Genre Classification with Convolutional Neural Networks

Arjun Bahuguna
arjun.bahuguna01@estudiant.upf.edu

15 November 2025

Abstract

This submission implements genre classification with a convolutional neural network on the MagnaTagATune dataset [1], and compares it against a multilayer perceptron baseline. Overall, the CNN outperforms the MLP throughout training, showing lower losses and higher validation accuracy. The MLP exhibits weaker generalization, reflecting its limited capacity to model the temporal–spectral structure of audio features. Evaluation results confirm this pattern, with the CNN achieving 20% better ROC-AUC score on the test set. These findings highlight the importance of architectural choices, demonstrating that convolutional models are better suited than simple feed-forward networks for music genre classification.

Contents

1 Objective

The task requires training a genre classification system on the MagnaTagATune [1] dataset using two architectures: a multilayer perceptron (MLP) and a convolutional neural network (CNN). For both models, we record training loss, validation loss, and validation accuracy for each epochs, and report final test-set accuracy.

2 Dataset

The MagnaTagATune dataset contains 25,863 29-second audio clips with a sample rate of 16 kHz, stored in the mp3 format, taken from songs by 230 artists, and annotated with 188 tags. These tags span instrument, genre, rhythm, gender, and perceptual tags like 'loud', and emotion-based tags like 'happy'. It is a ready-to-use research dataset for MIR tasks such as automatic tagging. We source the MagnaTagATune dataset from City University MIRG website here.

The dataset includes:

- Human-generated annotations collected through Edith Law's *TagATune* game
- Audio clips from magnatune.com, encoded at 16 kHz, 32 kbps, mono MP3
- The source code for the scripts used to generate the dataset
- A detailed analysis of musical structure including rhythm, pitch, and timbre

The data was collected using the TagATune game and music from the Magnatune label. Tags in the dataset are verified [1] (i.e. a tag is associated with an audio clip only if it is generated independently by more than 2 players) and useful for training learning algorithms (i.e. only tags that are associated with more than 50 songs are included). The audio are binned into 16 shards (directories numbered 0-9 and a-f), the sharding key is artist + album name. In Figure 2 from Keunwoo Choi [2], a histogram of the tags can be observed, demonstrating imbalance in the frequency distribution of tags.

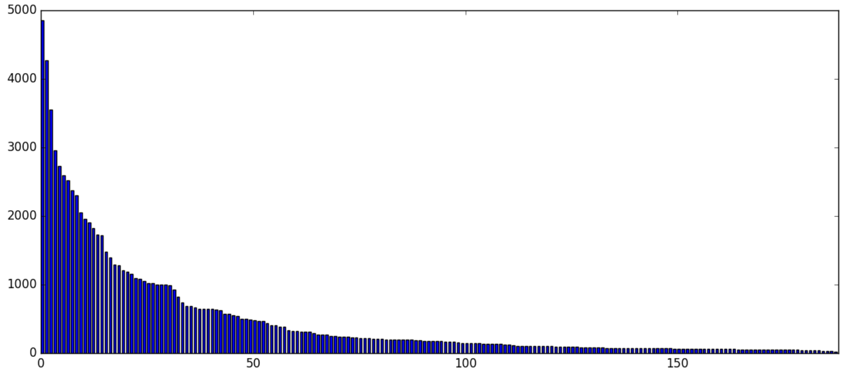

Figure 1: Histogram of tags in the MagnaTagATune dataset.

3 Feature Engineering

To reduce annotation noise and consolidate semantically related tags, we constructed a hierarchical genre grouping comprising 7 parent genres and 43 associated sub-genres. Each parent node aggregates similar labels (like trance, house, electro) into a unified category. This mapping mitigates extreme class imbalance by pooling sparse sub-genre tags into higher-level categories, yielding more stable and learnable multi-label targets for model training. The genres are mapped as such:

Rock: rock, hard rock, soft rock, punk, metal, heavy metal

Classical: classical, clasical, opera, baroque, medieval, operatic

Electronic: electronic, electro, electronica, techno, trance, house, disco, industrial, ambient, new age

Jazz_Blues: jazz, jazzy, blues, funk, funky

Folk_Country: folk, country, celtic, irish

Urban: hip hop, rap, reggae

World: world, indian, india, arabic, middle eastern, eastern, oriental, spanish, tribal

