.; supervision, C.-W.L. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the Ministry of Science and Technology (Taiwan), grant No. 108-2628-E-006-003-MY3.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

**Conflicts of Interest:** The authors declare no conflict of interest.

Appendix A

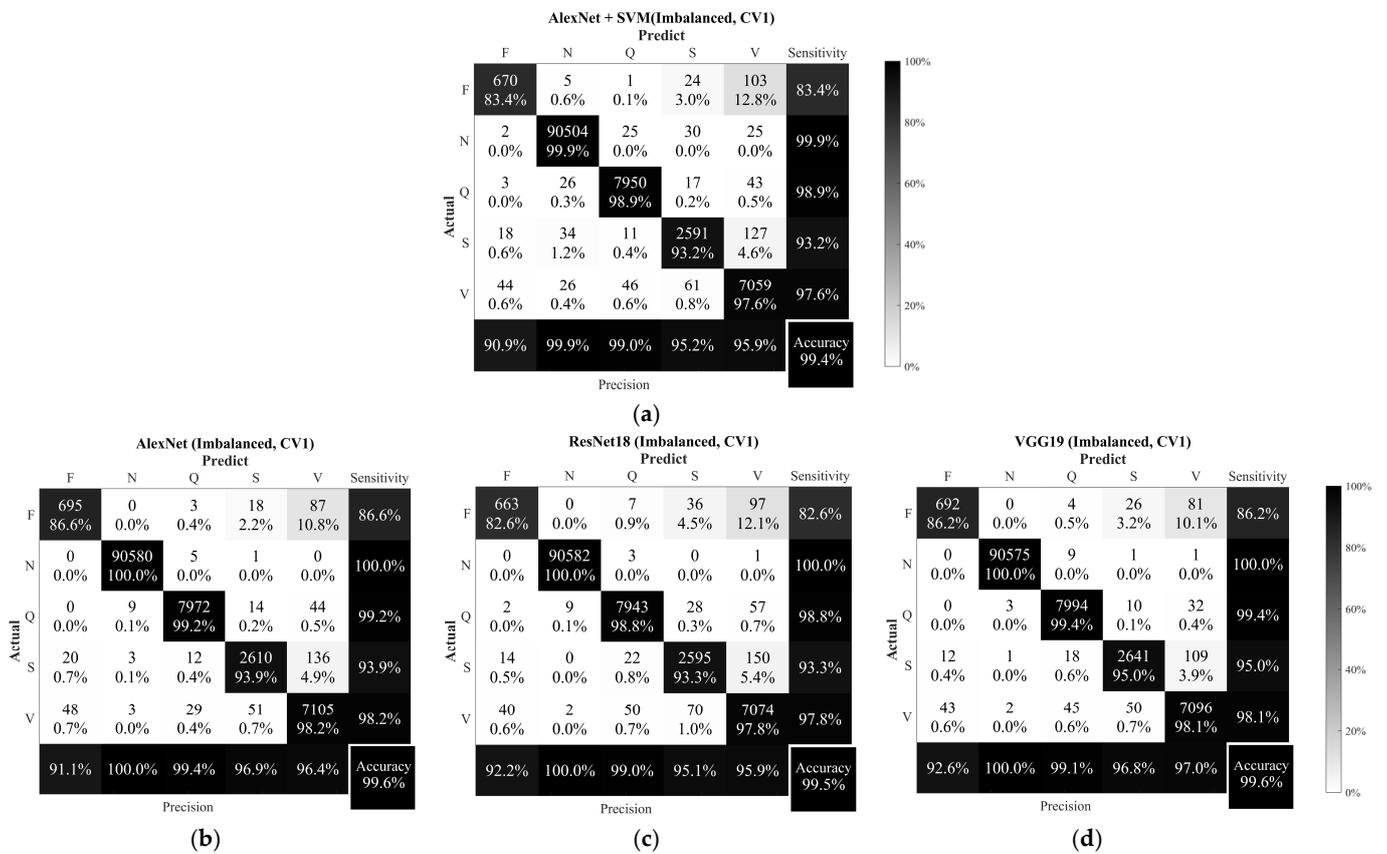
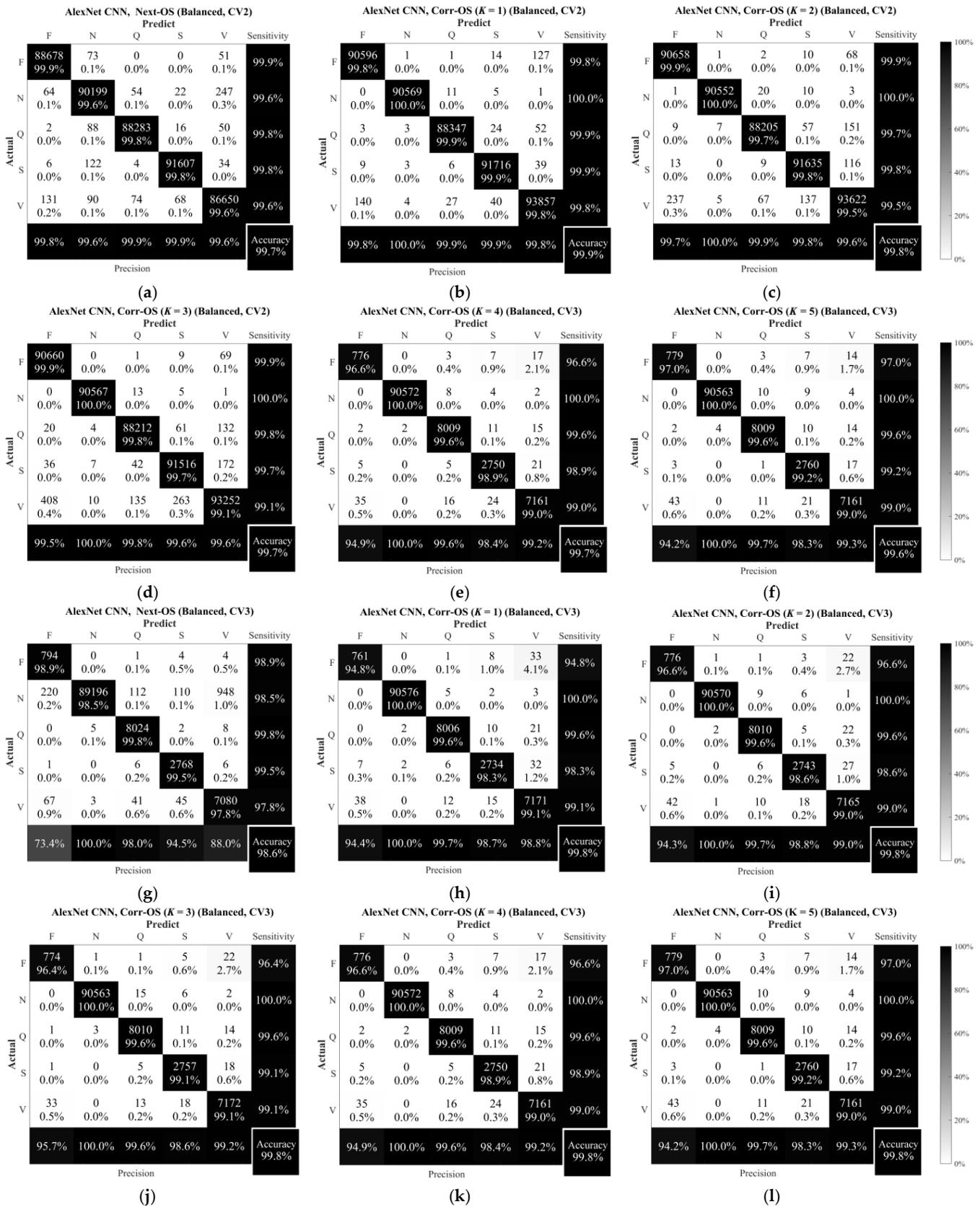
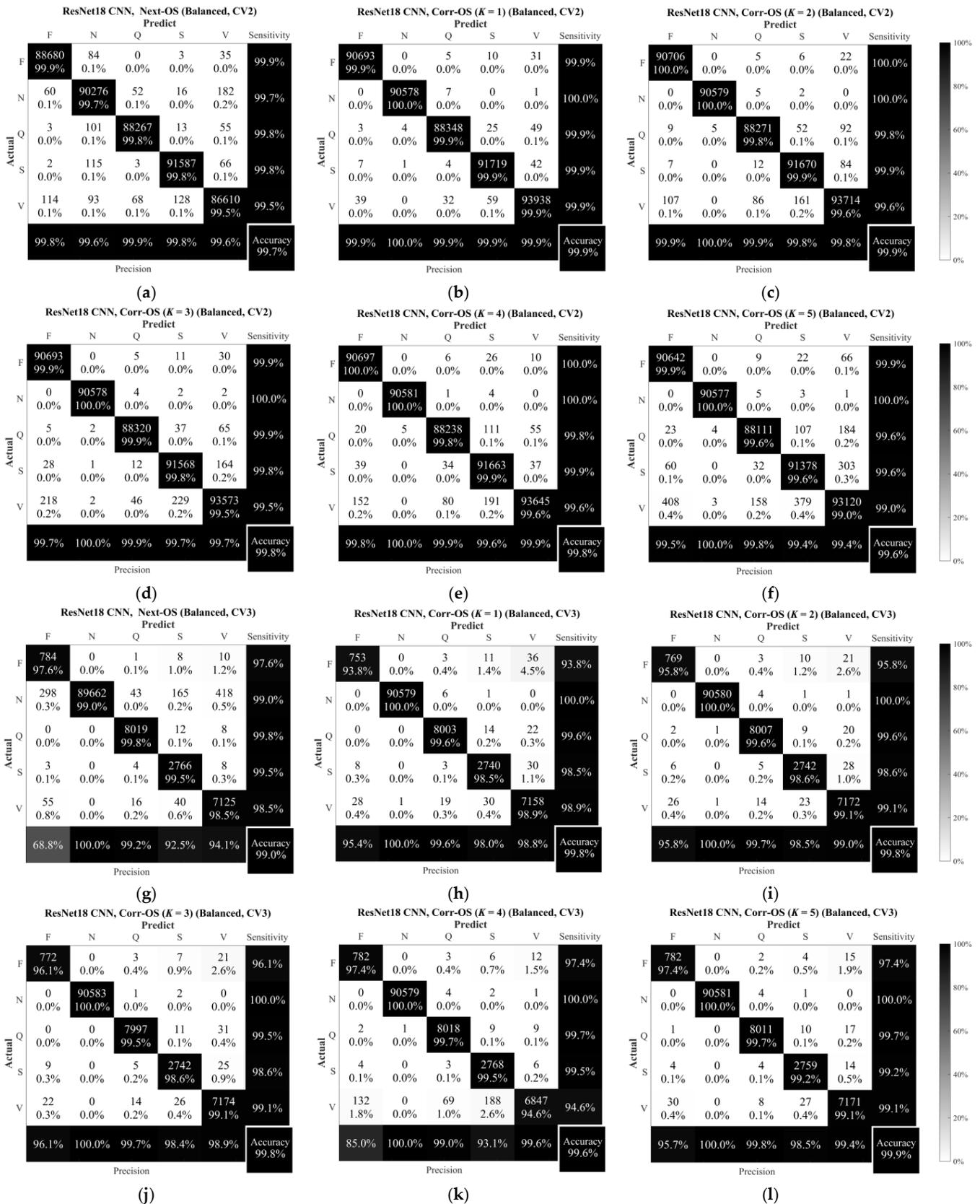


