QRS Complex Detection Using Novel Deep Learning Neural Networks

WENJIE CAI AND DANQIN HU

School of Medical Instrument and Food Engineering, University of Shanghai for Science and Technology, Shanghai 200093, China

Corresponding author: Wenjie Cai (wjcai@usst.edu.cn)

ABSTRACT *Objective:* Accurate QRS complex detection is essential for electrocardiography (ECG) diagnosis. Many proposed algorithms don't perform satisfactorily on noisy and arrhythmia ECGs. The purpose of this study is to develop a noise resistant and generalizable method to detect QRS complexes accurately. *Methods:* Two deep learning models based on multi-dilated convolutional blocks are proposed. One model (CNN) is mainly composed of convolutional blocks and Squeeze-and-Excitation networks (SENet). The other model (CRNN) contains a hybrid convolutional and recurrent neural network. With 5-fold cross-validation approach the models are trained and tested on four open-access ECG databases: the China Physiological Signal Challenge (2019) database (CPSCDB), the MIT-BIH Noise Stress Test Database (NSTDB), the MIT-BIH Arrhythmia Database (MITDB) and the QT Database (QTDB). *Results:* The F1 score of CNN model on CPSCDB, NSTDB, MITDB and QTDB are 0.9929, 0.9892, 0.9994 and 0.9998 respectively. The F1 score of CRNN model on these four databases are 0.9947, 0.9953, 0.9995 and 0.9998 respectively. The ensemble of both models scored the first place in the China Physiological Signal Challenge (2019). *Conclusion:* The proposed models achieve state-of-the-art performance in QRS complex detection and show good generalization on different databases. This work might help make better ECG diagnosis.

INDEX TERMS Convolutional neural network, deep learning, electrocardiography, QRS complex, recurrent neural network.

I. INTRODUCTION

Cardiovascular diseases (CVD) are the leading cause of death globally, taking around 17.8 million lives each year [1]. Electrocardiogram (ECG) is the most widely used diagnostic tool for CVD. It is easily performed, noninvasive and can give immediate information. About 3 million ECGs are produced each day throughout the world [2]. With the development of wearable devices, more and more ECGs are generated for analysis. Automated diagnostic methods are required to process ECGs generated by wearable devices and to reduce doctors' workload. Many diagnostic methods are based on accurate QRS complex detection. QRS complexes serve as the beat positions and provide information about rhythm and intraventricular conduction. Normally they are the most prominent parts of the ECG and can be easily identified by human eyes. A lot of algorithms have been developed to automatically detect QRS complex since several decades ago. Common QRS complex detectors share a two-stage structure

including the preprocessing stage and the decision stage [3]. The preprocessing stage takes advantages of linear filtering and non-linear transformation to enhance QRS complex and attenuate other waves, noises and artifacts. The decision stage establishes the peak detection logic and additional decision rules to optimize the detection results. The popular methods include digital filtering [4], [5], wavelet transform [6]–[9], empirical mode decomposition [10], Hilbert transform [5], [8], and machine learning [11], [12]. A recent study tested ten widely used QRS detection algorithms on six ECG databases with varying degrees of noise and found that these algorithms showed very high detection accuracy on high quality ECG databases but poor accuracy on low quality ECG signals [13]. For long-term ECG monitoring, intermittent strong noise is unavoidable due to patient movement, muscle activity or even loose lead contact. It still remains a challenge to locate them accurately on noisy arrhythmic ECG.

Deep learning has been very successful in computer vision, natural language processing and speech recently. It's reported that a deep neural network achieved better than average

The associate editor coordinating the review of this manuscript and approving it for publication was Jingchang Huang.
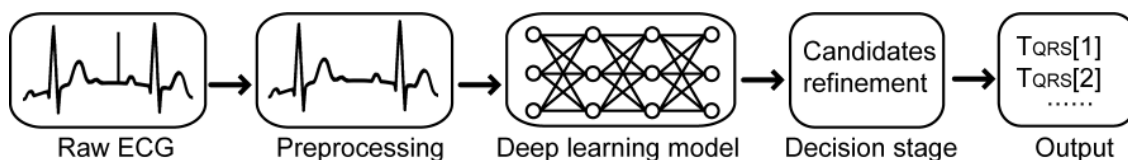
FIGURE 1. Schematic diagram for the proposed QRS complex detection method.

cardiologists in classifying 12 rhythm classes ECG [14]. And there have already been a few deep learning approaches to detect QRS complexes. Wang *et al.* proposed two parallel residual neural network (ResNet) like deep neural networks and achieved positive predictive value of 99.98% and sensitivity of 99.92% on ECG data from the MIT-BIH Arrhythmia Database (MITDB) [15]. However, they discarded 2 records and the last 4.44 seconds of each remaining record, which might attenuate the model's generalization. Xiang *et al.* constructed two-level convolutional neural network and got positive predicted value of 99.91% and sensitivity of 99.77% for the MITDB data [16]. Yang *et al.* turned one dimensional ECG into two dimensional picture and used a faster Regional CNN model to detect QRS complexes. They tested the model on 24-h wearable ECG recordings and got a sensitivity of 98.76% and a positive predictively of 98.52% [17]. These results are comparable to state-of-the-art approaches and show promising application of deep learning in QRS complex detection. However, according to the research done by Habib *et al.*, the CNN model didn't generalize well when the testing database was different from the training database [18]. A more generalized and robust QRS detector is required for real application.

