information

MDPI

Article

Comparative Analysis of Deep Learning Architectures and Vision Transformers for Musical Key Estimation

Manav Garg [1], Pranshav Gajjar [1], Pooja Shah [2], Madhu Shukla [3], Biswaranjan Acharya [3,*], Vassilis C. Gerogiannis [4,*] and Andreas Kanavos [5,*]

[1] Department of Computer Science and Engineering, Institute of Technology, Nirma University, Ahmedabad 382481, Gujarat, India; 19bce062@nirmauni.ac.in (M.G.); 19bce060@nirmauni.ac.in (P.G.)
[2] School of Technology, Pandit Deendayal Energy University, Gandhinagar 382426, Gujarat, India; pooja.shah@sot.pdpu.ac.in
[3] Department of Computer Engineering—AI and BDA, Marwadi University, Rajkot 360003, Gujarat, India; madhu.shukla@marwadieducation.edu.in
[4] Department of Digital Systems, University of Thessaly, 41500 Larissa, Greece
[5] Department of Informatics, Ionian University, 49100 Corfu, Greece
* Correspondence: biswaranjan.acharya@marwadieducation.edu.in (B.A.); vgerogian@uth.gr (V.C.G.); akanavos@ionio.gr (A.K.)

Abstract: The musical key serves as a crucial element in a piece, offering vital insights into the tonal center, harmonic structure, and chord progressions while enabling tasks such as transposition and arrangement. Moreover, accurate key estimation finds practical applications in music recommendation systems and automatic music transcription, making it relevant across academic and industrial domains. This paper presents a comprehensive comparison between standard deep learning architectures and emerging vision transformers, leveraging their success in various domains. We evaluate their performance on a specific subset of the GTZAN dataset, analyzing six different deep learning models. Our results demonstrate that DenseNet, a conventional deep learning architecture, achieves remarkable accuracy of 91.64%, outperforming vision transformers. However, we delve deeper into the analysis to shed light on the temporal characteristics of each deep learning model. Notably, the vision transformer and SWIN transformer exhibit a slight decrease in overall performance (1.82% and 2.29%, respectively), yet they demonstrate superior performance in temporal metrics compared to the DenseNet architecture. The significance of our findings lies in their contribution to the field of musical key estimation, where accurate and efficient algorithms play a pivotal role. By examining the strengths and weaknesses of deep learning architectures and vision transformers, we can gain valuable insights for practical implementations, particularly in music recommendation systems and automatic music transcription. Our research provides a foundation for future advancements and encourages further exploration in this area.

Keywords: music information retrieval (MIR); musical key estimation; deep learning; vision transformers; convolutional neural networks (CNNs)

check for updates

Citation: Garg, M.; Gajjar, P.; Shah, P.; Shukla, M.; Acharya, B.; Gerogiannis, V.C.; Kanavos, A. Comparative Analysis of Deep Learning Architectures and Vision Transformers for Musical Key Estimation. *Information* **2023**, *14*, 527. https://doi.org/10.3390/info14100527

Academic Editors: Krzysztof Ejsmont, Aamer Bilal Asghar, Yong Wang and Rodolfo Haber

Received: 12 July 2023
Revised: 20 September 2023
Accepted: 21 September 2023
Published: 28 September 2023

1. Introduction

Estimation of musical key plays a fundamental role in music information retrieval (MIR), finding wide applications in music analysis, recommendation systems, and automatic music transcription [1,2]. The key of a musical piece serves as the central tonal reference point, shaping harmonic progressions and providing a foundation for the development and resolution of harmonic tension [3]. It represents the precise scale—either major or minor—upon which a Western music composition is built [4]. Determining the key of a musical composition is crucial for understanding its tonality and harmonic framework, as a major or minor key signifies its association with the respective major or minor scale.