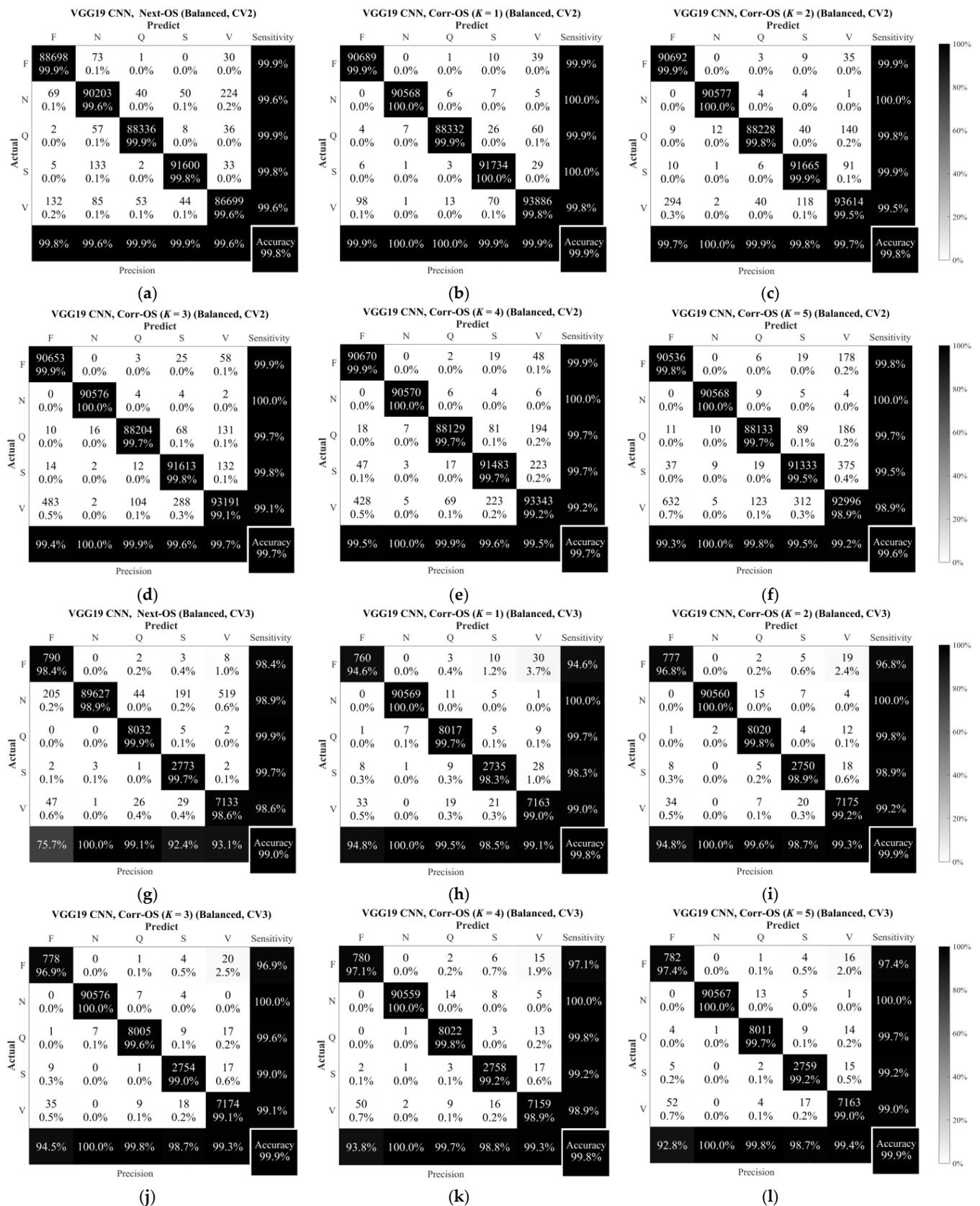
Figure A1. Confusion matrix of (a) AlexNet + SVM, (b) AlexNet, (c) ResNet18, and (d) VGG19 for the CV1 (imbalanced dataset).



