Predicting Genre from Album Artwork - GenreVision: CS 7643

Kaushik Naresh
knaresh6@gatech.edu

Pranavi Nambi
pnambi3@gatech.edu

Praneeth Gaggenapalli
pgaggenapalli3@gatech.edu

Abstract

Album covers often reflect the style and genre of music, making them valuable visual cues for genre classification. In this work, we address the challenge of predicting music genres from album artwork, a task complicated by the abstract and inconsistent relationship between visual features and musical styles. We apply image classification techniques using deep learning models, including a CNN trained from scratch, and pretrained versions of ResNet152, EfficientNet, and Vision Transformer (ViT). Our approach leverages transfer learning from ImageNet to identify genre-related patterns within album covers, leading to improved model performance even with limited labeled data. Experimental results show that transfer learning substantially outperforms models trained from scratch, although certain genres, such as blues, electronic, and rock, remain more difficult to classify. This work not only establishes strong baselines for album cover genre classification but also has potential applications in enhancing music recommendation systems and organizing large-scale music libraries more efficiently.

1. Introduction/Background/Motivation

We tried multiple methods to predict the genre of music using only album cover images. Instead of using audio clips as traditional methods, we wanted to investigate whether album artwork could help identify genre. The main objective was to explore how useful album covers are for this task.

Music genre classification is performed mainly using audio features such as tempo, pitch, and other spectral features. Deep learning models are often trained along with audio spectrograms and have achieved great results. When it comes to predicting genre just using visual features, traditionally, attempts were made using handcrafted visual features combined with traditional machine learning methods like KNN/SVM. These approaches showed some promising results but struggled to generalize well, especially on larger and diverse datasets. Even today, this problem has been less explored compared to audio-based methods. This is a harder problem because the connection between visual design and music style is indirect.

Classifying album cover genres can help organize music libraries, improve music recommendation systems and helps as an alternative when audio data is missing. It can also help that visual design carries meaningful information about your musical style.

We used the "20k album covers within 20 genres" dataset from Hugging Face. It contains around 20,000 album cover images with 20 different music genre labels. Each sample includes a JPEG file and respective genre label. This data set is balanced across genres, which made it well-suited for our classification task without requiring us to use additional rebalancing techniques.

The code for this project can be found on GitHub.

2. Approach

The data we considered is a balanced dataset where we have 1000 image samples for each of the 20 genres, allowing us to focus more on model training to build a better learner without needing to deal aspects of class imbalance and generalizing which is common in most datasets. Due to the lack of established baseline performance metrics for this dataset, our primary goal was to explore and benchmark multiple models for the genre classification of album covers. We trained three distinct model architectures to evaluate different strategies using both pretrained and untrained versions of the following model architectures: (i) ResNet (ii) EfficientNet (iii) Vision Transformer (ViT)

Recognizing that the relationship between album artwork and music genre is not straightforward, and that the dataset contained a relatively small number of samples per class, we hypothesized that transfer learning would significantly benefit model performance. Transfer learning, leveraging representations learned from large-scale datasets, was expected to help overcome limitations posed by the dataset size and the abstract nature of the classification task.

Our approach builds upon earlier work where VGG models pretrained on CIFAR were used for similar genre classification tasks [4]. We extended this by incorporating more advanced architectures—ResNet, EfficientNet, and Vision Transformer—which offer better performance and parameter efficiency compared to VGG networks. In doing so, we

aimed not only to establish strong baselines for this task but also to systematically analyze the relative performance improvements achievable through different modern pretrained models.

We initially anticipated that album artwork might not strongly correlate with music genres. For instance, a metal album could feature minimalist artwork, while a jazz album might display surrealistic or abstract designs. As a result, we expected that even highly accurate predictions of artwork style might not reliably translate into correct genre classification. Consequently, we anticipated that overall model performance could be limited by the intrinsic ambiguity between visual features and genre labels.

During experimentation, we encountered consistent challenges with certain classes: genres such as blues, electronic, and rock exhibited lower classification performance compared to others. Notably, this trend persisted across all models evaluated, suggesting a broader difficulty in visually distinguishing these genres based solely on album cover art. While the initial training process completed without major technical issues, these observations confirmed our concern that the relationship between artwork and genre would present fundamental challenges for this task.

