Proceedings of CONF-SPML 2027 Symposium: International Symposium on Agentic and Trustworthy Learning for Autonomous Systems
DOI: 10.54254/2755-2721/2027.SE39184

Evaluating Full-Track Log-Mel Features and a CNN-BiGRU Model for Music Genre Classification Using the FMA-Small Dataset

Liyu Deng

School of Computer Science and Technology, Tongji University, Shanghai, China
2554198@tongji.edu.cn

Abstract. Automatic music genre classification is complicated by overlapping genre boundaries and by distribution differences between development and test data. This study evaluates whether expanding temporal coverage and recurrent modeling improves a compact convolutional baseline on the official FMA-small split. All 8,000 thirty-second clips are represented as 128-bin log-Mel spectrograms. The baseline CNN uses a fixed 512-frame excerpt, whereas the proposed system retains the full track, samples random 512-frame crops during training, uses a bidirectional gated recurrent unit after convolutional feature extraction, and averages five deterministic crops at inference. SpecAugment, mixup, class-weighted cross-entropy, AdamW, and cosine learning-rate decay are applied without changing the official 6,400/800/800 train-validation-test partition. The proposed system raises validation accuracy from 46.50% to 62.25% and validation macro-F1 from 43.93% to 59.62%. However, test accuracy changes from 48.75% to 47.63%, while test macro-F1 increases modestly from 45.19% to 47.30%. The resulting 14.62-percentage-point validation-to-test accuracy gap shows that the optimization package improved development-set performance but did not establish stronger overall generalization. The work therefore contributes a reproducible, leakage-aware evaluation and demonstrates why final claims must be based on held-out test behavior rather than validation gains alone.

Keywords: music genre classification, FMA-small, log-Mel spectrogram, CNN-BiGRU, macro-F1

1. Introduction

Automatic music genre classification (MGC) supports music retrieval, catalog organization, playlist generation, and recommendation. The task is nevertheless challenging because a genre is not defined by a single acoustic property. Instrumentation, rhythm, harmony, production style, and cultural context overlap across labels, while recordings assigned to the same genre may vary considerably. FMA-small provides a public and balanced benchmark with 8,000 thirty-second clips in eight top-level genres and an official train-validation-test partition [1]. Its manageable scale makes it suitable for controlled experiments, while its real-world audio still exposes the generalization problems of genre recognition.

© 2026 The Authors. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

DOI: 10.54254/2755-2721/2027.SE39184

Two broad approaches have been widely used for music genre classification. Traditional systems summarize a track with handcrafted descriptors such as Mel-frequency cepstral coefficients (MFCCs) and classify the resulting vectors. These systems are computationally compact, but global statistical pooling removes temporal order. Deep models instead learn patterns from time-frequency representations. Log-Mel spectrograms can be generated reproducibly with established audio-processing software, and convolutional recurrent neural networks (CRNNs) can combine local spectral modeling with sequential context [2, 3]. Data augmentation methods such as mixup and SpecAugment are often used to reduce sensitivity to particular examples [4, 5]. AdamW further separates weight decay from the adaptive gradient update [6]. PyTorch provides the implementation framework used here [7].

Recent audio research has increasingly emphasized Transformer and self-supervised encoders. Hierarchical token-semantic models, Patchout-based training, masked audio autoencoding, self-supervised spectrogram transformers, acoustic-token pretraining, audio-language contrastive learning, large-scale music pretraining, and denoising transformers for genre classification all demonstrate the value of broad pretraining or efficient long-context modeling [8-15]. However, a small supervised CRNN remains useful as a baseline model that can be trained with moderate computational resources and analyzed under a fixed split. It also allows the effect of temporal coverage to be studied without importing large external corpora.

The experiment grew from a practical limitation in the first CNN run: only the opening 512 frames of each 30s track were used, and event order disappeared after global pooling. Therefore, this study evaluates whether broader temporal coverage and a bidirectional gated recurrent unit (BiGRU) could improve classification under the same official split. The comparison includes the 26-dimensional MFCC linear support vector machine (SVM) reference, the fixed-crop CNN, and the full-track CNN-BiGRU. This study focuses on the difference between validation behavior and the final held-out result, rather than on claiming state-of-the-art performance.