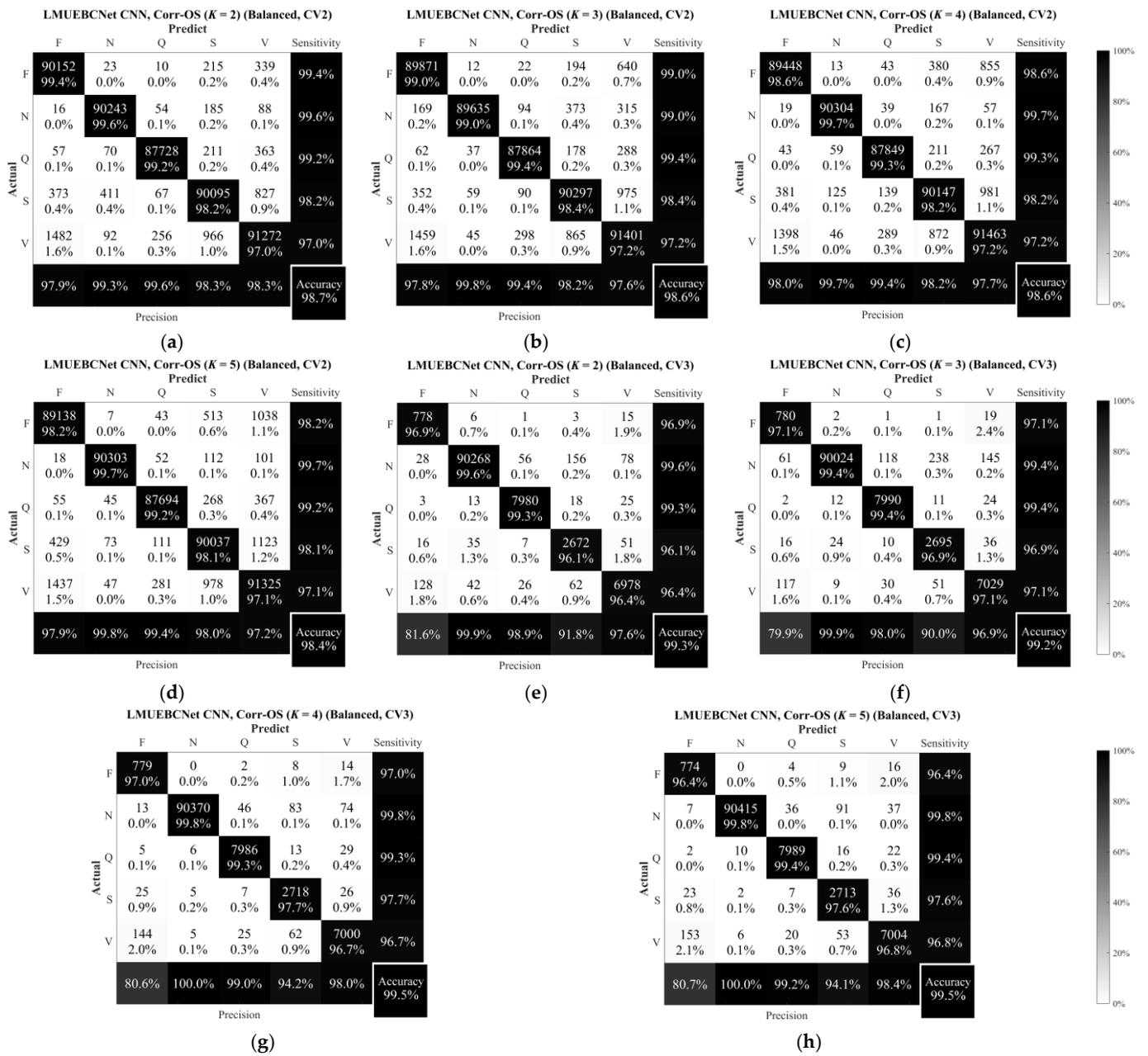
**Figure A2.** Confusion matrix of AlexNet using CV2 for (a) Next-OS, (b) Corr-OS (K = 1), (c) Corr-OS (K = 2), (d) Corr-OS (K = 3), (e) Corr-OS (K = 4), and (f) Corr-OS (K = 5); using CV3 for (g) Next-OS, (h) Corr-OS (K = 1), (i) Corr-OS (K = 2), (j) Corr-OS (K = 3), (k) Corr-OS (K = 4), and (l) Corr-OS (K = 5).



