ChatGPT

Heart Murmur Detection Architectures

Heart murmur classification from ~900 raw PCG recordings is challenging due to limited data and subtle audio patterns. Recent studies show that CNNs on spectrograms (e.g. ResNet) reach ~82% accuracy, but top systems (PhysioNet 2022) only achieve ~80–83% [1]. The goal (>95%) is extremely ambitious. We compare leading approaches:

- **Audio Spectrogram Transformer (AST)**: AST treats a log-mel spectrogram as an image and applies a Vision Transformer [2]. It achieved SOTA on generic audio tasks (95.6% ESC-50) [2]. For heart murmurs, transfer-learned AST gave only ~0.65 weighted accuracy [3] on the CirCor dataset, worse than SSL models. In practice, AST requires computing spectrograms and extensive fine-tuning. Its strength is modeling global context, but with 900 samples it risks overfitting. Niizumi *et al.* found supervised AST (AudioSet labels) underperformed a self-supervised model (M2D) on murmurs [3] [4]. AST may help as part of an ensemble, but alone is unlikely to hit 95%.

- **Wav2Vec 2.0**: A raw-waveform transformer pre-trained by self-supervision. Wav2Vec2 can be fine-tuned for classification. Panah *et al.* adapted Wav2Vec2 to murmurs and achieved ~0.80 weighted accuracy (UAR 0.70) [5] on CirCor data, and noted it is "robust to small fine-tuning data sizes" [6]. This suggests Wav2Vec2 features generalize to heart sounds. Implementation: feed raw audio (e.g. 4k–16k Hz) to Wav2Vec2 (or a Conformer variant) and fine-tune a classifier head. Use lower learning rate for stability [7]. Because it handles raw input, Wav2Vec2 avoids hand-crafted preprocessing. Its main trade-off is model size and computation. Given its SSL pretraining, it's a promising candidate, though published results (~80%) remain below target [1].

- **Conformers**: Conformers embed convolutional modules into transformer layers to capture both local and global features. Gulati *et al.* showed a Conformer model "significantly outperforms… previous Transformer and CNN models" on speech recognition by modeling local CNN filters and global attention [8]. A Wav2Vec2-Conformer (CNN in place of attention) similarly improved ASR error rates [9]. For murmurs, no direct results are reported, but a Conformer should theoretically better capture fine-grained murmur patterns (via convolution) plus context. It would use raw audio like Wav2Vec2. Conformers are heavy, but pretrained speech Conformers (e.g. fairseq's S2T) could be fine-tuned. In summary, Conformers likely improve upon vanilla Wav2Vec2, but still constrained by data size.

- **HSMM+RNN (Segmentation-based pipeline)**: The CinC2022-020 pipeline by McDonald *et al.* explicitly segments each heartbeat using multiple HSMMs conditioned on different murmur hypotheses. It computes a log-spectrogram (50 ms window, 0–800 Hz range) and feeds it to a bi-directional GRU whose outputs drive parallel HSMMs. The pipeline achieved 0.776 weighted accuracy (2nd place) on the PhysioNet test set [10]. This model is fully implementable from raw audio (via spectrogram pre-processing) [11]. Advantages: interpretable segmentation (identifies S1/S2/murmur intervals). However, its performance (~78%) is well below 95%. The RNN/HSMM approach effectively reduces temporal data to cycle-level features, but in this case it did not reach state-of-art. In principle, segmentation could aid a classifier by aligning beats, but by itself this pipeline is unlikely to achieve >95%. It could be used as a preprocessing step (e.g. feed each cycle's segment features into a CNN/RNN), but heavy reliance on it may limit accuracy.

- **Other Hybrid Models**: Architectures that combine CNN, RNN, and/or Transformer layers can leverage both local and temporal patterns. For example, **CNN–RNN (CRNN)** models use a CNN on spectrograms for spatial features and an RNN (LSTM/GRU) for temporal context. Zolya *et al.* (2025) built a CRNN with preprocessing (band-pass filter, normalization) and heavy augmentation (noise, time/pitch shifts) and reported ~90.5% accuracy on murmurs [12]. However, cross-validation likely overfits: their external validation on CinC2016 was only ~87.3% [13]. The survey by Alkhodari *et al.* also notes CNN–RNN hybrids (with 1D wavelet smoothing) can reach high CV accuracy but generalize around 87% [13]. **CNN–Transformer** hybrids (e.g. FAST) stack CNN front-ends with transformer blocks. FAST (2025) combined MobileNetV2-like CNN layers and transformer, achieving SOTA on audio tasks with far fewer parameters [14]. This suggests CNN–Transformer can be powerful; in practice one could adapt FAST or a similar model to spectrogram inputs. **Transformer–RNN** combos are rarer, but conceptually one could use an RNN for final classification after a Transformer encoder or vice versa. In short, hybrids often perform well on audio. Empirically, CNNs still dominate murmur tasks: Sondermann *et al.* (2025) found CNN models (AUROC ~0.795) outperform "zero-shot" audio transformers (~0.657–0.701) [15]. This indicates tuned CNNs (or CNN+time-model) currently beat vanilla Transformers on PCGs. Nonetheless, hybrids leveraging both (e.g. Conformer, FAST) may offer a good trade-off if pretrained.