The aim of this study is to propose a noise-resistant deep learning method that reaches cutting edge performance for QRS complex detection and generalizes well in different ECG databases. Our algorithms won the first place in the CPSC2019 and achieved state-of-the-art accuracies on three other common ECG databases.

II. DATABASES

The China Physiological Signal Challenge (2019) database (CPSCDB) consists of 5232 single-lead ECG recordings which were collected from patients with CVD [19]. It contains many noisy ECG excerpts together with various arrhythmia patterns. All recordings are sampled at 500 Hz and each is 10 s long. The training set has 2000 recordings and was used for training. The test set has 3232 recordings and was used for algorithm performance evaluation by the challenge committee. This database can be accessed at http://2019.icbeb.org/Challenge.html.

MITDB consists of 48 half-hour two-lead ECGs which are sampled at 360 Hz [20]. It's the most popular standard ECG database tested for QRS detection algorithms. Because there is only one record that contains ventricular flutter segments, model training and testing can't be reasonably arranged. In the same way as others [7], [10], [21] we also excluded the ventricular flutter segments in record 207 and used all 109494 beats in this study.
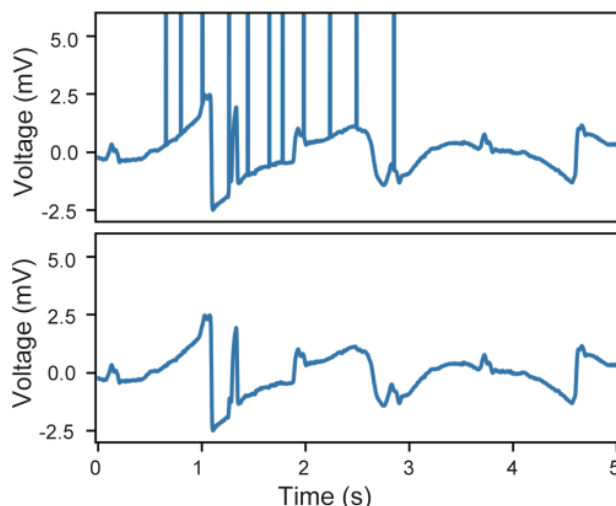

FIGURE 2. Preprocessing for removing towering spikes.

NSTDB has 12 half-hour ECG recordings. These recordings were created by adding calibrated amounts of noise to two clean records (118 and 119) fm MITDB. The noise signal was added intermittently after the first 5 min of each record. The signal-to-noise ratios (SNR) of the noisy segments are: 24, 18, 12, 6, 0, −6 dB [22].

QTDB contains 105 fifteen-minute two-lead recordings which have various QRS and ST-T morphologies. 23 records have no annotations and the remaining 82 records were selected for our research. These recordings have a sampling rate of 250 samples per second. To be consistent with most reports, the first lead data in these three databases were used in this study.

III. METHODS

The flowchart of our proposed method is shown in Figure 1. The raw ECG is preprocessed and then fed into a deep learning model. The model's output information is further judged by the decision rules to obtain the final result.

A. PREPROCESSING

Single towering spike whose voltage is more than 20 mV is examined and replaced by the normal sample immediately before it (Figure 2). These spikes are only existed in CPSCDB and the spike removal algorithm makes no change to the recordings in other three databases. All ECG recordings are resampled to 500 Hz using fast Fourier transformation method. To achieve better model generalization, the mean of signal values is subtracted for each recording. No further preprocessing such as differentiating, normalization or noise filtering is required.
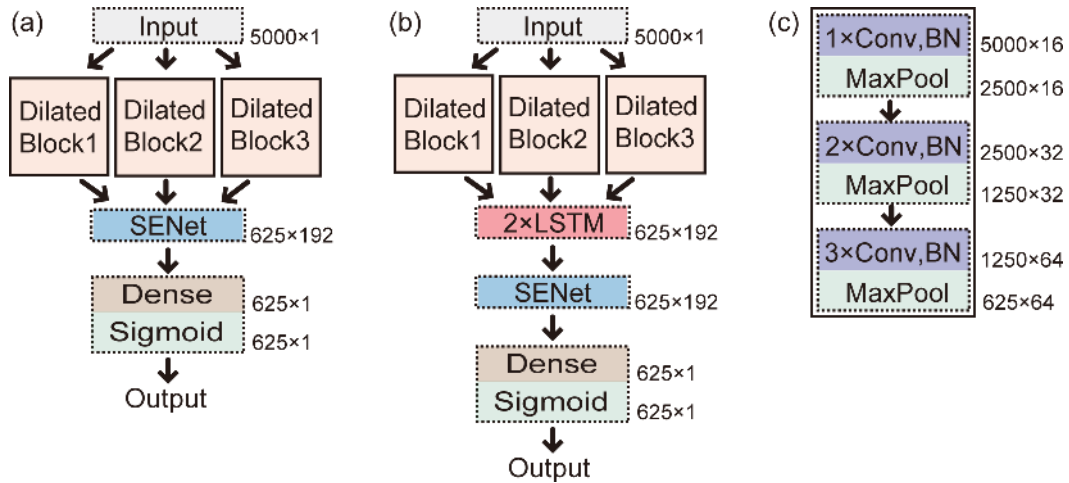
FIGURE 3. The architectures of proposed deep learning models. (a) CNN model. (b) CRNN model. (c) The architecture of each dilated block.