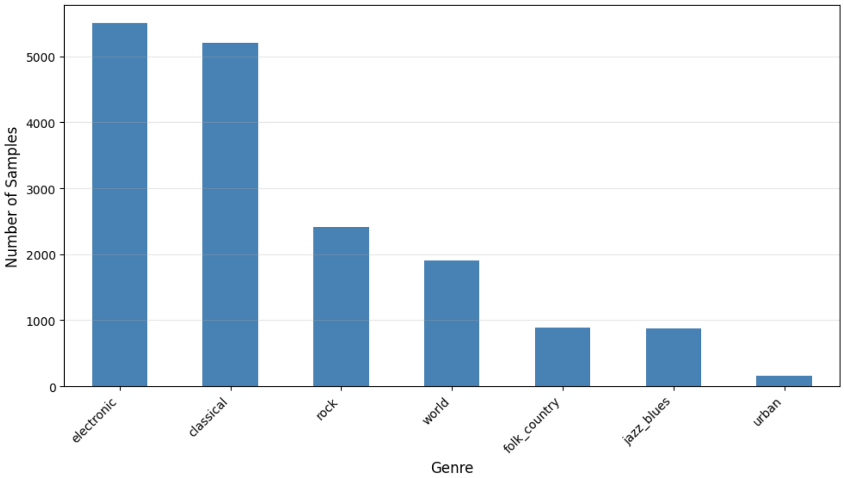

Figure 2: Distribution of genres

For the audio files, we precompute and save normalized log-mel spectrograms into train, validation, and test splits, using a 70% - 15% - 15% division of the dataset. This enables faster training and evaluation by loading preprocessed features directly instead of computing them on-the-fly. We optimize the spectrogram preprocessing parameters to reduce the storage footprint, which leads to an order of magnitude change in the storage footprint. We reduce the number of mel bands, increase the hop length, and limit audio duration to 20 seconds. Additionally, we switch the data type from `float32` to `float16`. These combined modifications reduce the overall storage footprint from 10 GB to 1 GB, enabling more efficient disk usage and faster data loading during model training.

Parameter	Before	After
mel bins	128	80
hop length	512	1024
duration (in seconds)	30	20
dtype	float32	float16
Total size	**10 GB**	**1 GB**

Table 1: Reducing feature resolution leads to reduced storage footprint

4 Model Architecture

4.1 Baseline Multi-layer perceptron

As a baseline, we implement a multi-layer perceptron (MLP) [3] that directly flattens the input spectrogram and passes it through multiple fully connected layers with ReLU activations and dropout [4]. While simpler and lacking the hierarchical feature extraction of CNNs, the MLP provides a useful reference point to evaluate the benefit of convolutional representations for music genre classification.

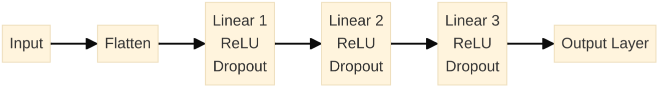

Figure 3: Architecture of proposed multi-layer perceptron

4.2 Convolutional Neural Networks

We adopt a convolutional neural network (CNN) architecture [5], as it has proven effective for audio representation learning due to its ability to capture local spectral-temporal patterns in both spectrograms and raw audio [6]. The proposed CNN model consists of four sequential convolutional blocks, each comprising a 2D convolution layer followed by batch normalization, ReLU activation, and max pooling. This hierarchical design allows the network to progressively extract higher-level features from the input spectrograms. Dropout is incorporated after the dense layers to mitigate overfitting.

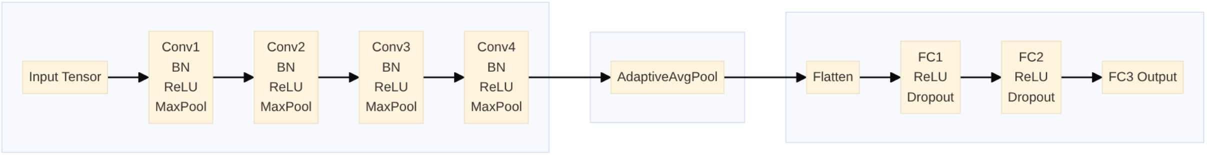

Figure 4: Architecture of proposed convolutional neural network

5 Training Procedure

We train both architectures for multi-label genre classification using the precomputed spectrograms. The audio clips are preprocessed into spectrograms with 80 Mel bands, hop length of

1024 samples, truncated or zero-padded to 20 seconds, and converted to float16. Spectrograms are normalized to zero mean and unit variance to improve numerical stability.