2. Materials and methods

2.1. Dataset source

The experiment uses FMA-small, comprising 8,000 clips of about 30s from Electronic, Experimental, Folk, Hip-Hop, Instrumental, International, Pop, and Rock [1]. Each official evaluation split contains 100 tracks per genre, so accuracy and macro-F1 can be compared without the influence of class imbalance in validation or test data. The official split is preserved: 6,400 tracks for training, 800 for validation, and 800 for testing. One MFCC preprocessing run retained 7,994 valid files because six audio files could not be used. Its training partition therefore contains 6,394 tracks, while both evaluation partitions remain at 800. No test item is used for feature standardization, hyperparameter choice, or checkpoint selection.

2.2. Audio preprocessing and feature extraction

Every track is converted to mono and resampled to 22,050 Hz. For waveform $x[n]$, the windowed short-time Fourier transform (STFT) at frame m and frequency bin k is defined in equation (1).

$$X(m,k) = \sum_n x[n]w[n - mH]\exp(-j\frac{2\pi kn}{N}) \quad (1)$$

DOI: 10.54254/2755-2721/2027.SE39184

In equation (1), $X(m,k)$ is the complex STFT coefficient, $x[n]$ is the waveform sample at discrete index n, and m and k denote the frame and frequency-bin indices, respectively. Here, w is the analysis window, H is the hop size, j is the imaginary unit, and N is the transform length. The power spectrum is projected onto $B = 128$ triangular Mel filters $M_b(k)$ as shown in equation (2).

$$E(m,b) = \sum_k |X(m,k)|^2 M_b(k), \qquad b = 1, \ldots, 128 \tag{2}$$

In equation (2), $E(m,b)$ is the Mel-filter energy at frame m and Mel-band index b, while M_b (k) is the response of the b th Mel filter at frequency bin k.

A small positive constant ε prevents the logarithm from becoming undefined. The input feature is the normalized log-Mel value in equation (3).

$$Z(m,b) = \frac{\log(E(m,b)+\varepsilon) - \mu_{\text{train}}}{\sigma_{\text{train}}} \tag{3}$$

Here, μ_{train} and σ_{train} denote the mean and standard deviation estimated from the training data, respectively. A full 30s clip produces approximately 1,292 frames. A crop $C_s(Z)$ begins at temporal offset s and contains 512 frames. The original CNN always retained the first crop, covering only about 11.9s. The proposed training procedure samples s randomly at each epoch, increasing the portions of each track seen across training without enlarging the per-step tensor.

Substituting equation (1) into equation (2), followed by the logarithmic compression and training-set normalization in equation (3), gives the complete waveform-to-feature mapping in equation (4).

$$Z(m,b) = \frac{\log\left[\sum_k \left|\sum_n x[n] w[n-mH] \exp\left(-j\frac{2\pi k n}{N}\right)\right|^2 M_b(k) + \varepsilon\right] - \mu_{\text{train}}}{\sigma_{\text{train}}} \tag{4}$$

Equation (4) shows that each network input value is derived through four consecutive operations: local Fourier analysis, squared-magnitude energy calculation, Mel-frequency aggregation, and normalized logarithmic compression. The logarithm reduces the dynamic range of spectral energy, while normalization prevents absolute recording level from dominating the learned representation.