TABLE 1. Details of the dilated blocks.

Layer	Kernel Size	Dilation Rate		
		Block1	Block2	Block3
Conv1	11	1	2	4
Conv2	7	1	2	4
Conv3	7	1	4	8
Conv4	5	1	8	16
Conv5	5	1	8	32
Conv6	5	1	8	64

B. PROPOSED DEEP LEARNING MODELS

The proposed CNN model is shown in Figure 3a. There are three parallel dilated CNN blocks following the input layer and each convolutional block contains six 1D convolution layers (Figure 3c). The first convolution layer has the kernel size of 11. The second and third convolution layers are stacked and have the kernel size of 7. The rest three convolution layers are also stacked with the kernel size of 5. Different sets of dilation rate for convolution layers were designed for these dilated blocks. A dilation rate of 2 means a convolution takes every other point as the input. The combination of different dilation rates is to get different receptive fields for the output neurons. The details are shown in Table 1. Block 1 has the smallest receptive field size that is equivalent to 0.18 s of original samples. The receptive fields of block 2 and block 3 are 0.97 s and 3.97 s of ECG samples respectively. The batch normalization layer connected to the convolution layer is used to speed up the training process and improve generalization. The Max pooling layer following the batch normalization layer is to down-sample the features while keeping important information. Then the concatenation features extracted by convolutional blocks are fed into the squeeze-and-excitation networks (SENet) followed by three fully connected layers. The last layer uses sigmoid activation to predict QRS complexes. The size of the output layer is one-eighth the size of the input layer. So every output point

denotes 0.016 s of the original signal. And each QRS complex is expected to correspond to 7 points in the output which equal 0.112 s.

The proposed CRNN model is basically the same as the CNN model except that two stacked LSTM layers are added before the SENet (Figure 3b). LSTM layers are good at dealing with time series data. They can extract temporal features while convolution layers can only extract local features.

The input layer of both models accept variable input sizes. In current study, we input 10 s long segments of the training sets into the models during training to take full advantage of the parallel performance of the GPU. And we used the original recordings in the test sets for fast model inference.

C. MODEL TRAINING

The models are built using Keras which is a user-friendly python library for deep learning. Adam is selected as the training optimizer and its learning rate is set between 1e-3 and 1e-4. The models are trained 60 to 100 epochs with the batch size of 200. Data augmentation techniques such as adding random amount of Gaussian noise, combining a sinusoidal signal with random initial phase and amplitude [23], randomly shifting the baseline and making the signal upside down are applied on the fly to the input data. And So the models hardly see two identical inputs during training. It's very helpful for deep learning models to improve their performance and robustness.

D. PEAK LOCALIZATION

The decision stage is to localize the QRS complexes by finding the peaks from the output of the deep learning model. A fixed threshold of 0.5 is set for the output to determine whether the samples belong to a QRS complex. Another threshold of 64 ms for the duration of clustering positive samples is set to eliminate some wrong predictions. If the duration of clustering positive samples is longer than 64 ms, the midpoint of these samples is considered as a QRS complex candidate. After all the candidates are determined, the distances of adjacent candidates are calculated. If there are

two candidates whose distance is less than 100 ms, the candidate with low confidence score will be removed. The search will be repeated until all the distances between adjacent candidates are more than 100 ms. This algorithm was used in CPSC 2019. However, it may miss paced beats. To fix that, further search should be performed to locate where the distances are greater than 1200 ms between adjacent QRS complexes. For these periods, if there exists at least one point that is great than 0.5, the threshold of the duration of clustering positive samples is reduced by 16 ms and this process will continue until a new QRS candidate is found or the threshold decreases to zero.

E. PERFORMANCE EVALUATION

For CPSCDB, the accuracy of QRS location and heart rate (HR) estimations are used for performance evaluation. The evaluation algorithm, which is provided by the challenge committee, compares the reference QRS annotations and the predicted ones for every single ECG recording from 0.5 s to 9.5 s in the test set. The first and last half second is omitted. Each predicted location is deemed accurate if it lies within 75 ms duration of the reference location. When all the predicted locations and the annotated locations are totally matched, that recording scores one point. If there is only one false positive (FP) or false negative (FN) in the prediction, the recording scores 0.7 and 0.3 points respectively. For other situations, the recording scores 0 point. The detailed QRS scoring rules for a single recording are as follows:

$$QRS_{score} = \begin{cases} 1, & FP + FN = 0 \\ 0.7, & FP = 1, FN = 0 \\ 0.3, & FP = 0, FN = 1 \\ 0, & FP + FN > 1 \end{cases} \quad (1)$$

The final QRS score is calculated as follows:

$$QRS_{acc} = \frac{\sum QRS_{score}}{N} \quad (2)$$

where N is the number of test recordings.