Given the multi-label nature of the task, each log-mel spectrogram x_i is associated with a binary label vector $\mathbf{y}_i \in \{0,1\}^C$, where C denotes the number of genres. The trained models produce logits $\hat{\mathbf{y}}_i \in \mathbb{R}^C$, which are transformed into probabilities using the sigmoid activation:

$$p_{i,c} = \sigma(\hat{y}_{i,c}) = \frac{1}{1 + e^{-\hat{y}_{i,c}}}, \quad c = 1, \ldots, C. \tag{1}$$

The MLP receives flattened spectrograms as input and consists of three hidden layers of sizes 1024, 512, 256 with dropout of 0.4. The CNN operates on the 2D spectrograms with convolutional and pooling layers and dropout of 0.3 to prevent overfitting. Both models are trained using binary cross-entropy loss for multi-label classification and optimized with Adam [7] with a learning rate of 0.001 and weight decay of 1e-4.

$$\mathcal{L}_{\text{BCE}} = -\frac{1}{N} \sum_{i=1}^{N} \sum_{c=1}^{C} \left[y_{i,c} \log p_{i,c} + (1 - y_{i,c}) \log(1 - p_{i,c}) \right] \tag{2}$$

where N is the number of samples, C is the number of classes (genres), $y_{i,c} \in \{0,1\}$ denotes the ground-truth label for sample i and class c, $p_{i,c} = \sigma(\hat{y}_{i,c}) \in [0,1]$ is the predicted probability obtained by applying the sigmoid activation to the logits $\hat{y}_{i,c}$, and $\sigma(\cdot)$ is the sigmoid function.

During each epoch, we perform a forward pass, compute the loss, backpropagate gradients, and update model weights. Predictions are obtained using a sigmoid activation to map outputs to probabilities. Training uses a batch size of 32 with 4 workers for efficient data loading. Models are trained for 30 epochs, with the best-performing model saved based on validation macro-averaged ROC AUC. Final evaluation is performed on a held-out test set, reporting test loss and ROC AUC, allowing comparison between the MLP baseline and the CNN model.

Optional augmentations, such as time stretching, pitch shifting, and additive noise, can be applied to improve model robustness. Due to time constraints, these are left for future work.

6 Evaluation

The trained models are evaluated on a held-out test set comprising 15% of the MagnaTagATune dataset. Each preprocessed log-mel spectrogram is fed into the model to produce logits $\hat{\mathbf{y}}_i$, which is then transformed into probabilities $p_{i,c}$ using the sigmoid activation. Predictions and ground truth labels are collected across the entire test set, and the ROC-AUC for each class are computed and macro-averaged to obtain ROC-AUC$_{\text{macro}}$.

6.1 Evaluation Metric

We employ the macro-averaged area under the receiver operating characteristic curve (ROC-AUC$_{\text{macro}}$) as our evaluation metric, which provides a threshold-independent measure of classifier performance in multi-label settings [8][9]. ROC curve plots the true positive rate (TPR) against the false positive rate (FPR) for varying decision thresholds $\tau \in [0,1]$. For a single class c, these quantities are defined as:

$$\mathrm{TPR}_c(\tau) = \frac{\mathrm{TP}_c(\tau)}{\mathrm{TP}_c(\tau) + \mathrm{FN}_c(\tau)}, \tag{3}$$

$$\mathrm{FPR}_c(\tau) = \frac{\mathrm{FP}_c(\tau)}{\mathrm{FP}_c(\tau) + \mathrm{TN}_c(\tau)}, \tag{4}$$

where TP_c, FP_c, TN_c, and FN_c denote true positives, false positives, true negatives, and false negatives for class c at threshold τ. The area under the ROC curve (AUC) is then computed as:

$$\mathrm{AUC}_c = \int_0^1 \mathrm{TPR}_c(\mathrm{FPR}_c^{-1}(t))\,dt. \tag{5}$$