The existing academic literature often adopts a similar pipeline for key estimation methods. The initial step involves generating a spectrogram or constant Q transform, which represents the time and frequency components of an audio signal. Subsequently, a pitch-class profile, also referred to as a chroma feature, is extracted from each time frame. This profile provides an octave-independent and timbre-invariant representation of pitch classes. By comparing these accumulated features over time with feature templates for each key, the global key for the musical piece is estimated. Several systems, such as those presented in [5–8], have employed this approach. However, it is worth noting that while this method performs well for many genres, like pop/rock and electronic music, it may not effectively handle key modulations commonly encountered in classical music.

Beyond standalone musical key estimation, another approach focuses on simultaneous estimation of both the key and chords [9–12]. These algorithms aim to enhance the accuracy of estimations by leveraging the intrinsic relationship between keys and chords in music. To handle modulations between keys, these systems often estimate local keys. While dedicated key and chord estimation algorithms currently yield superior results, it is important to note that these simultaneous estimation approaches have the potential to provide a more comprehensive understanding of the harmonic content of musical sounds.

To tackle the task of predicting the global key of musical works, various techniques have been employed, broadly classified as template-based, geometry-based, or probabilistic algorithms. Template-based approaches, as exemplified by the Krumhansl–Schmuckler key-finding algorithm [3,8,13] and the Temperley–Kostka–Payne key-finding algorithm [3], utilize statistical analysis of pitch-class distributions to estimate the key. Geometry-based algorithms, such as Harte's key detection algorithm [14], employ Fourier transform and harmonic templates for key estimation. Probabilistic algorithms like Fujishima's dynamic Bayesian network [15] and MADRIGAL [2] leverage probabilistic models or neural networks, considering features such as pitch-class distributions, chord progressions, and melodic contour for key estimation. These algorithms have been applied to both symbolic and audio-based data, with the accuracy and efficiency of the estimation influenced by the specific characteristics of the music being analyzed.

Tonality analysis plays a vital role in music information retrieval (MIR) by identifying the musical key of a piece. The key represents the central tonal center around which the pitches are organized, and it can undergo shifts known as modulations throughout the composition [16]. In Western tonal music, these tonal centers give rise to distinct regions of local keys, which are different from the global key [17]. The global key, also referred to as the tonality, is defined by the set of pitches that characterize the tonal identity of a musical work or excerpt [18].

The accurate identification of the musical key is of paramount importance, as it highlights significant notes and influences the playability, texture, and range of various instruments. However, research has indicated that individuals with varying levels of musical expertise may perceive tonality differently, requiring a high degree of perceptual skill for manual key identification [13]. To tackle this challenge, automated key estimation and analysis have been developed using diverse approaches, including spectral analysis and machine learning techniques.

Studies have shown that classical music often exhibits a strong adherence to a global key within a single piece, as highlighted in [19]. In contrast, tonality in popular music tends to be more intricate and less clearly defined. Recognizing the importance of automated key estimation and analysis [20], emphasizing their significance in retrieving music information, particularly in the context of popular music. By accurately detecting the key of a musical piece, music information retrieval systems can enhance the precision of music recommendation systems, automatic chord identification, melody extraction, and various other applications.

Advancements in computational intelligence have paved the way for the development of surrogate or predictive systems, which exhibit superior accuracy levels [21–24]. These developments span various domains, yielding heuristic implications and observable utility [25]. Deep learning, a subset of machine learning algorithms, utilizes neural architectures capable of identifying inherent patterns and extracting valuable information from data [26,27]. Convolutional neural networks (CNNs), which are prominent architectures in this domain, have shown promise in the field of music information retrieval (MIR) [28–31]. However, recent advancements in image processing have introduced vision transformers, which incorporate the fundamentals of convolutional architecture and demonstrated remarkable performance across diverse domains [32–35]. The literature also suggests their relevance and utility in MIR, including key estimation [32,34,35]. Building upon our previous work that enhanced CNNs using Siamese networks and augmentative approaches [36], we aim to further enhance the deep encoder strategy by leveraging vision transformers.