HR is calculated between 5.5 s and 9.5 s from each recording. Its scoring rules are described in the following two equations.

$$HR_{rel} = \frac{|HR_{ref} - HR_{test}|}{HR_{ref}} \quad (3)$$

where HR_{ref} is the reference QRS location and HR_{test} is the predicted QRS location.

$$HR_{score} = \begin{cases} 1, & HR_{rel} \leq 0.02 \\ 0.75, & 0.02 \leq HR_{rel} \leq 0.05 \\ 0.5, & 0.05 \leq HR_{rel} \leq 0.1 \\ 0.25, & 0.1 \leq HR_{rel} \leq 0.2 \\ 0, & HR_{rel} > 0.2 \end{cases} \quad (4)$$

The final QRS score is calculated as follows:

$$HR_{acc} = \frac{\sum HR_{score}}{N} \quad (5)$$

where N is the number of test recordings.

Sensitivity (Se), positive predictive value (PPV), error rate (ER) and F1 are calculated in all databases. These metrics are defined as follows:

$$Se\,(\%) = \frac{TP}{TP + FN} \times 100 \quad (6)$$

$$PPV\,(\%) = \frac{TP}{TP + FP} \times 100 \quad (7)$$

$$Er\,(\%) = \frac{FP + FN}{TP + FP + FN} \times 100 \quad (8)$$

$$F1\,(\%) = \frac{2 \times Se \times PPR}{Se + PPR} \quad (9)$$

where TP is true positive and TN is true negative. The standard grace period of 150 ms is used for beat-by-beat comparison [24].

Three kinds of model evaluation strategies are used on NSTDB, MITDB and QTDB. At first we performed cross-database testing, which means the model is trained first on CPSCDB and then tested on other databases. Secondly we performed 5-fold cross-validation for MITDB and QTDB. Since the data in NSTDB are basically mixture of two records from MITDB with electrode motion (EM) artifact and some other noise, 5-fold cross-validation will lead to data leakage. We referred to Jia *et al's* method [6] to make a training set by adding the unused part of EM noise in NSTDB to the last 20 records in MITDB and then tested the models on the whole NSTDB. Lastly we performed fine-tuning, which means the model is trained first on CPSCDB and then evaluated on MITDB and QTDB with 5-fold cross-validation strategy or the specific method mentioned above for NSTDB. When 5-fold cross-validation is used, the recordings of a database are randomly split into 5 folds. The data in each unique fold used for testing is kept in an unsegmented state, whereas the recordings in the remaining folds are cut into 10 s segments, which are fed into the model during the training process.

IV. RESULTS

Two examples of using the proposed method for QRS detection are showed in Figure 4. The upper figure shows our CNN model identifies QRS complexes on an excerpt of arrhythmic ECG with drifted baseline. The lower figure shows that dynamic threshold for duration of clustering positive samples in our decision stage avoids a missed paced heartbeat detection.

A. CPSCDB

To optimize the deep learning architecture for QRS detection, different dilated CNN blocks or SENet of our proposed models were removed. Their performance was evaluated using 5-fold cross-validation (Table 2). The CNN model showed good predictive ability. Its QRS_{acc}, HR_{acc}, Se, PPV, ER and F1 were 91.23%, 94.67%, 99.26%, 99.31%, 1.42% and 0.9929 respectively. When one of the dilated blocks was removed from the proposed CNN model, the performance on all these metrics decreased. Among these three blocks, block 3 was more important, and the performance fell most

TABLE 2. QRS detection performance of different architectures on CPSCDB.

| architecture	trainable params	QRSacc (%)	HRacc (%)	Se (%)	PPV (%)	Er (%)	F1
CNN	218,969	91.23	94.67	99.26	99.31	1.42	0.9929
CNN – Block1	147,065	90.82	94.76	99.21	99.31	1.46	0.9926
CNN – Block2	147,065	91.00	94.21	99.21	99.33	1.45	0.9927
CNN – Block3	147,065	89.80	93.49	99.13	99.13	1.73	0.9913
CNN – Block1&2	73,097	89.12	93.55	99.02	99.16	1.80	0.9909
CNN– Block1&3	73,097	88.85	93.65	98.88	99.25	1.85	0.9906
CNN – Block2&3	73,097	77.81	86.53	96.65	98.72	4.55	0.9768
CNN – SENet	209,537	89.89	93.98	98.96	99.37	1.66	0.9916
CRNN	2,708,417	92.62	**96.63**	99.32	99.61	1.06	0.9947
Ensemble	2,927,386	**92.93**	96.15	**99.70**	**99.83**	**0.47**	**0.9976**
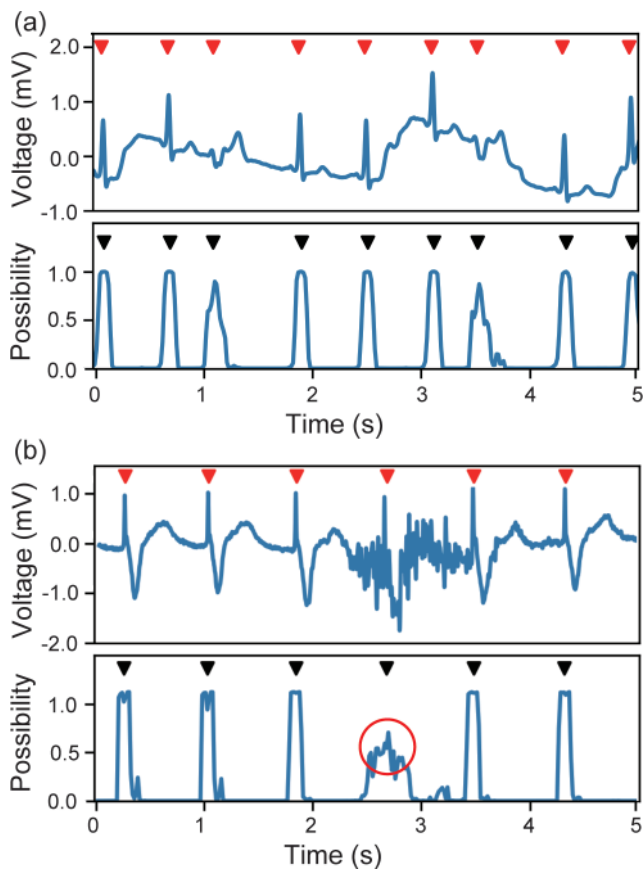
FIGURE 4. Two examples of using the proposed method for QRS detection. For each figure, the upper part is the raw ECG signal and the lower part is the model's output. Red inverted triangles indicate reference annotations. Black inverted triangles indicate predicted QRS locations. Red circle shows where dynamic threshold for duration of clustering positive samples is used in the decision stage. (a) is from recording 956 in CPSCDB and (b) is from recording 104 in MITDB.