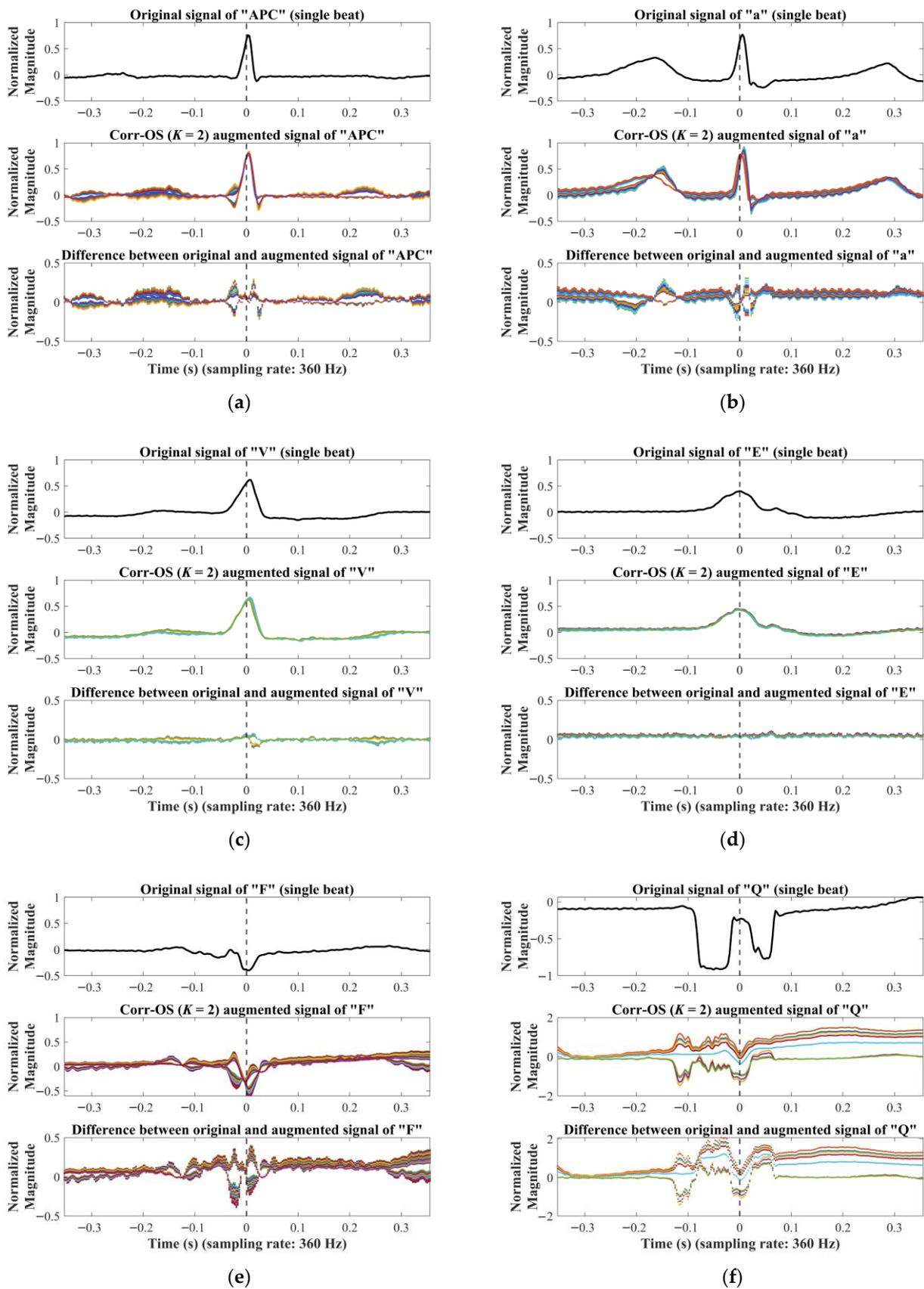
**Figure A3.** Confusion matrix of ResNet18 using CV2 for (a) Next-OS, (b) Corr-OS (K = 1), (c) Corr-OS (K = 2), (d) Corr-OS (K = 3), (e) Corr-OS (K = 4), and (f) Corr-OS (K = 5); using CV3 for (g) Next-OS, (h) Corr-OS (K = 1), (i) Corr-OS (K = 2), (j) Corr-OS (K = 3), (k) Corr-OS (K = 4), and (l) Corr-OS (K = 5).



**Figure A4.** Confusion matrix of VGG19 using CV2 for (a) Next-OS, (b) Corr-OS (K = 1), (c) Corr-OS (K = 2), (d) Corr-OS (K = 3), (e) Corr-OS (K = 4), and (f) Corr-OS (K = 5); using CV3 for (g) Next-OS, (h) Corr-OS (K = 1), (i) Corr-OS (K = 2), (j) Corr-OS (K = 3), (k) Corr-OS (K = 4), and (l) Corr-OS (K = 5).



**Figure A5.** Confusion matrix of LMUEBCNet using CV2 for (a) Corr-OS ( $K = 2$ ), (b) Corr-OS ( $K = 3$ ), (c) Corr-OS ( $K = 4$ ), and (d) Corr-OS ( $K = 5$ ); using CV3 for (e) Corr-OS ( $K = 2$ ), (f) Corr-OS ( $K = 3$ ), (g) Corr-OS ( $K = 4$ ), and (h) Corr-OS ( $K = 5$ ).