3. Experiments and Results

3.1. Experiments

We experimented with EfficientNet (B0, B1, B2), ResNet32 (without pretraining by [3]) and Resnet152 (pretrained on Imagenet), and ViT-B16 models. We chose to apply transfer learning using ImageNet-pretrained weights, based on the assumption that models trained on large and diverse datasets like ImageNet can transfer well to album cover classification, especially with the limited training data. These pretrained models already capture general visual information such as edges, textures and shapes, which allows them to start from a strong baseline than learning everything from scratch. This approach leads to faster convergence and better performance. As expected, the pretrained models consistently outperformed their non-pretrained counterparts across all evaluation metrics, confirming the effectiveness of this strategy. The detailed results of these comparisons and assumptions are presented below.

As our models consisted both pretrained and non-pretrained variants, the learned parameters in each model were primarily contained within the convolutional (or self-attention) layers responsible for hierarchical feature extraction. For pretrained models, the base architectures were initialized with ImageNet weights and fine-tuned by replacing and training the final classification layer. Non-learned components included the final post-processing step, where raw model outputs (logits) were converted to class predictions using the argmax function applied over softmax outputs during evaluation. This step was purely deterministic and had no trainable parameters.

The models expected input images in the shape of three-channel RGB tensors, resized to 224×224 pixels to match the input dimensions of ImageNet-pretrained networks. Although our raw album cover images were initially sized at 300×300 pixels, we resized them during pre-processing to optimize memory usage via the data loader pipeline; however, we did not apply further normalization or pixel-wise standardization (e.g., ImageNet mean / std normalization), which could have improved model convergence. The labels were encoded as integer class indices for each genre and passed to the loss function in this format.

3.1.1 ResNet

We evaluated the performance of two convolutional neural networks based on Resnet architecture [2]: ResNet152 pre-trained on ImageNet and ResNet32 [3] is trained from scratch for the task of recognizing album genre based on cover artwork. ResNet152 outperformed ResNet32, consistent with expectations given its deeper architecture and prior exposure to large-scale visual features through pre-training. The performance gap highlights the benefit of transfer learning, especially in settings with limited domain-specific data.

We trained the models using CrossEntropyLoss, which is appropriate for multi-class classification tasks. Optimization was performed using the Adam optimizer with a learning rate of 0.0001, batch size of 256, and weight decay of 0.0001. These hyper-parameters were selected empirically based on convergence stability and GPU memory constraints. While a learning rate scheduler was available in our pipeline, we focus our analysis on fixed learning rate behavior across 15-30 training epochs for comparability.

Model	Acc. (%)	Prec. (%)	Recall (%)	F1 (%)
Resnet152(pretrained)	22.5	20.72	22.64	20.99
Resnet32 (no pretrain)	11.54	10.41	11.47	8.7

Table 1. Comparison of Resnet with and without pretraining.

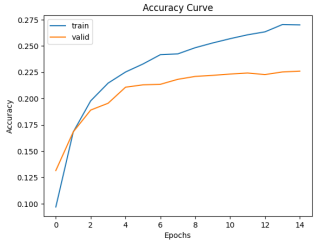

Figure 1. accuracy - resnet152 pretrained

ResNet152, with pretrained weights, achieved a significantly higher **average class accuracy of 22%**, demonstrat-