significantly after it was removed. The lack of block 2 had the least impact on the overall performance. When two of the blocks were removed, the model performance further decreased. Retaining block 3 performed slightly better than retaining any other blocks. SENet could help the model get better overall performance. The CRNN model has two more

TABLE 3. Model performance on CPSCDB test sets.

Rank	Code No.	Team	QRSacc	HRacc
1	CPSC0416	Cai *et al.*	**92.14**	**94.89**
2	CPSC0433	Qiu *et al.*	91.55	94.29
3	CPSC0413	Zhu *et al.*	91.49	94.31
4	CPSC0436	Qi *et al.*	90.79	93.77
5	CPSC0430	Wang *et al.*	90.04	94.21
		Pan-Tompkins algorithm	35.42	55.97

stacked LSTM layers than the CNN model and it significantly improved QRS detection performance, especially QRS_{score} and HR_{score}. Ensemble model of CNN and CRNN, which averages CNN output and CRNN output, showed best in all metrics except HR_{acc}. And it was used to participate in the CPSC2019 and won the first place. As shown in Table 3, our final QRS_{acc} and HR_{acc} reached 92.14% and 94.89% which were 0.59% and 0.60% higher respectively than that of the second place.

B. NSTDB
The Se and PPV of different training strategies on NST database are shown in Table 4. In all four different cases, signals with high SNR were all well identified, whereas the performance differed significantly in signals with low SNR. The overall performance comparison of our methods with others' is shown in Table 5. Our cross-database testing performance of both models is comparable to most published results. The CNN model trained from scratch showed better results and the fine-tuned CNN model further improved its performance. The state-of-the-art result was got by the fine-tuned CRNN model which was pre-trained on CPSCDB and retrained on NSTDB.

C. MITDB
The performance of our models and some recent published results on MITDB is reported in Table 6. Cross-database testing showed the CNN model had better recognition of

TABLE 4. Model performance on NSTDB.

SNR	CNN cross-database testing		CNN cross-validation		CNN fine-tuned		CRNN fine-tuned	
	SE(%)	PPV(%)	SE(%)	PPV(%)	SE(%)	PPV(%)	SE(%)	PPV(%)
24	100.00	99.93	100.00	99.95	100.00	99.95	100.00	99.98
18	99.95	99.65	100.00	99.79	100.00	99.95	100.00	99.98
12	99.48	97.41	100.00	99.44	100.00	99.74	100.00	99.95
6	97.80	93.00	100.00	98.43	99.95	99.12	99.98	99.86
0	93.29	88.21	99.62	96.04	99.51	97.88	99.70	99.25
-6	82.77	80.01	96.51	91.51	96.37	94.66	98.34	97.42

TABLE 5. Comparison of the overall QRS detection performance on NSTDB.

Methods	Se(%)	PPV(%)	ER(%)	F1
CNN cross-database testing	95.55	92.93	10.92	0.9422
CNN cross-validation	99.36	97.47	3.14	0.9840
CNN fine-tuned	99.30	98.54	2.14	0.9892
CRNN cross-database testing	95.18	92.62	11.53	0.9388
CRNN fine-tuned	**99.67**	**99.40**	**0.92**	**0.9953**
Pan-Tompkins algorithm	94.99	81.83	21.56	0.8792
Jia et al.[6]	99.25	96.31	4.55	0.9776
Nayak et al.[5]	95.23	94.41	10.41	0.9482
Merah et al.[25]	95.30	93.98	10.81	0.9464
Lee et al.[26]	94.08	93.45	12.57	0.9376

TABLE 6. Comparison of the QRS detection performance on MITDB.