Segmentation Pipeline Assessment

The HSMM+RNN segmentation pipeline is implementable from raw audio (via spectrogram features) [11]. It excels at aligning heart cycles and pinpointing murmur timing, which aids interpretability. However, its recorded accuracy (~77.6%) is far from 95% [10]. In fact, top challenge teams (using XGBoost and other features) only reached ~80% [1]. Thus a pure segmentation approach is unlikely to yield the required accuracy alone. It could be combined with DNNs (e.g. using segmented features as input to a CNN/RNN), but on its own it seems insufficient. In practice, segmentation could be used to augment feature extraction (e.g. computing cycle-based statistics or feeding state labels into a classifier), but the classification model will still need powerful learning (likely deep nets) to surpass ~80%. Given the 95% target, segmentation might best be used as auxiliary information (e.g. adding "S1/S2/murmur" timing features) rather than the final model.

Model Comparison and Input Formats

Model	Input Format	Pretraining / Data	Reported Murmur Perf.
AST (pure Transformer)	Spectrogram (image) [2]	AudioSet-sup (SL) [16]	~65% Wacc on CirCor [3]
Wav2Vec2 (Transformer)	Raw waveform [17]	Large speech SSL [17]	~80% Wacc [5]
Conformer (CNN+Transformer)	Raw waveform	Speech SSL (e.g. XLSR)	Not reported (ASR SOTA [8])

Model	Input Format	Pretraining / Data	Reported Murmur Perf.
CNN–RNN (CRNN)	Spectrogram or 1D Raw [18]	Typically ImageNet/ AudioSet SL	~90% (CV) [12], ~87% external [13]
CNN only (ResNet/EEF)	Spectrogram	ImageNet or AudioSet SL	~82% (user's test)
HSMM+RNN (segmentation)	Spectrogram [11]	–	~78% Wacc [10]
AST+CNN (e.g. FAST)	Spectrogram	AudioSet SL + novel CNN/ Transformer design [14]	General audio SOTA (unknown for murmur)

Implementation notes: AST and CNN-based models use a 2D spectrogram (often log-Mel) input [2] [11]. Wav2Vec2/Conformer take raw waveform. HSMM+RNN uses log-spectrogram (0–800 Hz) [11]. Pretraining on large audio (AudioSet, speech) is crucial [19] [1]. For small data, one should freeze most layers and train only a classifier head or fine-tune lightly.

Small-Data Strategies

With only ~900 samples, transfer learning and augmentation are key. **Transfer learning:** Use pretrained models (e.g. AudioSet CNNs, Audio Transformers, Wav2Vec2). Niizumi *et al.* showed AudioSet-supervised CNN14 and AST gave poor murmur results (58–65% Wacc) [3], while SSL models (M2D, Wav2Vec2) outperformed them [3] [1]. Fine-tuning on PCG data with a low learning rate is recommended [7]. If possible, pretrain on any unlabeled heart sound data (e.g. via an autoencoder or contrastive SSL) to adapt to this domain.

Data augmentation: Crucial for generalization. Methods include time-domain noise, shifting and scaling, and spectrogram masking (SpecAugment). Zolya *et al.* used Gaussian noise, random time-shifts, pitch shifts, etc., and emphasized that "augmentation helps balance the dataset and improve generalization" [20]. Niizumi *et al.* found SpecAugment dramatically improved performance for low-data CNNs [21]. Key augmentations: adding Gaussian noise, time shifts, pitch shifts, random cropping/masking [22] [21]. Also consider balancing classes (murmur vs normal), e.g. oversample murmurs or use synthetic minority oversampling. Standard techniques (normalization, band-pass filtering 20–1000 Hz, fixed-length padding) are also recommended [23].

Ensembling: Combining diverse models often helps. Niizumi *et al.* showed ensembles of different pretrained models (AST+M2D, CNN+M2D) further improved recall on rare classes [24] [25]. Given different models excel on different murmur types, ensembling (averaging or voting) can boost overall accuracy.

Cross-validation: Always validate with k-fold CV due to small data. Report metrics like weighted accuracy and UAR. Be wary of overfitting: very high CV accuracy (e.g. 99%) can collapse on held-out sets [13].