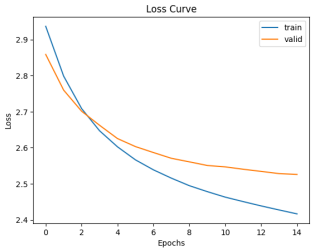

Figure 2. Loss curve - resnet 152 pretrained

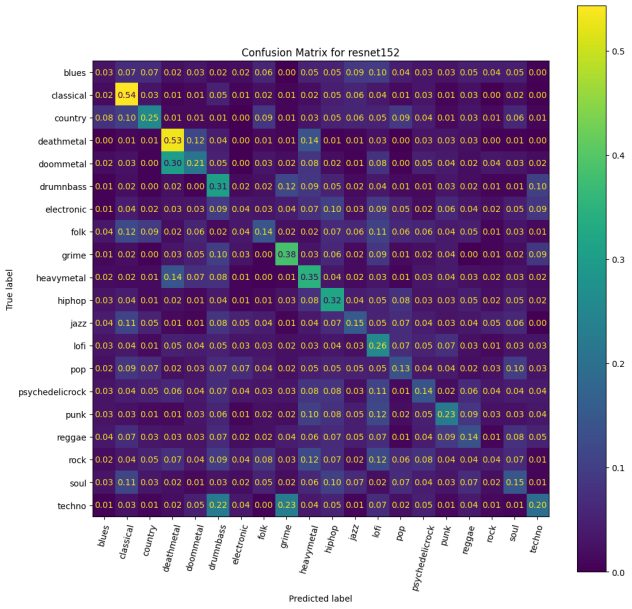

Figure 3. Confusion matrix for Pretrained ResNet152

ing the benefits of transfer learning in low-data, abstract-vision settings. It performed particularly well in genres such as *classical* (54%), *deathmetal* (53%), and *grime* (38%), indicating its ability to capture genre-relevant visual cues in more visually distinctive categories. Confusion matrix 3 highlights the classification matrix for classification of genre labels via pretrained resnet 152.

In contrast, **ResNet32**, trained without pretraining, achieved only **11% average class accuracy**. This model consistently underperformed across genres, likely due to its limited capacity and the lack of prior knowledge learned from large-scale datasets like ImageNet.

The comparison underscores that both **model depth** and **pretraining** are critical to achieving better generalization in this task. However, even the stronger model (ResNet152) struggled with genres such as *blues*, *rock*, and *electronic*, suggesting that album artwork alone may not offer sufficient discriminative information for certain categories .

However, despite the superior performance of ResNet152, overall classification accuracy remained moderate. This suggests that the visual features learned from ImageNet, while broadly effective for general object recognition tasks, may not transfer well to the genre classification of album covers. Unlike the natural images used in ImageNet, album artwork is often abstract, stylized, or symbolic, and its relationship to music genre is neither direct nor consistent. Therefore, even models with strong performance on large-scale image benchmarks (e.g., 78% top-1 accuracy on ImageNet) may struggle in this domain due to the semantic gap between visual design and musical category.

3.1.2 EfficientNet

EfficientNet, introduced by Tan and Le in 2019 [5], was a breakthrough in image classification and has since become widely adopted for supervised vision tasks due to its strong performance and computational efficiency. It achieves high accuracy with fewer parameters and lower resource usage through a compound scaling method that uniformly scales the depth, width, and input resolution.

We selected EfficientNet for our project due to this efficiency and effectiveness. To explore how model complexity affects performance, we experimented with three variants: B0, B1, and B2. B0 is the baseline model that is lightweight and fast, making it suitable for initial experimentation. B1 and B2 gradually increase in model size and capacity, allowing them to capture more complex patterns. This helped us analyze the trade-offs between model size, training time, and classification accuracy for our dataset. This experiment was also conducted using both pre-trained and non pre-trained versions to evaluate the effect of transfer learning.

During training, we used consistent hyper-parameters across EfficientNet variants to ensure a fair comparison: input size of 224, batch size of 32, 30 epochs, learning rate of 0.0005 with Adam optimizer and weight decay of 0.0001. The original classification head was replaced with a dropout layer followed by a fully connected layer to produce predictions for 20 classes. While the architecture allowed for freezing the base feature extractor using freeze_base=True, in our experiments we trained the entire model end-to-end by setting freeze_base=False.

Model	Params (M)	Acc. (%)	Prec. (%)	Recall (%)	F1 (%)
EfficientNet-B0	5.3	32.20	31.86	32.20	31.78
EfficientNet-B1	7.8	33.16	33.24	33.16	33.03
EfficientNet-B2	9.1	**33.33**	32.94	**33.33**	**32.99**

Table 2. Performance and parameter comparison of pretrained EfficientNet variants.

Model	Acc. (%)	Prec. (%)	Recall (%)	F1 (%)
EfficientNet-B2 (pretrained)	33.33	32.94	33.33	32.99
EfficientNet-B2 (no pretrain)	12.61	13.43	12.61	12.63

Table 3. Comparison of EfficientNet-B2 with and without pretraining.