Methods	Se(%)	PPV(%)	ER(%)	F1
CNN cross-database testing	99.91	99.90	0.19	0.9991
CNN cross-validation	99.93	99.89	0.18	0.9991
CNN fine-tuned	99.95	99.94	0.11	0.9994
CRNN cross-database testing	99.83	99.87	0.30	0.9985
CRNN fine-tuned	99.94	**99.97**	**0.09**	**0.9995**
Pan-Tompkins algorithm	99.19	99.41	1.39	0.9930
Jia et al.[6]	99.89	99.90	0.21	0.9989
Chin et al.[21]	99.85	99.85	0.29	0.9985
Sharma et al.[7]	99.89	99.83	0.29	0.9986
Hossain et al.[10]	**99.96**	99.89	0.15	0.9992

TABLE 7. Comparison of the QRS detection performance on QTDB.

Methods	Se(%)	PPV(%)	ER(%)	F1(%)
CNN cross-database testing	99.97	99.92	0.11	0.9995
CNN cross-validation	**99.98**	99.97	0.05	**0.9998**
CNN fine-tuned	99.97	**99.99**	**0.04**	**0.9998**
CRNN cross-database testing	99.93	99.84	0.23	0.9989
CRNN fine-tuned	99.97	**99.99**	**0.04**	**0.9998**
Pan-Tompkins algorithm	99.57	98.34	2.08	0.9895
Nayak et al.[5]	**99.98**	99.96	0.07	0.9997
Hossain et al.[10]	99.97	99.93	0.11	0.9994
Chin et al.[21]	99.96	99.94	0.10	0.9995
Lee et al. [26]	99.96	99.96	0.08	0.9996

QRS complexes than the CRNN model. Its Se, PPV, ER and F1 are 99.91%, 99.90%, 0.19% and 0.9991 respectively. Cross-validation results of the CNN model showed similar results. The fine-tuned CNN model outperformed other published methods in terms of all four types of metrics. The fine-tuned CRNN model got the best overall performance with Se of 99.94%, PPV of 99.97%, ER of 0.09% and F1 of 0.9995.

D. QTDB

The performance comparison of our methods with others' is shown in Table 7. Cross-database testing results of the CRNN model were only better than Pan-Tompkins algorithm while the results of the CNN model were comparable to other reported results. In other situations our CNN model and CRNN model reached the new high evaluation score with F1 of 0.9998. They were superior to known published results from various methods.

V. DISCUSSION

In this report, we introduced two novel deep learning models that could perform accurate, robust and noise-resistant QRS complex detection. Unlike many algorithms whose performance decreased significantly when tested on different

databases [13], our methods were validated on a challenge database and three commonly used database and showed good generalization. Deep learning models normally get better results when they were trained with more data. However, the model's generalization capacity does not increase by simply adding more similar ECG samples [18]. That's because

ECG is composed of many repeated patterns and when the model has already learned them, more similar data won't help improve the model's performance. But a diverse range of subjects are useful [18] because they have many new patterns and help improve the model's robustness. Since CPSCDB has noisier data and the recordings are abundant with various arrhythmic patterns, it's quite different from other databases. So the model's performance can be further improved when the model is trained on a new database together with CPSCDB. And we believe with more dynamic arrhythmic ECG data, our models can achieve even better performance in QRS complex detection.

Besides the data, deep learning algorithm is critical for output results. The key part of our algorithm is the three parallel dilated CNN blocks. They share the same parameters except the dilation rates which affect the receptive field of an output neuron. A small receptive field makes the correspondence between output and input data more accurate, while a large receptive field makes the output result represent more original data. Table 3 shows that combination of three different receptive fields has optimal results. Removal of a medium-sized receptive field has less effect on the model performance than removal of a small or large receptive field. This is because the model can still take into account both local features and enough nearby information simultaneously when block 2 is removed. If only one block is left in the model, larger receptive field size is better because the identification of QRS waves in noisy ECG cannot be directly recognized from the local morphology, but requires a long period of data for comprehensive consideration. So models with three parallel convolutional blocks showed robust and outstanding performance in various ECG databases. SENet, which was introduced in ILSVRC 2017 classification challenge and won the first place [27], can further improve the end results. It introduces channel-wise attention mechanism to the output features of convolutional blocks and improves the models' performance at minimal additional computational cost.

The fine-tuned CRNN model showed powerful ability in identifying QRS complexes in different ECG databases. It increased F1 value by 0.62% in NSTDB and by 0.18% in CPSCDB compared with the fine-tuned CNN model. However, it improved performance just a little bit in MITDB and QTDB. The reason is probably related to the noise level existed in the ECG database. NSTDB has the noisiest recordings and CPSCDB contains many low signal quality recordings, while MITDB and QTDB have relatively clean ECG data. For hard ECG excerpts, it's difficult to identify QRS complexes just from their morphology. Instead, the location of the QRS complexes can be inferred from adjacent heartbeats or farther signals with high quality. The stacked LSTM layers in CRNN model are good at dealing with long sequential data because an LSTM unit has memories of previous data by controlling the information flow through three gates which are an input gate, a forget gate and an output gate [28]. Whereas, convolutional layers can only extract

TABLE 8. Comparison of models' inference speed.