The primary objective of this study is to explore the feasibility of estimating musical keys using vision transformers and compare their effectiveness with that of traditional convolutional network architectures commonly employed in music information retrieval tasks. Notably, this study presents novel experiments conducted on the GTZAN dataset utilizing transformer architectures, which contribute to the existing body of knowledge. The obtained results offer valuable insights to researchers by shedding light on the application of transformers in music and their potential benefits for key estimation tasks.

Section 2 provides an overview of relevant literature in the field of musical key estimation, highlighting previous research and established techniques. Section 3 outlines the research methodology employed in this study, including the dataset used for validation and the evaluated deep learning architectures. In Section 4, we present a detailed analysis of the obtained results, including performance metrics and outcomes for each architecture, as well as the time taken for training and testing. Finally, in Section 5, we provide concluding remarks and discuss the significance of vision transformers in musical key estimation, along with potential directions for future research.

2. Related Work

This section explores recent papers and research articles that have made notable contributions to the field of music research, with a specific focus on key estimation. The authors of [37] presented a system for real-time musical accompaniment, wherein a computer-driven orchestra learns from and follows a soloist using a hidden Markov model and a Kalman filter-like model. The system generates output audio by phase vocoding a pre-existing recording. The study showcases examples of the system in action, highlighting its connection to machine learning and suggesting potential new directions for research. In [38], the authors provided a comprehensive review of deep learning techniques for the processing of audio signals, encompassing speech, music, and environmental sound processing. The review explores the commonalities and distinctions among these domains, identifies key references, and explores the potential for cross fertilization. It covers topics such as feature representations, models, and application areas and concludes by addressing key issues and proposing future directions for deep learning in audio signal processing.

In addition to the aforementioned papers, several other studies have made significant contributions to the field of music research. The authors of [39] employed machine learning and word feature analysis to identify texts belonging to the Black Fantastic genre within the HathiTrust Digital Library. This project presents a pilot predictive modeling process that computationally identifies these texts by leveraging curated word feature sets for each data class. Another study [40] explored deep learning approaches for estimation of the difficulty of musical pieces, which is crucial for music learners to choose appropriate pieces based on their skill level. The study proposes a pipeline that converts MIDI files to piano roll representations, trains models using corresponding difficulty labels, and achieves a state-of-the-art F1 score of 76.26% with multiple deep convolutional neural networks. The implications of these results extend to automated difficulty-controlled music generation. Furthermore, the work reported in [41] discusses the utilization of artificial intelligence in music composition to overcome the expenses associated with creating original music for promotional videos. The authors proposed a globally accessible web-based platform leveraging multiple machine learning algorithms to recognize pleasing sound combinations and produce unique music.

In computer vision, the Orthogonal Transformer [42] has emerged as a prominent vision transformer backbone, combining global self-attention and local correlations, demonstrating remarkable performance across various computer vision tasks. Pure transformer models [43] have shown exceptional performance in video classification tasks, leveraging spatiotemporal token extraction techniques and regularization methods. The intersection of quantum computing and music generation was explored by the authors of [44], who delved into the use of partitioned quantum cellular automata for the creation of original music compositions, highlighting the potential impact of emerging quantum computing technology on the music industry. In the domain of music harmonization, the work reported in [45] introduced an innovative methodology for generating new harmonizations of jazz standards by combining melodies and chords from different songs. These studies demonstrate the potential of computational creativity and music education in diverse musical applications.

Table 1 provides a comprehensive summary of existing papers and prominent literature in the field. Notably, current algorithms predominantly rely on conventional convolutional architectures, lacking the properties and benefits offered by vision transformers, such as uniform representation across layers, self-attention, and improved feature representation. While conventional CNNs may be computationally efficient, we conducted a thorough temporal analysis to investigate the effects of vision transformers.

Table 1. Papers related to the proposed domain of research.