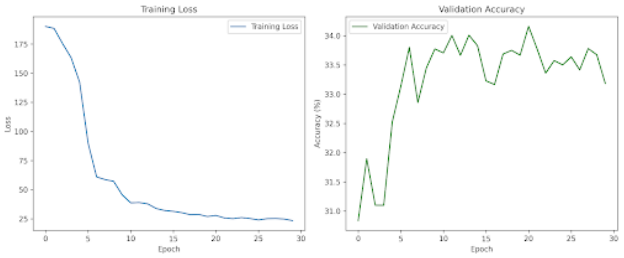

Figure 4. Training loss curve and validation accuracy curve for EfficientNet-B2

Qualitatively, we observed how model scaling and transfer learning affected our performance. Among EfficientNet variants, the B2 model with pre-training achieved the best overall performance as seen in the Table 3 with **33.33% accuracy.** This indicates that larger models with more capacity will be better to learn complex patterns from album cover art. Comparing B0,B1 and B2, we observed a consistent improvement in all metrics as model size increases. Additionally, as seen in table 2 when comparing pre-trained and non pre-trained versions of EfficientNet-B2 the model significantly outperformed the non-pretrained one, reinforcing the importance of transfer learning with limited data,

The overall accuracy remained around 30-33% which is relatively low and it stems from the limitations of using album cover images alone for genre classification. It was challenging because lot of genres like metal, electronic and rock tend to have overlapping visual styles leading to frequent misclassifications. These challenges concluded that even though EfficientNet is able to extract meaningful visual features, the visual cues alone may not be sufficient for achieving high classification performance on this task.

In contrast, genres like Metal(Classes - 3,4) or Electronic (Class 6) showed lower per-class accuracy, may be because of its high visual overlap with genres like Rock or Punk. These genres often use dark, abstract visuals that look similar across multiple classes. These variations highlights that some genres might benefit from identity, other classes or genres suffer from ambiguity and similarities, which affected overall classification accuracy.

This trend is further reflected in the confusion matrix, where we observe strong diagonal values for visually distinctive genres such as classical and country, but more scattered predictions for genres like metal and electronic. This tells us confusion between visually similar genres and reinforces the limitations of relying completely on album artwork.

3.1.3 ViT_B_16

For the vision transformer, we used PyTorch's built-in ViT_B_16 model architecture, which is in turn adapted from the "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale" paper. The choice of this model was made due to the drastic difference in the number of parameters between the various PyTorch ViT models, which range from 86M to 632M[1]. By choosing the "base" model we aimed to avoid using a model too large given our dataset of 20k items. We added a classifier section to the end this model to be able to predict probabilities for our 20 classes, consisting of a dropout layer followed by a pair of linear layers with a ReLU layer between them. This choice was made after testing with different options, including just a linear layer, dropout+single linear layer, and varying the hidden size between these two layers before finally settling on 128 for the hidden size producing the best results. Both models used the cross-entropy loss function.

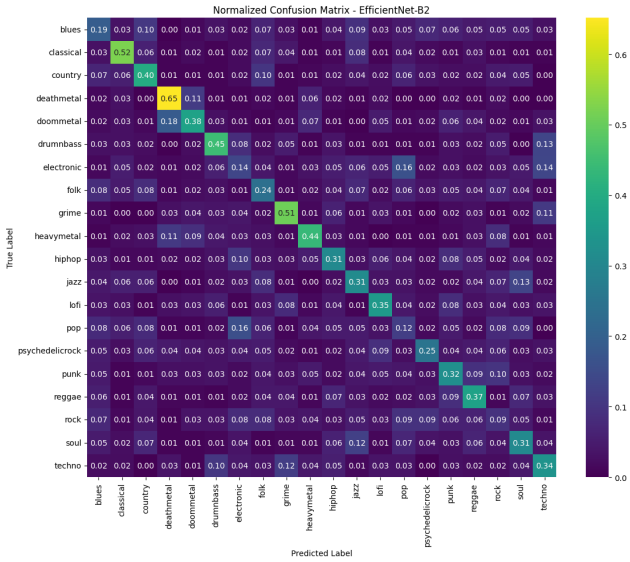

Figure 5. Confusion matrix for EfficientNet-B2

First, we examined directly training this model on our dataset, without any pre-training. This experiment was to ascertain how well the architecture could model our data from scratch. As album artwork is a more nuanced form of measuring genre, we hoped to at least see results that would be better than random chance (1 out of 20 – 5%). This model resulted in a max validation accuracy of 9.62% across 30 epochs, almost 2x better than random chance, signifying that perhaps the ViT architecture could indeed help us in our problem. The accuracy on our test dataset was 10.35%.