Database	CNN		CRNN	
	Wall time	CPU time	Wall time	CPU time
CPSCDB	1.4	15.7	29.6	250.0
MITDB	0.7	7.5	28.6	239.4
QTDB	0.6	7.3	27.9	235.6

The unit of models' inference speed is ms/one second sample.

local morphology features and lack of information from farther signals. So the CRNN model is superior to CNN model in identifying QRS complexes of noisy ECG data. However, the cross-database testing showed that the generalization of the CRNN model was not as good as the CNN model. It implies that the CRNN model is more prone to overfitting when trained in a single database. Another disadvantage of the CRNN model is that it has 12 times more trainable parameters than the CNN model. That leads to slow model training and slow inference. Table 8 shows the models' average inference time for one second ECG sample. The wall time indicated the real time consumption and was influenced greatly by the numbers of CPUs and threads that participated in the program running. The CPU time indicated the total amount of time that used by all CPU cores for running the program. The CNN model took 15.7 ms of CPU time to process a one-second sample in CPSCDB, and it took less than half of the CPU time for samples in MITDB and QTDB. Whereas, the time consumed by the CRNN model is 17-32 times that of the CNN model. The wall time of processing a 30-min MITDB ECG recording by the CNN model is around 1.26 s and it's comparable with some conventional algorithms [29]. The CNN model can be implemented for real-time heartbeats monitoring and ECG analysis, while the CRNN model is suitable to be deployed on a workstation for static ECG analysis.

There are several limitations of our method. First, the predicted QRS locations are only approximate to the R-peak. That is because the size of the model output is only one eighth of the model input size and the final QRS locations are obtained by multiplying the peak positions in the output by eight. Second, our models can't recognize ventricular flutter. We just excluded the flutter segments of MITDB when testing our models. Third, the threshold of models' output confidence is arbitrarily set at 0.5 in the decision stage. Other values or dynamic threshold may further improve model's performace.

VI. CONCLUSION
In this paper, we proposed two models with multi-dilated convolutional blocks and tested them on various ECG databases. The CNN model runs fast, achieves high performance and generalizes well. The CRNN model makes new state-of-the-art performance on several databases but it's computationally expensive. Both models show the powerful potential of artificial intelligence in ECG analysis.

W. Cai, D. Hu: QRS Complex Detection Using Novel Deep Learning Neural Networks

IEEE Access

REFERENCES

[1] S. S. Virani, A. Alonso, E. J. Benjamin, M. S. Bittencourt, C. W. Callaway, A. P. Carson, A. M. Chamberlain, A. R. Chang, S. Cheng, F. N. Delling, and L. Djousse, "Heart disease and stroke statistics—2020 update: A report from the American Heart Association," *Circulation*, vol. 458, pp. 139–596, Jan. 2020.

[2] M. Shenasa, "Learning and teaching electrocardiography in the 21st century: A neglected art," *J. Electrocardiol.*, vol. 51, no. 3, pp. 428–429, May 2018.

[3] J. P. do Vale Madeiro, J. M. da Silva Monteiro Filho, and P. R. F. Rodrigues, "Techniques for QRS complex detection," in *Developments and Applications for ECG Signal Processing*, New York, NY, USA: Academic, 2019, pp. 89–118.

[4] J. Pan and W. J. Tompkins, "A real-time QRS detection algorithm," *IEEE Trans. Biomed. Eng.*, vol. BME-32, no. 3, pp. 230–236, Mar. 1985.

[5] C. Nayak, S. K. Saha, R. Kar, and D. Mandal, "An optimally designed digital differentiator based preprocessor for R-peak detection in electrocardiogram signal," *Biomed. Signal Process. Control*, vol. 49, pp. 440–464, Mar. 2019.

[6] M. Jia, F. Li, J. Wu, Z. Chen, and Y. Pu, "Robust QRS detection using high-resolution wavelet packet decomposition and time-attention convolutional neural network," *IEEE Access*, vol. 8, pp. 16979–16988, 2020.

[7] A. Sharma, S. Patidar, A. Upadhyay, and U. R. Acharya, "Accurate tunable-Q wavelet transform based method for QRS complex detection," *Comput. Electr. Eng.*, vol. 75, pp. 101–111, May 2019.

[8] M. Rakshit and S. Das, "An efficient wavelet-based automated R-peaks detection method using Hilbert transform," *Biocybernetics Biomed. Eng.*, vol. 37, no. 3, pp. 566–577, 2017.

[9] K. Mourad and B. R. Fethi, "Efficient automatic detection of QRS complexes in ECG signal based on reverse biorthogonal wavelet decomposition and nonlinear filtering," *Measurement*, vol. 94, pp. 663–670, Dec. 2016.

[10] M. B. Hossain, S. K. Bashar, A. J. Walkey, D. D. Mcmanus, and K. H. Chon, "An accurate QRS complex and P wave detection in ECG signals using complete ensemble empirical mode decomposition with adaptive noise approach," *IEEE Access*, vol. 7, pp. 128869–128880, 2019.

[11] K. Arbateni and A. Bennia, "Sigmoidal radial basis function ANN for QRS complex detection," *Neurocomputing*, vol. 145, pp. 438–450, Dec. 2014.