Paper	Domain	Description
[46]	Genre Classification	Provides an overview of music genre classification within music information retrieval, discussing techniques, datasets, challenges, and trends in machine learning applied to music annotation, in addition to reporting a music genre classification experiment comparing various machine learning models using Audioset.
[47]	Music Generation	Discusses limitations in using deep learning for music generation and suggests approaches to address these limitations, in addition to highlighting recent systems that show promise in overcoming the limitations.
[48]	Music Generation	Introduces DeepJ, an end-to-end generative model for composing music with tunable properties based on a specific mixture of composer styles, demonstrating a simple technique for controlling the style of generated music that outperforms the biaxial long short-term memory (LSTM) approach.
[49]	Tempo Estimation	Explores the potential of using deep learning to improve the accuracy of global tempo estimation, considering the applications and limitations of evaluation metrics and datasets, including a survey of domain experts to understand current evaluation practices, in addition to providing a public repository with evaluation codes and estimates from different systems for popular datasets.
[50]	Key Estimation	Proposes a machine learning approach for determining the musical key of a song, which is important for various music information retrieval tasks, testing the model with four algorithms and achieving a maximum accuracy of 91.49% using support vector machine (SVM).
[51]	Genre Classification	Proposes novel approaches for music genre classification utilizing machine learning, transfer learning, and deep learning concepts, testing five approaches on three music datasets. The proposed BAG deep learning model combines bidirectional long short-term memory (BiLSTM) with an attention and graphical convolution network (GCN), achieving a classification accuracy of 93.51%.

3. Methodology

In this section, we provide a comprehensive review of the advanced deep learning architectures utilized in our experimentation, as well as the dataset employed in this study. Specifically, we integrate ResNet, Vision Transformer, and SWIM Transformer, which are widely recognized deep learning architectures. The subsequent subsections offer detailed explanations of these architectures.

Figure 1 provides a graphical representation of the complete flow of the research paper, illustrating the different sections and their interconnections.

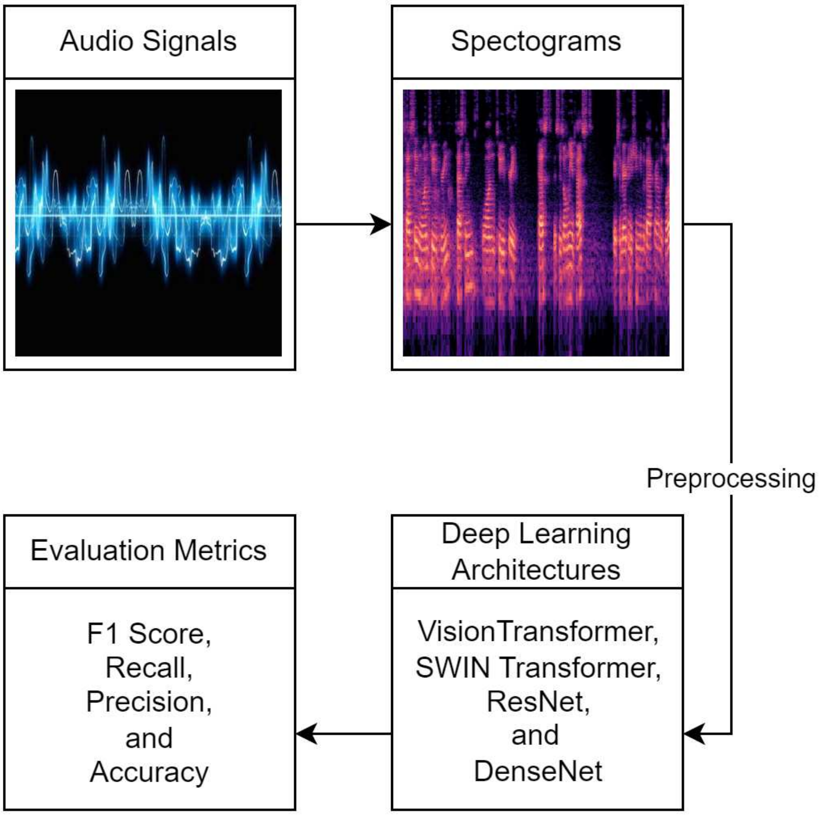
Audio Signals
Spectograms
Preprocessing
Evaluation Metrics
F1 Score, Recall, Precision, and Accuracy
Deep Learning Architectures
VisionTransformer, SWIN Transformer, ResNet, and DenseNet