Recommendations

Given these findings, we recommend the following approach:

1. **Pretrained audio Transformer (spectrogram-based):** Fine-tune a high-capacity model like AST (Vision Transformer on log-mel spectrogram) or the FAST CNN-Transformer hybrid [14]. Start from weights pretrained on AudioSet or a similar large audio corpus. Use low learning rate and heavy regularization. This leverages the AST's strength in capturing global patterns [2].

2. **Self-supervised waveform model:** Fine-tune Wav2Vec2 or a Conformer on the raw PCG signals [5] [8]. Wav2Vec2 is already pre-trained on speech and known "robust to small data" [6]. Consider using the Conformer variant for added CNN-style context [8]. Freeze most layers initially, train only a linear head. If possible, continue SSL pretraining on the heart sounds themselves before supervised fine-tuning.

3. **CNN–RNN model with augmentation:** Develop a CRNN on spectrograms, following Zolya *et al.* [12]. Apply aggressive augmentations (noise, time/pitch shifts, SpecAugment [22] [21]). This model can be shallower to avoid overfitting, or use pretrained CNN backbone (e.g. PANNs) to reduce parameter count. Monitor generalization (87% ext. validation in literature [13]), and use ensembling or dropout to mitigate overfit.

4. **Segmentation-aware features:** Optionally incorporate a segmentation module (CNN/RNN) to label S1/S2/murmur phases. Use those labels as auxiliary inputs or to align features. For example, pass spectrogram through a pretrained HSMM/RNN to generate segment probabilities [11], then concatenate these with CNN features. This could help the classifier focus on murmur intervals, though it alone won't reach 95% [10].

5. **Ensemble diverse models:** Combine the above models (e.g. Wav2Vec2+CNN-RNN+AST) by averaging probabilities. Ensembles improved accuracy in related studies [24] [25], especially on hard classes. This is likely necessary to approach high accuracy.

6. **Preprocessing and normalization:** Standardize audio (e.g. 4 kHz resampling, band-pass 20–1000 Hz [23]) and normalize amplitude. Use uniform clip lengths (e.g. 10–15 s zero-padded) as in Zolya *et al.* [26]. Consistent preprocessing helps all models.

In summary, a hybrid strategy is advised: leverage **pretrained self-supervised transformers** (Wav2Vec2/Conformer on raw waveforms, or AST on spectrograms) because they have shown strong general-purpose audio representations [2] [19]. Complement them with a **CNN–RNN** model to capture fine details and with aggressive **data augmentation** [20] [21]. Incorporate segmentation cues if feasible, and ensemble for robustness. Even with these, >95% is unprecedented; one should set realistic goals (~85–90%) and use cross-validation to avoid overfitting. Critical factors are transfer learning from large audio/speech data and maximizing variability through augmentation [6] [21].

Sources: We drew on recent literature on heart sound classification, including AST (Gong *et al.*, 2021) [2], Wav2Vec2 adaptation (Panah *et al.*, EUSIPCO 2023) [5], Conformer (Gulati *et al.*, 2020) [8], the HSMM+RNN pipeline (McDonald *et al.*, CinC 2022) [10], and surveys comparing CNN/Transformer models (Niizumi *et al.*, 2024 [1]; Sondermann *et al.*, 2025 [15]). Data augmentation and transfer learning best-practices are also drawn from these and related works [20] [21].

---

[1] [3] [4] [16] [19] [21] [24] [25] arxiv.org
https://arxiv.org/pdf/2404.17107

[2] [7] Audio Spectrogram Transformer
https://huggingface.co/docs/transformers/en/model_doc/audio-spectrogram-transformer

[5] [6] [17] eurasip.org
https://eurasip.org/Proceedings/Eusipco/Eusipco2023/pdfs/0001010.pdf

[8] Paper page - Conformer: Convolution-augmented Transformer for Speech Recognition
https://huggingface.co/papers/2005.08100

[9] Wav2Vec2-Conformer
https://huggingface.co/docs/transformers/en/model_doc/wav2vec2-conformer

[10] [11] cinc.org
https://www.cinc.org/archives/2022/pdf/CinC2022-020.pdf

[12] [20] [22] [23] [26] AI-Enhanced Detection of Heart Murmurs: Advancing Non-Invasive Cardiovascular Diagnostics - PMC
https://pmc.ncbi.nlm.nih.gov/articles/PMC11945174/

[13] [18] Deep Learning in Heart Sound Analysis: From Techniques to Clinical Applications - PMC
https://pmc.ncbi.nlm.nih.gov/articles/PMC11461928/

[14] FAST: Fast Audio Spectrogram Transformer © 2025 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.
https://arxiv.org/html/2501.01104v1

[15] [2507.07058] Comparative Analysis of CNN and Transformer Architectures with Heart Cycle Normalization for Automated Phonocardiogram Classification
https://arxiv.org/abs/2507.07058