**Figure A6.** Differences between native and augmented ECG segments generated by Corr-OS ( $K = 2$ ) of (a) symbol A, (b) symbol a, (c) symbol V, (d) symbol E, (e) symbol F, and (f) symbol Q (figure adapted from [59]).

---

**Algorithm A1. Pseudo-code of Next-OS()**

---

1: **Inputs:** Symbol (downloaded from PhysioBank ATM) array of the subject:  $S$   
 30-min ECG Lead II signal of the subject:  $ecg$   
 Total number (beats) of the majority class:  $N_{maj}$   
 Total number (beats) of the minority class:  $N_{minor}$   
 Dilution of the CWT:  $a = 0.1$   
 Translation of the CWT:  $b = 0$

2: **Method:**

```

for  $h = 1$  to  $nS$  (total number of subjects)
  for  $i = 1$  to  $n$  (total number (beats) of the record)
     $S\_R =$  sampling rate = 360 Hz in this study
     $L =$  data length
    if  $(S[i] < S\_R \times 0.35)$  or  $(S[i] > L - S\_R \times 0.36)$ 
      continue
    else
       $T[i] = ecg[S[i] - (S\_R \times 0.35):S[i] + (S\_R \times 0.36)]$ 
      if  $i == n$ 
         $T[i + 1] = ecg[S[1] - (S\_R \times 0.35):S[1] + (S\_R \times 0.36)]$ 
      else
         $T[i + 1] = ecg[S[i + 1] - (S\_R \times 0.35):S[i + 1] + (S\_R \times 0.36)]$ 
      end if
       $AugN =$  round to the nearest integer of  $(N_{maj}/N_{minor}) - 1$ 
      for  $j=1$  to  $AugN$ 
         $\delta =$  random numbers where  $\delta \in [0,1]$ 
         $NewT[h][i][j] = T[i] + (T[i + 1] - T[i]) \times \delta$ 
      end
    end if
  end
end
    
```

} Generate Next-OS beat

3: **Output:** Array for Next-OS beats:  $NewT$   
 CWT matrix:  $C$

---

**Table A1.** Sensitivity/precision of LMUEBCNet with existing models using native/Corr-OS/Next-OS dataset under CV1.

Data Augmentation	Classifier	N		S		V		F		Q		Avg. Sen.
		Sen.	Pre.	Sen.	Pre.	Sen.	Pre.	Sen.	Pre.	Sen.	Pre.	
Native (CV1)	SVM	99.9	99.9	93.2	95.2	97.6	95.9	83.4	90.9	98.9	99.0	94.6
	AlexNet	100.0	100.0	93.9	96.9	98.2	96.4	86.6	91.1	99.2	99.4	95.6
	ResNet18	100.0	100.0	93.3	95.1	97.8	95.9	82.6	92.2	98.8	99.0	94.5
	VGG19	100.0	100.0	95.0	96.8	98.1	97.0	86.2	92.6	99.4	99.1	95.7
	LMUEBCNet	100.0	99.9	86.9	93.4	95.4	94.3	80.3	83.2	98.3	98.0	92.2

**Table A2.** Sensitivity/precision of LMUEBCNet with existing models using native/Corr-OS/Next-OS dataset under CV2.

Data Augmentation	Classifier	N		S		V		F		Q		Avg. Sen.	
		Sen.	Pre.	Sen.	Pre.	Sen.	Pre.	Sen.	Pre.	Se.	Pre.		
Augmented (CV2)	Next-OS	AlexNet	99.6	99.6	99.8	99.9	99.6	99.6	99.9	99.8	99.8	99.9	99.7
		ResNet18	99.7	99.6	99.8	99.8	99.5	99.6	99.9	99.8	99.8	99.9	99.7
		VGG19	99.6	99.6	99.8	99.9	99.6	99.6	99.9	99.8	99.9	99.9	99.8
	Corr-OS (K = 1)	AlexNet	100.0	100.0	99.9	99.9	99.8	99.8	99.8	99.8	99.9	99.9	99.9
		ResNet18	100.0	100.0	99.9	99.9	99.9	99.9	99.9	99.9	99.9	99.9	99.9
		VGG19	100.0	100.0	100.0	99.9	99.8	99.9	99.9	99.9	99.9	100.0	99.9
		LMUEBCNet	99.4	99.9	99.6	99.1	98.4	99.3	99.8	99.0	99.7	99.7	99.4
	Corr-OS (K = 2)	AlexNet	100.0	100.0	99.8	99.8	99.5	99.6	99.9	99.7	99.7	99.9	99.8
		ResNet18	100.0	100.0	99.9	99.8	99.6	99.8	100.0	99.9	99.8	99.9	99.9
		VGG19	100.0	100.0	99.9	99.8	99.5	99.7	99.9	99.7	99.8	99.9	99.8
		LMUEBCNet	99.6	99.3	98.2	98.3	97.0	98.3	99.4	97.9	99.2	99.6	98.7
	Corr-OS (K = 3)	AlexNet	100.0	100.0	99.7	99.6	99.1	99.6	99.9	99.5	99.8	99.8	99.7
		ResNet18	100.0	100.0	99.8	99.7	99.5	99.7	99.9	99.7	99.9	99.9	99.8
		VGG19	100.0	100.0	99.8	99.6	99.1	99.7	99.9	99.4	99.7	99.9	99.7
		LMUEBCNet	99.0	99.8	98.4	98.2	97.2	97.6	99.0	97.8	99.4	99.4	98.6
Corr-OS (K = 4)	AlexNet	100.0	100.0	99.8	99.5	99.0	99.5	99.8	99.5	99.7	99.8	99.7	
	ResNet18	100.0	100.0	99.9	99.6	99.6	99.9	100.0	99.8	99.8	99.9	99.8	
	VGG19	100.0	100.0	99.7	99.6	99.2	99.5	99.9	99.5	99.7	99.9	99.7	
	LMUEBCNet	99.7	99.7	98.2	98.2	97.2	97.7	98.6	98.0	99.3	99.4	98.6	
Corr-OS (K = 5)	AlexNet	100.0	100.0	99.6	99.4	98.9	99.4	99.8	99.4	99.6	99.8	99.6	
	ResNet18	100.0	100.0	99.6	99.4	99.0	99.4	99.9	99.5	99.6	99.8	99.6	
	VGG19	100.0	100.0	99.5	99.5	98.9	99.2	99.8	99.3	99.7	99.8	99.6	
	LMUEBCNet	99.7	99.8	98.1	98.0	97.1	97.2	98.2	97.9	99.2	99.4	98.4	