Figure 1. Overview of the proposed methodology.

3.1. Dataset

For our experiments, we utilized the GTZAN dataset [20], a widely recognized dataset in the fields of music information retrieval (MIR) and audio signal processing. This dataset consists of 1000 audio tracks, each with a duration of 30 s, evenly distributed across 10 different music genres. To prepare the data for our experiment, we generated spectrograms for every 7.5 s segment of the audio tracks that had an available global key. The key values used in our experiment were obtained from [52]. In total, we produced 3348 images, each sized at 180 × 40 pixels. These images were mapped to 24 global key types, encompassing 12 tonics in both minor and major modes, as well as 9 genre classes. As part of the preprocessing procedure, the images were converted to grayscale and subjected to image normalization by dividing the pixel values of each image by 255.

3.2. ResNet

The Residual Network (ResNet) serves as the baseline architecture for our experimental study. ResNet was introduced in 2015 by researchers at Microsoft Research to address the issue of vanishing/exploding gradients [53]. This problem arises when training deep neural networks with a large number of layers, as the gradients become either too small (vanishing) or too large (exploding), impeding effective learning.

To overcome this challenge, ResNet incorporates residual blocks, which utilize skip connections to bypass intermediate layers and connect the activations of a layer to further layers. By doing so, ResNets enable the network to fit the residual mapping (B(x) = A(x) + x) instead of the actual mapping (B(x) = A(x)), where A(x) represents the function the network learns from the input, and x is the input itself. This approach facilitates the movement of data across layers, allowing deeper networks to be trained without hindering the model's capacity to learn. Figure 2 provides a visualization of a residual block.

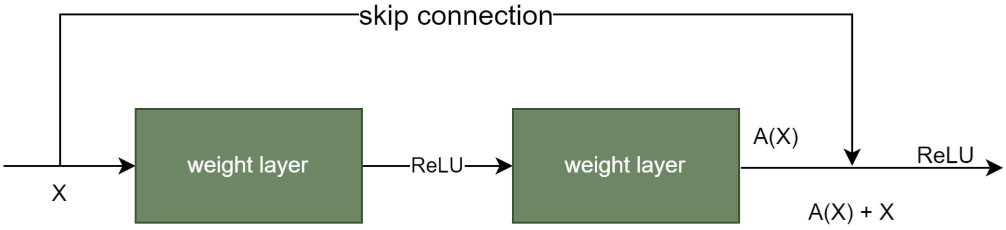

Figure 2. ResNet architecture [53].

The key advantage of skip connections is that if any layer adversely affects the model's performance, it can be bypassed, enabling the data to flow through unaffected. The underlying concept is that instead of learning to behave like an identity function independently, which requires selection of appropriate values to produce the desired output, it is simpler for the network to learn to transform the value of A(x) into zero, effectively imitating an identity function. ResNets are easier to optimize and train compared to conventional deep neural networks, resulting in enhanced accuracy for deep learning tasks.