2.3. Baselines and proposed network

The traditional baseline computes 13 MFCCs per frame, then concatenates the temporal mean and standard deviation of each coefficient into a 26-dimensional vector. Training-set standardization is followed by a linear soft-margin SVM. For a binary class pair, the primal objective is given in equation (5).

$$\begin{aligned} \min_{\mathbf{w},b,\boldsymbol{\xi}} \quad & \tfrac{1}{2} \parallel \mathbf{w} \parallel^2 + C \textstyle\sum_i \xi_i \\ \text{subject to} \quad & y_i \left(\mathbf{w}^{\mathsf{T}} \mathbf{x}_i + b\right) \geq 1 - \xi_i, \qquad \xi_i \geq 0 \end{aligned} \tag{5}$$

Here, $\mathbf{w}$ is the weight vector, b is the bias term, ξ_i is the slack variable for sample i, C is the regularization parameter, $\mathbf{x}_i$ is the feature vector, and y_i is the corresponding class label. The experiment uses $C = 1$. Sequential minimal optimization solves 28 one-versus-one binary problems for the eight genres. This baseline provides a reference for evaluating the performance of compact, globally pooled timbral features.

DOI: 10.54254/2755-2721/2027.SE39184

The baseline CNN contains three convolutional blocks with 32, 64, and 128 channels, batch normalization, rectified linear units, and pooling. Adaptive global average pooling creates a 128-dimensional embedding, and a linear layer produces eight logits. The proposed network keeps the convolutional front end but preserves its temporal sequence for a BiGRU. If $\overrightarrow{h_t}$ and $\overleftarrow{h_t}$ denote forward and backward recurrent states, the sequence representation is defined in equation (6).

$$H = \{[\overrightarrow{h_t}; \overleftarrow{h_t}]\}_{t=1}^{T}, \qquad \mathbf{z} = W_o \mathrm{Pool}(H) + \mathbf{b}_o \tag{6}$$

In equation (6), H is the recurrent-state sequence, t is the time-step index, T is the sequence length, and W_o and b_o are the output-layer weight matrix and bias vector, respectively. Here, $\mathrm{Pool}(\cdot)$ converts the recurrent sequence to a track-level vector, and $\mathbf{z}$ contains eight class logits. This formulation allows local convolutional evidence to be interpreted using both earlier and later temporal context before classification.

2.4. Optimization and regularization

Random time and frequency masks are applied to the selected training crop following the general SpecAugment principle [5]. Mixup then combines two examples $(\mathbf{x}_i, \mathbf{y}_i)$ and $(\mathbf{x}_j, \mathbf{y}_j)$. With λ sampled from a beta distribution, the augmented sample is constructed in equation (7).

$$\tilde{\mathbf{x}} = \lambda \mathbf{x}_i + (1-\lambda)\mathbf{x}_j, \qquad \tilde{\mathbf{y}} = \lambda \mathbf{y}_i + (1-\lambda)\mathbf{y}_j \tag{7}$$

In equation (7), i and j index the two selected samples, while $\tilde{x}$ and $\tilde{y}$ denote the mixed input and mixed target, respectively. For class weights ω_c and predicted probabilities p_{ic}, the weighted cross-entropy is defined in equation (8).

$$L = -\frac{1}{N} \sum_i \sum_c \omega_c \tilde{y}_{ic} \log p_{ic} \tag{8}$$

In equation (8), L is the weighted cross-entropy loss, N is the number of samples, i and c index samples and classes, and $\tilde{y}_{ic}$ is the mixed target for sample i and class c.

The optimizer is AdamW [6]. Cosine learning-rate decay is applied, and the run was completed for 60 epochs. One complete training run took approximately seven hours. This cost limited the number of configurations and random seeds that could be tested within the project schedule. During validation and testing, five evenly spaced crops are processed, and their logits are averaged as shown in equation (9).

$$\bar{\mathbf{z}}(x) = \frac{1}{5} \sum_{r=1}^{5} f_\theta(C_{s_r}(Z_x)), \qquad \hat{y} = \operatorname*{argmax}_c \bar{z}_c(x) \tag{9}$$

In equation (9), f_θ is the model with parameters θ, r indexes the five crops, and c is the class index. The expression $\bar{z}_c$ (x) is the averaged logit for class c, and $arg\,max_c$ selects the class with the highest averaged logit as the predicted label $\hat{y}$.