**Table A3.** Sensitivity/precision of LMUEBCNet with existing models using Corr-OS/Next-OS dataset under CV3.

Data Augmentation	Classifier	N		S		V		F		Q		Avg. Sen.	
		Sen.	Pre.	Sen.	Pre.	Sen.	Pre.	Sen.	Pre.	Sen.	Pre.		
Next-OS	AlexNet	98.5	100.0	99.5	94.5	97.8	88.0	98.9	73.4	99.8	98.0	98.9	
	ResNet18	99.0	100.0	99.5	92.5	98.5	94.1	97.6	68.8	99.8	99.2	98.9	
	VGG19	98.9	100.0	99.7	92.4	98.6	93.1	98.4	75.7	99.9	99.1	99.1	
Corr-OS	(K = 1)	AlexNet	100.0	100.0	98.3	98.7	99.1	98.8	94.8	94.4	99.6	99.7	98.4
		ResNet18	100.0	100.0	98.5	98.0	98.9	98.8	93.8	95.4	99.6	99.6	98.2
		VGG19	100.0	100.0	98.3	98.5	99.0	99.1	94.6	94.8	99.7	99.5	98.3
		LMUEBCNet	99.7	100.0	97.1	93.7	97.4	97.3	93.8	86.4	99.4	98.8	97.5
	(K = 2)	AlexNet	100.0	100.0	98.6	98.8	99.0	99.0	96.6	94.3	99.6	99.7	98.8
		ResNet18	100.0	100.0	98.6	98.5	99.1	99.0	95.8	95.8	99.6	99.7	98.6
		VGG19	100.0	100.0	98.9	98.7	99.2	99.3	96.8	94.8	99.8	99.6	98.9
		LMUEBCNet	99.6	99.9	96.1	91.8	96.4	97.6	96.9	81.6	99.3	98.9	97.7
	(K = 3)	AlexNet	100.0	100.0	99.1	98.6	99.1	99.2	96.4	95.7	99.6	99.6	98.8
		ResNet18	100.0	100.0	98.6	98.4	99.1	98.9	96.1	96.1	99.5	99.7	98.7
		VGG19	100.0	100.0	99.0	98.7	99.1	99.3	96.9	94.5	99.6	99.8	98.9
		LMUEBCNet	99.4	99.9	96.9	90.0	97.1	96.9	97.1	79.9	99.4	98.0	98.0
(K = 4)	AlexNet	100.0	100.0	98.9	98.4	99.0	99.2	96.6	94.9	99.6	99.6	98.8	
	ResNet18	100.0	100.0	99.5	93.1	94.6	99.6	97.4	85.0	99.7	99.0	98.2	
	VGG19	100.0	100.0	99.2	98.8	98.9	99.3	97.1	93.8	99.8	99.7	99.0	
	LMUEBCNet	99.8	100.0	97.7	94.2	96.7	98.0	97.0	80.6	99.3	99.0	98.1	
(K = 5)	AlexNet	100.0	100.0	99.2	98.3	99.0	99.3	97.0	94.2	99.6	99.7	99.0	
	ResNet18	100.0	100.0	99.2	98.5	99.1	99.4	97.4	95.7	99.7	99.8	99.1	
	VGG19	100.0	100.0	99.2	98.7	99.0	99.4	97.4	92.8	99.7	99.8	99.1	
	LMUEBCNet	99.8	100.0	97.6	94.1	96.8	98.4	96.4	80.7	99.4	99.2	98.0	