ResNets comprise two fundamental building blocks: the identity block and the convolutional block. The identity block keeps the input dimensions unchanged across the network layers. Specifically, it takes an input tensor with dimensions of H × W × C and produces an output tensor with the same dimensions. On the other hand, the convolutional block modifies and reshapes the input tensor to match the output tensor of the identity block for subsequent addition. The convolutional block receives an input tensor with dimensions of H × W × C1 and generates an output tensor with dimensions of H × W × C2, where C2 represents the number of filters in the convolutional layer. The output tensor is then combined with the output tensor of the identity block after passing through a batch normalization layer and a ReLU activation function. This process is repeated over multiple layers to construct the complete ResNet architecture, which has demonstrated state-of-the-art performance on various benchmark datasets. In this paper, we used the ResNet50 architecture, which comprises a total of 50 layers.

3.3. DenseNet

Vanishing gradients have long been a challenge in training deep neural networks, leading to difficulties in information and gradient flow. While ResNets were introduced to address this issue, they still have some limitations. In certain variations, many layers in ResNets contribute very little, resulting in a high number of parameters. To mitigate this redundancy, DenseNets were developed in [54]. DenseNets aim to maximize information flow between layers, leverage feature reuse to enhance representational power, and reduce redundant data without making the architecture excessively deep or wide.

DenseNets employ a unique feature-reuse technique that combines feature maps from multiple layers. The mathematical formulation for the l^{th} layer in the ResNet architecture can be represented using Equation (1), where B_l represents a set of convolutional operations applied to the output of the previous layer, denoted as x_{l-1}. A key distinction between DenseNets and ResNets is the concatenation of feature maps, which increases the diversity in the input to subsequent layers and improves efficiency. DenseNets are simpler and more effective than inception networks, which also combine features from multiple layers.

$$x_l = B_l(x_{l-1}) + x_{l-1} \tag{1}$$

The resulting DenseNet architecture requires fewer parameters compared to equivalent CNN models. This is because DenseNet layers are narrow, reducing the number of parameters [54]. With reduced filters and preserved input, gradients and information flow more freely across the network. Additionally, the last layer has access to both the input and all feature maps, leading to implicit "deep supervision". The DenseNet architecture minimizes overfitting and facilitates training.

DenseNets consist of DenseBlocks, which adjust the number of filters between them while maintaining a constant feature map size within a block [54]. Transition layers are inserted between these blocks and perform operations such as 1 × 1 convolution, 2 × 2 pooling for downsampling, and batch normalization.

DenseNets leverage the combination of feature maps, increasing the channel dimension in each layer, as shown in Equation (2), where x_l represents the output feature map of the lth layer, and B_l is the operator responsible for generating k feature maps in that layer. The square brackets denote a concatenation operation, where the feature maps from previous layers $(x_0, x_1, \ldots, x_{l-1})$ are combined along a certain axis, allowing them to be processed together by B_l to produce the output feature map (x_l).

$$x_l = B_l[x_0, x_1, \ldots, x_{l-1}] \tag{2}$$

The growth rate hyperparameter (k) in Equation (2) regulates the amount of information contributed to the network in each layer. Each layer has access to the feature maps from the preceding layers, enabling every layer in the network to access the collective knowledge. Feature maps can be seen as the network's information, and by adding k feature maps, each layer contributes new insights to this knowledge base. The DenseNet architecture can be visualized by referring to Figure 3. In our research, we employed the DenseNet121 architecture, which encompasses a total of 121 layers, as the fundamental framework for conducting our experiments.

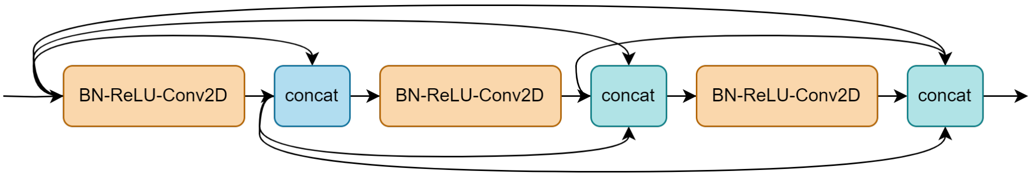

Figure 3. DenseNet architecture [54].