A checkpoint is selected only from validation results, using the mean of accuracy and macro-F1 rather than either metric alone, as defined in equation (10).

$$S_{\mathrm{val}} = \frac{1}{2}\left(\mathrm{Accuracy}_{\mathrm{val}} + \mathrm{MacroF1}_{\mathrm{val}}\right) \tag{10}$$

DOI: 10.54254/2755-2721/2027.SE39184

Here, S_{val} is the validation-based checkpoint-selection score, $\text{Accuracy}_{\text{val}}$ is validation accuracy, and $\text{MacroF1}_{\text{val}}$ is validation macro-F1. The factor $1/2$ gives the two metrics equal weight.

2.5. Evaluation measures

For N examples, accuracy is the fraction of correct predictions. For class c, precision P_c and recall R_c give $F1_c = 2P_cR_c/(P_c + R_c)$. Macro-F1 is the unweighted average of the eight class-level F1 values. The two reported measures are summarized in equation (11).

$$\text{Accuracy} = \frac{1}{N}\sum_i 1(\hat{y}_i = y_i), \qquad \text{MacroF1} = \frac{1}{8}\sum_{c=1}^{8} F1_c \tag{11}$$

In equation (11), i and c index samples and classes, $1(\cdot)$ is the indicator function, $\hat{y}_i$ is the predicted label, y_i is the true label for sample i, and $F1_c$ is the F1 score for class c.

Because the official evaluation subsets are balanced, accuracy measures total correctness directly, while macro-F1 remains important because it reveals whether performance is distributed more evenly across genres. All reported CNN values come from completed runs.

The settings of the baseline CNN and the proposed CNN-BiGRU are summarized in Table 1.

Table 1. Comparison of the baseline CNN and proposed CNN-BiGRU configurations

Component	Baseline CNN	Proposed CNN-BiGRU
Input	128 × 512 fixed log-Mel crop	Full 128 × approximately 1,292 log-Mel with a 512-frame sampled crop
Temporal model	Global adaptive average pooling	Bidirectional GRU after CNN features
Training	Cross-entropy, AdamW, 20 epochs	Weighted loss, AdamW with cosine decay, 60 epochs
Augmentation	Not documented	SpecAugment and mixup
Inference	Single crop	Five deterministic crops with averaged logits
Selection	Best validation macro-F1	Mean of validation accuracy and macro-F1

3. Results and discussion

3.1. Overall model comparison

The overall validation and test results are summarized in Table 2. The linear SVM using 26-dimensional MFCC features achieves 45.75% validation accuracy and 44.28% validation macro-F1, but its test scores decrease to 34.38% and 33.77%, respectively. The original CNN reaches 46.50% validation accuracy and 43.93% validation macro-F1. Unlike the SVM, it performs better on the test split, reaching 48.75% accuracy and 45.19% macro-F1.

The proposed full-track CNN-BiGRU achieves its best validation performance at epoch 60. Validation accuracy rises to 62.25%, an absolute gain of 15.75 percentage points over the original CNN. Validation macro-F1 rises from 43.93% to 59.62%, a gain of 15.69 percentage points. However, the held-out test result is materially different. Test accuracy is 47.63%, 1.12 percentage

DOI: 10.54254/2755-2721/2027.SE39184

points below the original CNN, while test macro-F1 is 47.30%, only 2.11 percentage points above the original CNN. The quantitative comparison of accuracy and macro-F1 is shown in Figure 1.

Table 2. Performance on the official FMA-small validation and test subsets

Model and split	Representation	Accuracy	Macro-F1
Linear SVM, validation	26-D MFCC	45.75%	44.28%
Linear SVM, test	26-D MFCC	34.38%	33.77%
Original CNN, validation	128 × 512 log-Mel	46.50%	43.93%
Original CNN, test	128 × 512 log-Mel	48.75%	45.19%
CNN-BiGRU, validation	Full track with sampled crop	62.25%	59.62%
CNN-BiGRU, test	Five-crop average	47.63%	47.30%