In multi-label classification, we report the macro-averaged ROC-AUC, which treats each class equally, mitigating the impact of label imbalance.

$$\text{ROC-AUC}_{\text{macro}} = \frac{1}{C}\sum_{c=1}^{C} \mathrm{AUC}_c, \tag{6}$$

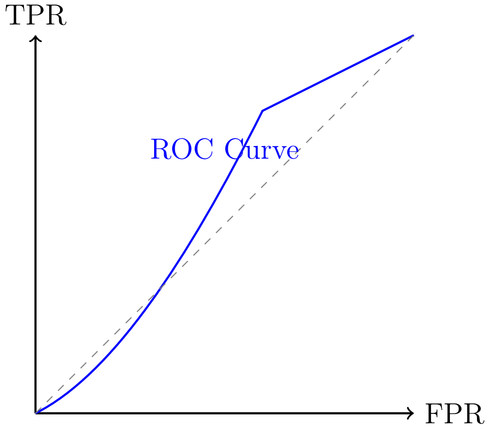

Figure 5: Schematic ROC curve illustrating the trade-off between true positive rate (TPR) and false positive rate (FPR). The area under this curve corresponds to the classifier's discriminative power.

6.2 Test Set Evaluation

The trained MLP and CNN models were evaluated on the held-out test set, comprising 15% of the MagnaTagATune dataset. The overall test performance is summarized in Table 2.

Model	Test Loss	Test ROC AUC
MLP Baseline	0.3074	0.7866
CNN	0.1600	0.9469

Table 2: The CNN significantly outperforms the MLP baseline

The CNN model demonstrates a relative improvement of **20.38%** in macro-averaged ROC-AUC over the MLP baseline, highlighting the effectiveness of convolutional architectures in capturing local spectro-temporal patterns in log-mel spectrograms.

6.3 Per-Genre Performance

Per-genre metrics were computed to evaluate the models' performance across different musical genres. Figure 6 shows F1 score and ROC-AUC by genre for both models. The CNN consistently achieves higher F1 scores and ROC-AUC across most genres, with particularly notable improvements in electronic, folk_country, and world music. The MLP baseline struggles with underrepresented genres, often predicting zero leading to F1 scores of 0.