## References

1. Tsao, C.W.; Aday, A.W.; Almarzooq, Z.I.; Alonso, A.; Beaton, A.Z.; Bittencourt, M.S.; Boehme, A.K.; Buxton, A.E.; Carson, A.P.; Commodore-Mensah, Y.; et al. Heart Disease and Stroke Statistics—2022 Update: A Report From the American Heart Association. *Circulation* **2022**, *145*, e153–e639. [PubMed]
2. Stronati, G.; Benfaremo, D.; Selimi, A.; Ferraioli, Y.; Ferranti, F.; Dello Russo, A.; Guerra, F. Incidence and predictors of cardiac arrhythmias in patients with systemic sclerosis. *Europace* **2022**, *24* (Suppl. S1), euac053-124. [CrossRef]
3. ANSI/AAMI EC57; Testing and Reporting Performance Results of Cardiac Rhythm and ST Segment Measurement Algorithms. ANSI: New York, NY, USA, 2012. Available online: <https://webstore.ansi.org/standards/aami/ansiaamiec572012r2020> (accessed on 7 April 2022).
4. Afkhami, R.G.; Azarnia, G.; Tinati, M.A. Cardiac arrhythmia classification using statistical and mixture modeling features of ECG signals. *Pattern Recognit. Lett.* **2016**, *70*, 45–51. [CrossRef]
5. Li, T.; Zhou, M. ECG classification using wavelet packet entropy and random forests. *Entropy* **2016**, *18*, 285. [CrossRef]
6. Desai, U.; Martis, R.J.; Nayak, C.G.; Sarika, K.; Nayak, S.G.; Shirva, A.; Nayak, V.; Mudassir, S. Discrete cosine transform features in automated classification of cardiac arrhythmia beats. In *Emerging Research in Computing, Information, Communication and Applications*; Springer: New Delhi, India, 2015; pp. 153–162.
7. Martis, R.J.; Acharya, U.R.; Min, L.C. ECG beat classification using PCA, LDA, ICA and discrete wavelet transform. *Biomed. Signal Process. Control.* **2013**, *8*, 437–448. [CrossRef]
8. Yang, W.; Si, Y.; Wang, D.; Guo, B. Automatic recognition of arrhythmia based on principal component analysis network and linear support vector machine. *Comput. Biol. Med.* **2018**, *101*, 22–32. [CrossRef]
9. Ebrahimi, Z.; Loni, M.; Daneshlab, M.; Gharehbaghi, A. A review on deep learning methods for ECG arrhythmia classification. *Expert Syst. Appl.* **2020**, *7*, 100033. [CrossRef]
10. Acharya, U.R.; Oh, S.L.; Hagiwara, Y.; Tan, J.H.; Adam, M.; Gertych, A.; San Tan, R. A deep convolutional neural network model to classify heartbeats. *Comput. Biol. Med.* **2017**, *89*, 389–396. [CrossRef]
11. Wang, H.; Shi, H.; Chen, X.; Zhao, L.; Huang, Y.; Liu, C. An improved convolutional neural network based approach for automated heartbeat classification. *J. Med. Syst.* **2020**, *44*, 35. [CrossRef]
12. Romdhane, T.F.; Pr, M.A. Electrocardiogram heartbeat classification based on a deep convolutional neural network and focal loss. *Comput. Biol. Med.* **2020**, *123*, 103866. [CrossRef]
13. Yao, G.; Mao, X.; Li, N.; Xu, H.; Xu, X.; Jiao, Y.; Ni, J. Interpretation of electrocardiogram heartbeat by CNN and GRU. *Comput. Math. Methods Med.* **2021**, *2021*, 6534942. [CrossRef] [PubMed]
14. Al Rahhal, M.M.; Bazi, Y.; Al Hichri, H.; Alajlan, N.; Melgani, F.; Yager, R.R. Deep Learning Approach for Active Classification of Electrocardiogram Signals. *Inf. Sci.* **2016**, *345*, 340–354. [CrossRef]
15. Xie, Q.; Tu, S.; Wang, G.; Lian, Y.; Xu, L. Feature enrichment based convolutional neural network for heartbeat classification from electrocardiogram. *IEEE Access* **2019**, *7*, 153751–153760. [CrossRef]
16. Zhai, X.; Tin, C. Automated ECG classification using dual heartbeat coupling based on convolutional neural network. *IEEE Access* **2018**, *6*, 27465–27472. [CrossRef]
17. Sellami, A.; Hwang, H. A robust deep convolutional neural network with batch-weighted loss for heartbeat classification. *Expert Syst. Appl.* **2019**, *122*, 75–84. [CrossRef]
18. Li, F.; Xu, Y.; Chen, Z.; Liu, Z. Automated heartbeat classification using 3-d inputs based on convolutional neural network with multi-fields of view. *IEEE Access* **2019**, *7*, 76295–76304. [CrossRef]
19. Lu, P.; Gao, Y.; Xi, H.; Zhang, Y.; Gao, C.; Zhou, B.; Zhang, H.; Chen, L.; Mao, X. KecNet: A light neural network for arrhythmia classification based on knowledge reinforcement. *J. Health Eng.* **2021**, *2021*, 6684954. [CrossRef]
20. He, Z.; Zhang, X.; Cao, Y.; Liu, Z.; Zhang, B.; Wang, X. LiteNet: Lightweight neural network for detecting arrhythmias at resource-constrained mobile devices. *Sensors* **2018**, *18*, 1229. [CrossRef]
21. Mathunjwa, B.M.; Lin, Y.T.; Lin, C.H.; Abbod, M.F.; Sadrawi, M.; Shieh, J.S. ECG Recurrence Plot-Based Arrhythmia Classification Using Two-Dimensional Deep Residual CNN Features. *Sensors* **2022**, *22*, 1660. [CrossRef]
22. Khan, A.; Sohail, A.; Zahoor, U.; Qureshi, A.S. A survey of the recent architectures of deep convolutional neural networks. *Artif. Intell. Rev.* **2020**, *53*, 5455–5516. [CrossRef]
23. Dey, M.; Omar, N.; Ullah, M.A. Temporal Feature-Based Classification Into Myocardial Infarction and Other CVDs Merging CNN and Bi-LSTM From ECG Signal. *IEEE Sens. J.* **2021**, *21*, 21688–21695. [CrossRef]
24. Tsinalis, O.; Matthews, P.M.; Guo, Y. Automatic sleep stage scoring using time-frequency analysis and stacked sparse autoencoders. *Ann. Biomed. Eng.* **2016**, *44*, 1587–1597. [CrossRef]
25. Amrane, M.; Oukid, S.; Gagaoua, I.; Ensarlı, T. Breast cancer classification using machine learning. In Proceedings of the 2018 Electric Electronics, Computer Science, Biomedical Engineering's Meeting (EBBT), Istanbul, Turkey, 18–19 April 2018; pp. 1–4.
26. Gu, Y.; Ge, Z.; Bonnington, C.P.; Zhou, J. Progressive Transfer Learning And Adversarial Domain Adaptation For Cross-domain Skin Disease Classification. *IEEE J. Biomed. Health Inform.* **2019**, *24*, 1379–1393. [CrossRef] [PubMed]
27. He, H.; Garcia, E.A. Learning from imbalanced data. *IEEE Trans. Knowl. Data Eng.* **2009**, *21*, 1263–1284.
28. Johnson, J.M.; Khoshgoftaar, T.M. Survey on deep learning with class imbalance. *J. Big Data* **2019**, *6*, 27. [CrossRef]
29. Lu, W.; Hou, H.; Chu, J. Feature fusion for imbalanced ECG data analysis. *Biomed. Signal Process. Control.* **2018**, *41*, 152–160. [CrossRef]