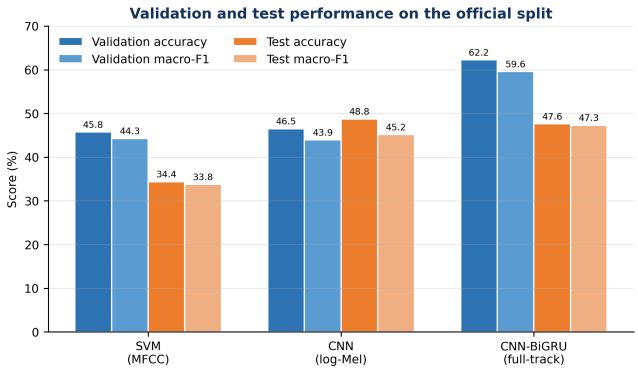


Figure 1. Quantitative comparison of accuracy and macro-F1. Values are percentages (picture credit: original)

3.2. Generalization analysis

The central quantitative result is the difference between validation and test performance. For the proposed model, the validation-to-test accuracy gap is $62.25 - 47.63 = 14.62$ percentage points. The corresponding macro-F1 gap is $59.62 - 47.30 = 12.32$ percentage points. In contrast, the original CNN test accuracy exceeds its validation accuracy by 2.25 percentage points, and its test macro-F1 exceeds validation by 1.26 percentage points. Therefore, the proposed system clearly fits the validation distribution better, but the evidence does not support a broad claim that it recognizes new tracks more accurately.

Macro-F1 adds a more nuanced observation. On the test split, the proposed model achieves lower accuracy but higher macro-F1 than the original CNN. The gap between its test accuracy and macro-F1 is only 0.33 percentage points, compared with 3.56 percentage points for the original CNN. Because the eight test genres are balanced, this convergence suggests that the proposed system distributes correct predictions somewhat more evenly across classes. Nevertheless, per-class CNN predictions were not retained in the available result record, so the paper does not attribute the improvement to particular genres.

DOI: 10.54254/2755-2721/2027.SE39184

3.3. Error structure of the handcrafted reference

The linear-SVM confusion matrix provides an interpretable error reference. Hip-Hop obtains the highest test class F1 at 56.22%, while Experimental obtains the lowest at 11.76%. The largest directed errors are Folk classified as Experimental for 35 of 100 tracks, Pop classified as Electronic for 29 tracks, and Instrumental classified as Folk for 28 tracks. These patterns are consistent with the limited information in a globally pooled MFCC vector. Mean and standard deviation describe spectral envelope statistics but discard event sequence, arrangement, and long-range rhythmic form. This explanation is plausible rather than causal. The confusion counts alone cannot establish which musical property produced each error.

3.4. Effects and limitations of the optimization strategies

The proposed design addresses identifiable limitations of the fixed-crop CNN. Random cropping exposes different parts of a track over successive epochs. The BiGRU retains temporal order after local convolutional features are extracted. Five-crop averaging reduces dependence on one excerpt. SpecAugment and mixup enlarge the effective training distribution. The simultaneous rise of validation accuracy and macro-F1 indicates that the model did not obtain its validation gain solely by favoring already easy genres.

However, these interventions were introduced as one package. The experiment cannot isolate the contribution of random cropping, recurrence, augmentation, class weighting, cosine scheduling, or multi-crop inference. Selecting the best checkpoint from repeated evaluations on the same validation set may also lead to over-adaptation to the validation data. The test result is therefore the appropriate basis for the final comparative claim.

3.5. Why validation and test results diverged

Several mechanisms may explain the accuracy gap of 14.62 percentage points. First, the larger temporal model has greater capacity and a longer optimization schedule than the baseline. Second, the selection score was evaluated at every epoch, making the selected checkpoint a function of repeated observations of the same validation distribution. Third, the training view is stochastic, whereas evaluation uses five fixed temporal views. Although multi-crop averaging is more stable than one crop, five excerpts may still miss discriminative events or emphasize sections that differ systematically across splits. Finally, FMA genre labels are track-level cultural categories. A short acoustic excerpt may be ambiguous even when the whole recording has a clear catalog label.