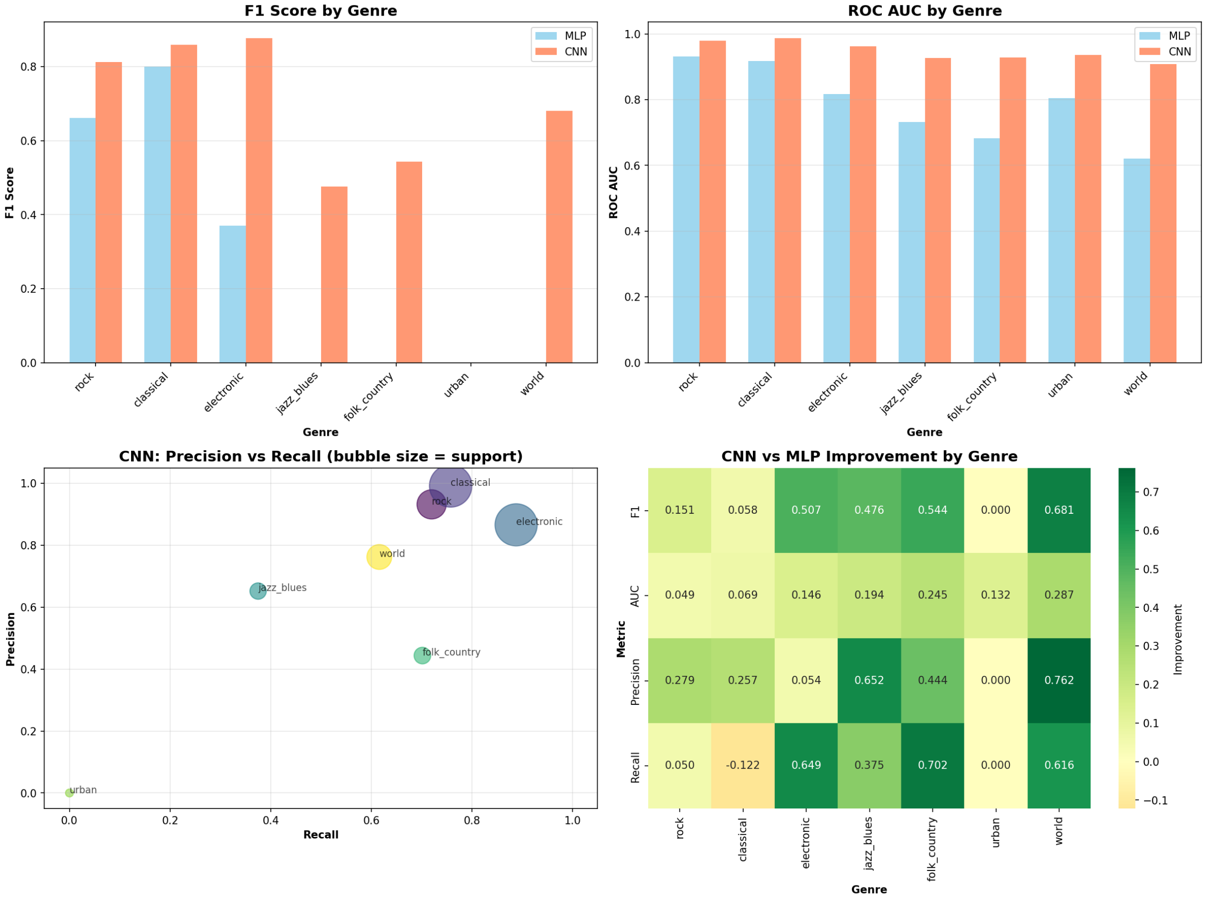

Figure 6: Per-genre evaluation of MLP and CNN models

Tables 3 and 4 provide a detailed breakdown of per-genre metrics, including precision, recall, F1 score, and ROC-AUC.

6.4 Genre Co-Occurrence Analysis

The predicted co-occurrences largely match the true distributions, indicating that the model captures genre correlations effectively, although rare genres like *urban* remain challenging.

Genre	Support	Accuracy	Precision	Recall	F1	AUC
Classical	805	0.8468	0.7352	0.8795	0.8009	0.9179
Rock	382	0.8860	0.6531	0.6702	0.6615	0.9307
Electronic	794	0.7180	0.8120	0.2393	0.3696	0.8167
Jazz_Blues	120	0.9478	0.0	0.0	0.0	0.7322
Folk_Country	124	0.9460	0.0	0.0	0.0	0.6829
Urban	28	0.9878	0.0	0.0	0.0	0.8048
World	276	0.8799	0.0	0.0	0.0	0.6210
Average	-	0.8875	0.3143	0.2556	0.2617	0.7866

Table 3: Per-genre performance metrics for the MLP baseline. Support indicates the number of samples in the test set for each genre.

Genre	Support	Accuracy	Precision	Recall	F1	AUC
Electronic	794	0.9138	0.8661	0.8879	0.8769	0.9629
Classical	805	0.9130	0.9919	0.7578	0.8592	0.9867
Rock	382	0.9447	0.9322	0.7199	0.8124	0.9796
World	276	0.9308	0.7623	0.6159	0.6814	0.9082
Folk_Country	124	0.9365	0.4439	0.7016	0.5438	0.9275
Jazz_Blues	120	0.9569	0.6522	0.3750	0.4762	0.9266
Urban	28	0.9878	0.0	0.0	0.0	0.9367
Average	-	0.9405	0.6641	0.5797	0.6071	0.9469

Table 4: Per-genre performance metrics for the CNN model. The CNN substantially improves performance on most genres compared to the MLP baseline.