To further build upon this, the same architecture was used after being pre-trained on the ImageNet 1k class dataset (using weights provided by PyTorch) with the expectation of better performance as a result of the pre-training. These weights were chosen instead of the other options available from PyTorch as the model was trained from scratch on the ImageNet 1k dataset to produce these weights, thus that dataset played a role of being the direct

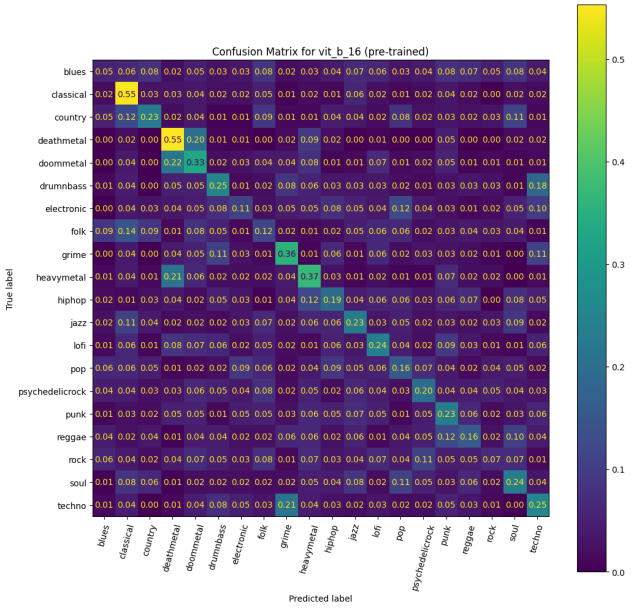

Figure 6. Confusion matrix for Pre-Trained ViT_B_16

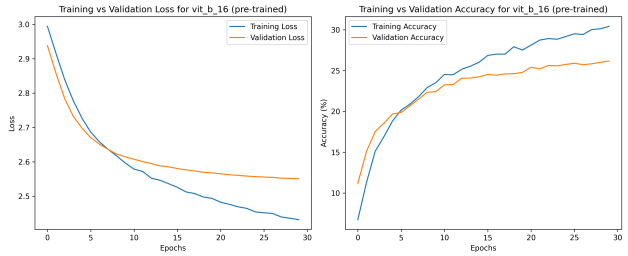

Figure 7. Loss and Accuracy Plots for Pre-Trained ViT_B_16

dataset that our model would be pre-trained on. Additionally, all model parameters except for the final layer of the ViT_B_16 architecture were frozen, to allow learning on our dataset to only occur on the additional layers we added to classify the 20 classes, as well as the final linear layer of the ViT_B_16 architecture. As expected, this model far outperformed the model trained from scratch, resulting in a max validation accuracy of 26.16%, and a test accuracy of 24.38%, almost 5x better than random chance.

Model	Acc. (%)	Prec. (%)	Recall (%)	F1 (%)
ViT_B_16 (pretrained)	24.38	22.88	24.38	23.14
ViT_B_16 (no pretraining)	10.35	8.71	10.35	8.24

Table 4. Comparison of ViT_B_16 with and without pretraining.

Both models used an input size of 224, 30 epochs of training, and the Adam optimizer with a learning rate of 0.0001 and a weight decay of 0.0005. In both cases to assist with learning and reduce overfitting, the learning rate was reduced by 25% and the weight decay was increased by 25% every 10 epochs. These hyper-parameters were chosen after training the models with several different sets of values. The only hyper-parameter changed between the models was the batch size – the model trained from scratch used a batch size of 32, while the pre-trained model used a batch size of 512. The batch size for the pre-trained model was increased to prevent overfitting, which was noticeable with lower batch size values – this can be attributed to the fact that the pre-trained model was only updating weights for the linear classifier section we added, and the final linear layer of the ViT-B-16 architecture, which included a hidden size of 128, while the model trained from scratch had the flexibility to update weights across the entire model. The hidden size of 128 might have played a role in overfitting the pre-trained model to the training data as it was larger than the batch size of 32 – and it was difficult to completely mitigate overfitting in both models, but these choices were able to reduce its impact.