The result illustrates a general methodological point. Validation metrics are decision-making instruments, not final estimates of deployment performance. Reporting only the 62.25% validation accuracy would make the optimization look successful but would omit the held-out evidence. Reporting both validation and test results provides a more complete evaluation: the proposed model improves validation performance and slightly increases test macro-F1, but does not improve overall test accuracy.

3.6. Threats to validity

Only one completed run is available for each CNN configuration, so variability across random seeds could not be assessed. The main practical reason was computation time: the 60-epoch optimized run required approximately seven hours. The experiment also changed several components together,

DOI: 10.54254/2755-2721/2027.SE39184

which prevents a causal attribution of the validation gain. In addition, per-class predictions for the optimized model were not retained, and the study covers only FMA-small. Therefore, the reported results should be interpreted with caution, as their stability across different random seeds and datasets has not been established.

4. Conclusion and future work

The proposed CNN-BiGRU improved validation performance but did not improve held-out test accuracy, although it produced a modest gain in test macro-F1. This discrepancy shows that validation improvements alone do not demonstrate better generalization.

Future work should isolate individual optimization components through ablation studies, evaluate stability across random seeds, incorporate complementary musical descriptors, and examine transfer learning and cross-dataset generalization.

References

[1] Defferrard, M., Benzi, K., Vandergheynst, P., & Bresson, X. (2017). FMA: A dataset for music analysis. *Proceedings of the 18th International Society for Music Information Retrieval Conference (ISMIR)*, 316-323.

[2] McFee, B. et al. (2015). librosa: Audio and music signal analysis in Python. *Proceedings of the 14th Python in Science Conference*, 18-24.

[3] Choi, K., Fazekas, G., Sandler, M., & Cho, K. (2017). Convolutional recurrent neural networks for music classification. *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2392-2396.

[4] Zhang, H., Cisse, M., Dauphin, Y. N., & Lopez-Paz, D. (2018). mixup: Beyond empirical risk minimization. *International Conference on Learning Representations (ICLR)*.

[5] Park, D. S., Chan, W., Zhang, Y., Chiu, C.-C., Zoph, B. et al. (2019). SpecAugment: A simple data augmentation method for automatic speech recognition. *Interspeech*, 2613-2617.

[6] Loshchilov, I., & Hutter, F. (2019). Decoupled weight decay regularization. *International Conference on Learning Representations (ICLR)*.

[7] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J. et al. (2019). PyTorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems, 32*, 8024-8035.

[8] Chen, K., Du, X., Zhu, B., Ma, Z., Berg-Kirkpatrick, T. et al. (2022). HTS-AT: A hierarchical token-semantic audio transformer for sound classification and detection. *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 646-650.

[9] Koutini, K., Schlüter, J., Eghbal-zadeh, H., & Widmer, G. (2022). Efficient training of audio transformers with patchout. *Interspeech*, 2753-2757.

[10] Huang, P.-Y., Xu, H., Li, J., Baevski, A., Auli, M. et al. (2022). Masked autoencoders that listen. *Advances in Neural Information Processing Systems, 35*, 28708-28720.

[11] Gong, Y., Lai, C.-I., Chung, Y.-A., & Glass, J. (2022). SSAST: Self-supervised audio spectrogram transformer. *Proceedings of the AAAI Conference on Artificial Intelligence, 36*, 10699–10709.

[12] Chen, S., Wu, Y., Wang, C., Liu, S., Tompkins, D., et al. (2023). BEATs: Audio pre-training with acoustic tokenizers. *Proceedings of the 40th International Conference on Machine Learning, 202*, 5178–5193.

[13] Elizalde, B., Deshmukh, S., Al Ismail, M., & Wang, H. (2023). CLAP: Learning audio concepts from natural language supervision. *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1–5.

[14] Li, Y., Yuan, R., Zhang, G., Ma, Y., Chen, X., et al. (2024). MERT: Acoustic music understanding model with large-scale self-supervised training. *International Conference on Learning Representations (ICLR)*.

[15] Wang, J., Li, S., & Sung, Y. (2023). Deformer: Denoising transformer for improved audio music genre classification. *Applied Sciences, 13*, 12673.