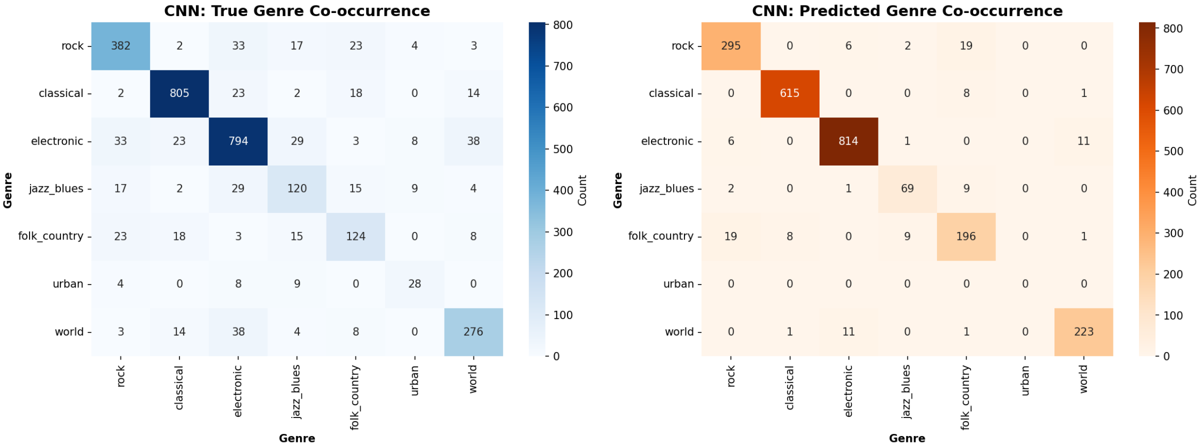

Figure 7: Genre co-occurence analysis. Darker colors indicate higher co-occurrence counts.

7 Conclusion

This work evaluated multilabel genre classification on MagnaTagATune using MLP and CNN architectures. The CNN achieved substantially superior performance, yielding a test ROC-AUC of 0.95 versus 0.79 for the MLP, corresponding to a 20.38% relative improvement. Per-genre analysis confirmed consistent CNN gains across most categories: large F1 and AUC improvements were observed for electronic (F1: 0.89 vs. 0.37), folk_country (0.54 vs. 0), world (0.68 vs. 0), and jazz_blues (0.48 vs. 0). These improvements reflect the CNN's capacity to model local spectro-temporal patterns in log-mel representations, which the MLP fails to capture. Both models struggled with severely underrepresented genres, particularly urban, which exhibited nonzero AUC but F1 scores of 0 across architectures, indicating insufficient positive predictions due to extreme label imbalance.

Genre co-occurrence analysis further demonstrated that the CNN internalizes label dependencies: its predicted co-occurrence matrix closely matches the empirical distribution, with discrepancies emerging mainly for rare genres such as urban. Overall, the results establish convolutional architectures as substantially more effective for multi-label music tagging in low-level time–frequency domains than multi-layer perceptrons.

References

[1] E. L. Law et al. "TagATune: A Game for Music and Sound Annotation." In: *ISMIR*. Vol. 3. Vienna, Austria. 2007, p. 2.

[2] K. Choi. *magnatagatune-list: List of automatic music-tagging research using the MagnaTagATune dataset*. https://github.com/keunwoochoi/magnatagatune-list. Accessed: 2025-12-08.

[3] F. Rosenblatt. "The perceptron: A probabilistic model for information storage and organization in the brain". In: *Psychological Review* 65.6 (1958), pp. 386–408. URL: https://doi.org/10.1037/h0042519.

[4] G. E. Hinton et al. "Improving neural networks by preventing co-adaptation of feature detectors". In: *CoRR* abs/1207.0580 (2012). arXiv: 1207.0580. URL: http://arxiv.org/abs/1207.0580.

[5] Y. Lecun et al. "Handwritten Digit Recognition with a Back-Propagation Network". In: *Advances in Neural Information Processing Systems 2*. Ed. by D. S. Touretzky. Morgan Kaufmann, 1990, pp. 396–404.

[6] S. Dieleman and B. Schrauwen. "End-to-end learning for music audio". In: *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2014, pp. 6964–6968.

[7] D. P. Kingma and J. Ba. "Adam: A Method for Stochastic Optimization". In: *arXiv preprint arXiv:1412.6980* (2015). arXiv:1412.6980. URL: https://arxiv.org/abs/1412.6980.

[8] T. Fawcett. "An introduction to ROC analysis". In: *Pattern Recognition Letters* 27.8 (2006). ROC Analysis in Pattern Recognition, pp. 861–874. ISSN: 0167-8655. URL: https://www.sciencedirect.com/science/article/pii/S016786550500303X.

[9] D. Hand and R. Till. "A Simple Generalisation of the Area Under the ROC Curve for Multiple Class Classification Problems". In: *Machine Learning* 45.2 (2001), pp. 171–186. URL: https://doi.org/10.1023/A:1010920819831.