Although neither model produced exceptional predictions, they were both able to predict genres better than random chance, and given the subjectivity of music genres and especially how album artwork may represent them, this presents itself as a positive outcome. When looking at the confusion matrix, we see that a lot of the genres are closely related to each other, for example there are three kinds of Metal, two kinds of Rock, and Punk, which are all related to each other and have similar themes across their album artwork. Thus, while the model has a high accuracy on the ImageNet dataset it was initially trained on (95.318%), the complexity of music genres makes it harder to distinguish in this problem, but nonetheless, the model performs better than random chance, and is able to make connections and generalizations that would seem natural given genre overlapping and similarities.

3.2. Evaluation Metrics

We measured performance using multiple multi-class classification metrics: Accuracy, precision, recall, f1-score and per-class accuracy.

3.2.1 Accuracy

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

Accuracy is the ratio of correct predictions to the total number of predictions. While it can be misleading for imbalanced datasets, our dataset is perfectly balanced, making accuracy a reliable metric for evaluation.

3.2.2 Precision

$$\text{Precision} = \frac{TP}{TP + FP}$$

Precision measures how many predicted labels for a class are actually correct. High precision ensures that model is

not over-classifying a certain genre just because it looks visually similar.

3.2.3 Recall

$$\text{Recall} = \frac{TP}{TP + FN}$$

Recall helps in measuring the ability of a model to correctly identify all the relevant instances of a class. It is the ratio of true positives to the total actual positives, indicating how well a model captures all examples of a genre.

3.2.4 F1-score

$$\text{F1 Score} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

F1-score helps balance precision and recall, especially for unbalanced datasets. For this classification task, even though precision is more important because misclassifying one genre as another can be misleading, F1-score gives us a better overall picture of how our models are performing in identifying the correct genre without missing few classes.

3.2.5 Per-class accuracy

Along with these metrics, per-class accuracy was also examined which helped us in understanding how well our model is performing on each individual genre rather than just overall performance. It highlights if the model is biased or is struggling to classify specific classes, even when the overall accuracy is high.

	resenet		**VIT**		**Efficientnet**		**Average**
pretrained	yes	no	yes	no	yes	no	
blues	3%	1%	5%	4%	19%	3%	6%
classical	54%	36%	55%	13%	52%	26%	42%
country	25%	14%	23%	12%	40%	12%	23%
deathmetal	53%	50%	55%	37%	65%	23%	52%
doommetal	21%	22%	33%	28%	38%	28%	28%
drumnbass	31%	2%	25%	2%	45%	10%	21%
electronic	4%	5%	11%	1%	14%	2%	7%
folk	14%	11%	12%	11%	24%	17%	14%
grime	38%	22%	36%	15%	51%	22%	32%
heavymetal	35%	4%	37%	19%	44%	14%	28%
hiphop	32%	0%	19%	0%	31%	6%	16%
jazz	15%	24%	23%	0%	31%	7%	19%
lofi	26%	3%	24%	26%	35%	15%	23%
pop	13%	1%	16%	1%	12%	7%	8%
psychedelicrock	14%	3%	20%	5%	25%	8%	14%
punk	23%	13%	23%	7%	32%	8%	20%
reggae	14%	8%	16%	7%	37%	15%	16%
rock	4%	0%	7%	0%	9%	3%	4%
soul	15%	1%	24%	5%	31%	10%	15%
techno	20%	11%	25%	14%	34%	14%	21%
Average	23%	11%	24%	10%	33%	13%	20%

Table 5. Genre classification accuracy across models (pretraining : yes vs without pretraining: no)

The general trend as shown in the summary table of the results 5 reveals that while deeper models trained are more effective at identifying genre-specific patterns in album cover art, the ambiguity and stylistic overlap between genres impose a natural performance ceiling. The results reinforce the value of transfer learning in domains with limited labeled data, but also highlight the limitations of relying purely on visual cues for genre classification, especially in cases where genre semantics are not strongly encoded in cover design.