[12] A. S. A. Huque, K. I. Ahmed, M. A. Mukit, and R. Mostafa, "HMM-based supervised machine learning framework for the detection of fECG R-R peak locations," *IRBM*, vol. 40, no. 3, pp. 157–166, Jun. 2019.

[13] F. Liu, C. Liu, X. Jiang, Z. Zhang, Y. Zhang, J. Li, and S. Wei, "Performance analysis of ten common QRS detectors on different ECG application cases," *J. Healthcare Eng.*, vol. 2018, May 2018, Art. no. 9050812.

[14] A. Y. Hannun, P. Rajpurkar, M. Haghpanahi, G. H. Tison, C. Bourn, M. P. Turakhia, and A. Y. Ng, "Cardiologist-level arrhythmia detection and classification in ambulatory electrocardiograms using a deep neural network," *Nature Med.*, vol. 25, no. 1, pp. 65–69, Jan. 2019.

[15] X. Wang and Q. Zou, "QRS detection in ECG signal based on residual network," in *Proc. IEEE 11th Int. Conf. Commun. Softw. Netw. (ICCSN)*, Jun. 2019, pp. 73–77, doi: 10.1109/ICCSN.2019.8905308.

[16] Y. Xiang, Z. Lin, and J. Meng, "Automatic QRS complex detection using two-level convolutional neural network," *Biomed. Eng. OnLine*, vol. 17, no. 1, p. 13, Jan. 2018.

[17] H. Yang, M. Huang, Z. Cai, Y. Yao, and C. Liu, "A faster R CNN-based real-time QRS detector," in *Proc. Comput. Cardiol. (CinC)*, Sep. 2019, p. 1, doi: 10.22489/CinC.2019.053.

[18] A. Habib, C. Karmakar, and J. Yearwood, "Impact of ECG dataset diversity on generalization of CNN model for detecting QRS complex," *IEEE Access*, vol. 7, pp. 93275–93285, 2019.

[19] H. Gao, C. Liu, X. Wang, L. Zhao, Q. Shen, E. Y. K. Ng, and J. Li, "An open-access ECG database for algorithm evaluation of QRS detection and heart rate estimation," *J. Med. Imag. Health Informat.*, vol. 9, no. 9, pp. 1853–1858, Dec. 2019.

[20] G. B. Moody and R. G. Mark, "The impact of the MIT-BIH arrhythmia database," *IEEE Eng. Med. Biol. Mag.*, vol. 20, no. 3, pp. 45–50, May/Jun. 2001.

[21] W.-L. Chin, C.-C. Chang, C.-L. Tseng, Y.-Z. Huang, and T. Jiang, "Bayesian real-time QRS complex detector for healthcare system," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 5540–5549, Jun. 2019.

[22] G. B. Moody, W. K. Muldrow, and R. G. Mark, "A noise stress test for arrhythmia detectors," *Comput. Cardiol.*, vol. 11, no. 3, pp. 381–384, 1984.

[23] J. H. Tan, Y. Hagiwara, W. Pang, I. Lim, S. L. Oh, M. Adam, R. S. Tan, M. Chen, and U. R. Acharya, "Application of stacked convolutional and long short-term memory network for accurate identification of CAD ECG signals," *Comput. Biol. Med.*, vol. 94, pp. 19–26, Mar. 2018.

[24] *Testing and Reporting Performance Results of Cardiac Rhythm and ST Segment Measurement Algorithms*, Assoc. Advancement Med. Instrum., Arlington, VA, USA, 2012.

[25] M. Merah, T. A. Abdelmalika, and B. H. Larbic, "R-peaks detection based on stationary wavelet transform," *Comput. Methods Programs Biomed.*, vol. 121, no. 3, pp. 149–160, 2015.

[26] M. Lee, D. Park, S.-Y. Dong, and I. Youn, "A novel R peak detection method for mobile environments," *IEEE Access*, vol. 6, pp. 51227–51237, 2018.

[27] J. Hu, L. Shen, S. Albanie, G. Sun, and E. Wu, "Squeeze- and-excitation networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, Apr. 29, 2019, doi: 10.1109/TPAMI.2019.2913372.

[28] H. Sak, A. Senior, and F. Beaufays, "Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition," 2014, *arXiv:1402.1128*. [Online]. Available: http://arxiv.org/abs/1402.1128

[29] M. Kashif, S. M. Jonas, and T. M. Deserno, "Deterioration of R-wave detection in pathology and noise: A comprehensive analysis using simultaneous truth and performance level estimation," *IEEE Trans. Biomed. Eng.*, vol. 64, no. 9, pp. 2163–2175, Sep. 2017.

WENJIE CAI received the B.M. degree in basic medicine from Fudan University, China, the B.Eng. degree in computer application from Shanghai Jiao Tong University, China, and the Ph.D. degree in physiology from Fudan University.

He is currently an Associate Professor with the University of Shanghai for Science and Technology, China. His research interests include bio-signal processing, biomedical image processing, and pattern recognition.

DANQIN HU received the B.Eng. degree in biomedical engineering from Southwest Medical University, Sichuan, China. She is currently pursuing the degree with the School of Medical Instrument and Food Engineering, University of Shanghai for Science and Technology, China.

Her research interests include bio-signal processing and pattern recognition.

● ● ●

VOLUME 8, 2020

97089