30. Mousavi, S.; Afghah, F. Inter-and intra-patient ECG heartbeat classification for arrhythmia detection: A sequence to sequence deep learning approach. In Proceedings of the ICASSP 2019–2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Brighton, UK, 12–17 May 2019.
31. Shaker, A.M.; Tantawi, M.; Shedeed, H.A.; Tolba, M.F. Generalization of convolutional neural networks for ECG classification using generative adversarial networks. *IEEE Access* **2020**, *8*, 35592–35605. [[CrossRef](#)]
32. Pandey, S.K.; Janghel, R.R. Automatic detection of arrhythmia from imbalanced ECG database using CNN model with SMOTE. *Australas. Phys. Eng. Sci. Med.* **2019**, *42*, 1129–1139. [[CrossRef](#)]
33. Bhattacharyya, S.; Majumder, S.; Debnath, P.; Chanda, M. Arrhythmic heartbeat classification using ensemble of random forest and support vector machine algorithm. *IEEE Trans. Artif. Intell.* **2021**, *2*, 260–268. [[CrossRef](#)]
34. Rao, K.N.; Reddy, C. An efficient software defect analysis using correlation-based oversampling. *Arab. J. Sci. Eng.* **2018**, *43*, 4391–4411. [[CrossRef](#)]
35. Devi, D.; Biswas, S.K.; Purkayastha, B. Correlation-based oversampling aided cost sensitive ensemble learning technique for treatment of class imbalance. *J. Exp. Theor. Artif. Intell.* **2022**, *34*, 143–174. [[CrossRef](#)]
36. Fahrudin, T.; Buliali, J.L.; Fatchah, C. Enhancing the performance of smote algorithm by using attribute weighting scheme and new selective sampling method for imbalanced data set. *Int. J. Innov. Comput. Inf. Control.* **2019**, *15*, 423–444.
37. Jiang, Z.; Pan, T.; Zhang, C.; Yang, J. A new oversampling method based on the classification contribution degree. *Symmetry* **2021**, *13*, 194. [[CrossRef](#)]
38. Zhang, Q.; Shen, Y.; Yi, Z. Video-based traffic sign detection and recognition. In Proceedings of the 2019 International Conference on Image and Video Processing, and Artificial Intelligence, SPIE, Shanghai, China, 23–25 August 2019; pp. 284–291.
39. Breve, B.; Caruccio, L.; Cirillo, S.; Deufemia, V.; Polese, G. Visual ECG Analysis in Real-world Scenarios. In Proceedings of the 27th International DMS Conference on Visualization and Visual Languages (DMSVIVA2021), Pittsburgh, PA, USA, 29–30 June 2021; pp. 46–54.
40. Zhang, Y.; Li, J.; Wei, S.; Zhou, F.; Li, D. Heartbeats Classification Using Hybrid Time-Frequency Analysis and Transfer Learning Based on ResNet. *IEEE J. Biomed. Health Inf.* **2021**, *25*, 4175–4184. [[CrossRef](#)] [[PubMed](#)]
41. Ahmad, Z.; Tabassum, A.; Guan, L.; Khan, N.M. ECG heartbeat classification using multimodal fusion. *IEEE Access* **2021**, *9*, 100615–100626. [[CrossRef](#)]
42. Time-Frequency Analysis. Available online: <https://bit.ly/30tdZlo> (accessed on 10 January 2022).
43. Du, P.; Kibbe, W.A.; Lin, S.M. Improved Peak Detection in Mass Spectrum by Incorporating Continuous Wavelet Transform-based Pattern Matching. *Bioinformatics* **2006**, *22*, 2059–2065. [[CrossRef](#)] [[PubMed](#)]
44. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going Deeper with Convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1–9.
45. Deep Learning Toolbox. Available online: <https://bit.ly/2XFBgPf> (accessed on 10 January 2022).
46. Pretrained Deep Neural Networks. Available online: <https://bit.ly/2NKkna> (accessed on 10 January 2022).
47. ImageNet. Available online: <http://image-net.org/index> (accessed on 10 January 2022).
48. Krizhevsky, A.; Sutskever, I.; Hinton, G. ImageNet Classification with Deep Convolutional Neural Networks. *Adv. Neural Inf. Process. Syst.* **2012**, 1106–1114. [[CrossRef](#)]
49. Kaiming, H.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
50. Transfer Learning Using AlexNet. Available online: <https://bit.ly/2XPhmFV> (accessed on 10 January 2022).
51. Çınar, A.; Tuncer, S.A. Classification of normal sinus rhythm, abnormal arrhythmia and congestive heart failure ECG signals using LSTM and hybrid CNN-SVM deep neural networks. *Comput. Methods Biomech. Biomed. Eng.* **2021**, *24*, 203–214. [[CrossRef](#)]
52. Chawla, N.V.; Bowyer, K.W.; Hall, L.O.; Kegelmeyer, W.P. SMOTE: Synthetic minority over-sampling technique. *J. Artif. Intell. Res.* **2002**, *16*, 321–357. [[CrossRef](#)]
53. Chiu, C.C.; Lin, T.H.; Liau, B.Y. Using correlation coefficient in ECG waveform for arrhythmia detection. *Biomed. Eng. Appl. Basis Commun.* **2005**, *17*, 147–152. [[CrossRef](#)]
54. Zheng, Q.; Yang, M.; Tian, X.; Jiang, N.; Wang, D. A full stage data augmentation method in deep convolutional neural network for natural image classification. *Discret. Dyn. Nat. Soc.* **2020**, *2020*, 4706576. [[CrossRef](#)]
55. Santos, M.S.; Soares, J.P.; Abreu, P.H.; Araujo, H.; Santos, J. Cross-validation for imbalanced datasets: Avoiding overoptimistic and overfitting approaches [research frontier]. *IEEE Comput. Intell. Mag.* **2018**, *13*, 59–76. [[CrossRef](#)]
56. Cross-Validation (Statistics). Available online: <https://bit.ly/2cEQ6Oz> (accessed on 10 January 2022).
57. Moody, G.B.; Mark, R.G. The impact of the MIT-BIH Arrhythmia Database. *IEEE Eng. Med. Biol. Mag.* **2001**, *20*, 45–50. [[CrossRef](#)] [[PubMed](#)]
58. Goldberger, A.; Amaral, L.; Glass, L.; Hausdorff, J.; Ivanov, P.C.; Mark, R.; Mietus, J.E.; Moody, G.B.; Peng, C.K.; Stanley, H.E. PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals. *Circulation* **2000**, *101*, e215–e220. [[CrossRef](#)]
59. Raj, S.; Ray, K.C. A personalized arrhythmia monitoring platform. *Sci. Rep.* **2018**, *8*, 11395. [[CrossRef](#)]
60. Elhaj, F.A.; Salim, N.; Harris, A.R.; Swee, T.T. Arrhythmia Recognition and Classification Using Combined Linear and Nonlinear Features of ECG Signals. *Comput. Methods Programs Biomed.* **2016**, *127*, 52–63. [[CrossRef](#)]

61. Ronald, S.; Kuo, C.-C.J. ECG-based biometrics using recurrent neural networks. In Proceedings of the 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), New Orleans, LA, USA, 5–9 March 2017.
62. Darmawahyuni, A.; Nurmaini, S.; Sukemi; Caesarendra, W.; Bhayyu, V.; Rachmatullah, M.N.; Firdaus. Deep learning with a recurrent network structure in the sequence modeling of imbalanced data for ECG-rhythm classifier. *Algorithms* **2019**, *12*, 118. [[CrossRef](#)]
63. Hou, B.; Yang, J.; Wang, P.; Yan, R. LSTM Based Auto-Encoder Model for ECG Arrhythmias Classification. *IEEE Trans. Instrum. Meas.* **2020**, *69*, 1232–1240. [[CrossRef](#)]

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.