3.3. Model Performance Comparison

All three pre-trained models-EfficientNet-B2, ResNet-152 and Vit-B16 have benefited from transfer learning but their performance varied based on their architecture design. EfficientNet-B2 outperformed others across all metrics as seen in As shown in Table 6 below. It is likely due to its scaling strategy that balances depth, width and resolution enabling more efficient feature extraction than others. ResNet-152, while deep might have overfit or under utilize features for this limited dataset due to its rigid hierarchical architecture. ViT-B16, even though it has lot of potential, requires much larger datasets to generalize well and struggled to learn from the limited visual patterns in album art. These results gave us a conclusion that EfficientNet's scalable design is better suited for mid-size datasets as it has balanced computational efficiency and model capacity to capture meaningful visual features.

All models struggled to achieve higher accuracy primarily due to the following limitations:

1. Visual overlap between Genres: Genres like rock, metal and electronic often share similar dark, abstract artwork leading to confusion while classification.

2. Lack of explicit genre signals and multi-modal input: Unlike audio information, album covers don't always contain consistent visual cues tied to genre. Combining it with audio or metadata could provide more context and improve performance.

3. High Intra-class variation: Even within genre, cover styles varied widely, making learning of the consistent features harder.

4. Dataset size: Even though it is a balanced dataset, it is not large enough for the deep networks to fully generalize from noisy visual patterns.

Model	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**F1 Score (%)**
EfficientNet-B2	**33.33**	**32.94**	**33.33**	**32.99**
ResNet-152	22.51	20.72	22.64	20.99
ViT-B16	24.38	22.88	24.38	23.14

Table 6. Performance comparison of pretrained models across accuracy, precision, recall, and F1-score.

4. Work Division

Student Name	Contributed Aspects	Details
Kaushik Naresh	Data Creation, ViT architecture review, Implementation and Analysis	Created the dataloader module to process the dataset from hugging face into tensors for the images and labels, and further shuffled and split the data into training and testing sets (dataloader.py). Created the classifier layer for the vision transformer (vit_b_16.py) and trained the hyperparameters (config_vit.yaml and train_vit.ipynb) for the vision transformer models after conducting research on appropriate ViT models to be used and additional classifier layers needed. Added logic to the training loop to dynamically alter learning rate and weight decay for the vision transformer (train_vit.ipynb). Added code to produce the confusion matrix plots (train_vit.ipynb). Conducted experiments and analysis on the results for the ViT_B_16 models (see 3.1.3).
Praneeth Gaggenapalli	Model architecture review. Experiment layout. Process optimization Implementation and analysis	Conducted literature review of model architectures for baseline as well as methodologies best suitable for Genre classification from Album art usecase and layed out experiments as summarized in 3.1. Trained the Resnet Models on both pretrained(resnet152) and base (resnet32) versions (see 3.1.1) Code and outputs for pretrained model are stored in codebase for (resnet152.ipynb) and training process and updates for resnet32.ipynb using config Added layers to save and extract model metrics and loss curves in training pipeline (see code in above .ipynb links). Identified gaps in training and data pipeline and updated pipeline to include learning scheduler to test performance gain but proved to be ineffective (see code in Trainer class under (resnet152.ipynb)). Performed hyperparameter tuning on resnet models to resolve for overfitting and generalization issues. Results of generalized model are made avaibale as part of 3.1.1 and summzrized results from all models along with resnet in table 5
Pranavi Nambi	Model architecture review Training pipeline and hyper-paramter tuning Implementation and analysis	Did literature review on CNN architectures and evaluation metrics(see 3.2 and identified the best suitable architecture to genre classification using visual features as described in section 3.1. Implemented and modified pre-existing layers for B0,B1 and B2 for both pre-trained and base model. This is the architecture file and added layers (efficientnet.py) Modified dataloader to incorporate validation sets. Handled training pipeline(trainefficientnet.ipynb) setup for multiple models, tuned hyper-parameters(config.yaml) and generated and analyzed training behavior using curves(see Fig 4), confusion matrix(see Fig 5) accuracy trends and identified corresponding issues. Section 3.1.2

Table 7. Contributions of team members.

References

[1] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *CoRR*, abs/2010.11929, 2020. 4

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015. 2

[3] Yerlan Idelbayev. Proper ResNet implementation for CIFAR10/CIFAR100 in PyTorch. https://github.com/akamaster/pytorch_resnet_cifar10. Accessed: 2025-04-10. 2

[4] Jonathan Quiang Li, Di Sun, and Tongxin Cai. Genre classification via album cover. 2020. 1

[5] Mingxing Tan and Quoc V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, pages 6105–6114. PMLR, 2019. 3