3.4. Vision Transformer

Vision Transformer (ViT) was introduced in [55] as a deep learning architecture for image categorization tasks. Unlike traditional convolutional neural networks (CNNs) that extract spatial features using convolutional layers, ViT treats images as token sequences.

The image is initially divided into fixed-size patches. A 2D image with dimensions of H × W is partitioned into N patches, where N = (H × W)/(P × 2). Self-attention has a quadratic cost if the image size is 48 × 48 and the patch size is 16 × 16, resulting in 9 patches in the image. Dividing the image into patches is necessary because the quadratic cost of self-attention becomes prohibitively expensive and cannot be scaled to a suitable input size.

Each patch is then linearly projected to a specified input dimension by flattening the 2D patch into a 1D patch embedding, which concatenates the pixel channels. Position embeddings are added to the patch embeddings to retain positional information, since transformers do not inherently capture the structure of the input elements. By including position embeddings in each patch, the model learns to understand the image's structure.

An additional "classification token" is also added to the sequence, which can be learned by the model. The patch-embedding vectors serve as the input sequence for the transformer encoder, with the length determined by the number of patches.

The transformer encoder consists of a stack of transformer blocks, each comprising a feedforward neural network and a multihead self-attention mechanism. The self-attention mechanism allows the model to attend to different parts of the image while processing each token, and the attended features are then non-linearly transformed by the feedforward network. The self-attention process in Vision Transformer (ViT) generates an output feature map (Z) as represented by Equation (3), where A and B are learned linear projections of the input features, each with dimensions of $(d_{\text{model}} \times d_k)$, where d_{model} is the model's hidden dimensionality, and d_k represents the dimensionality of the key vectors. C is another learned linear projection with dimensions of $(d_{\text{model}} \times d_{\text{model}})$. The parameter d_k denotes the dimensionality of the key vectors. The feature map (Z) is computed by applying a softmax operation to the dot product of A and the transpose of B scaled by the square root of d_k, then multiplied by C. This operation captures the essence of the self-attention mechanism. Equation (4) represents the overall operation of Vision Transformer, where Q, K, and V represent learned embeddings of the input image, each with dimensions of $(\text{num\_patches} \times d_{\text{model}})$, where d_{model} is the model's hidden dimensionality. The term num_patches refers to the division of the input image into a grid of patches, and each patch is treated as a separate input. Meanwhile, W_q, W_k, W_v, and W_o are learnable weight matrices, each with dimensions of $(d_{\text{model}} \times d_{\text{model}})$. The multihead attention function combines these embeddings using the self-attention mechanism, and the result is then multiplied by W_o. These matrix dimensions, along with the concept of num_patches, are fundamental to understanding how ViT processes input images and applies self-attention across multiple heads, ultimately producing meaningful feature representations for various downstream tasks in computer vision.

$$Z = \text{softmax}\left(\frac{AB^T}{\sqrt{d_k}}\right)C \tag{3}$$

$$\text{MultiHead}\Big(\text{Attention}\big(QW_q, KW_k, VW_v\big)\Big) \cdot W_o \tag{4}$$

Vision Transformer's classification head consists of a softmax activation function and a linear projection. The output of the last transformer block is projected to a lower-dimensional feature space using a linear projection. The final class probabilities are obtained by applying the softmax function to the resulting feature vector [55]. Figure 4 provides a clearer visualization of the Vision Transformer architecture. In our research, we utilized an alternative lightweight version of Vision Transformer, as presented in [56]. These layers are organized as follows: 2 initial convolutional layers for preprocessing and 21 layers distributed among three MobileNetV2 blocks, each incorporating depthwise separable convolution and batch normalization operations. Additionally, MobileViT features approximately 63 layers encompassing three blocks, each with 2, 4, and 3 transformer blocks characterized by multihead self-attention and MLP layers for local–global feature fusion. The architectural design concludes with a classification head consisting of two layers for global average pooling and dense computations, alongside an output layer for final predictions.

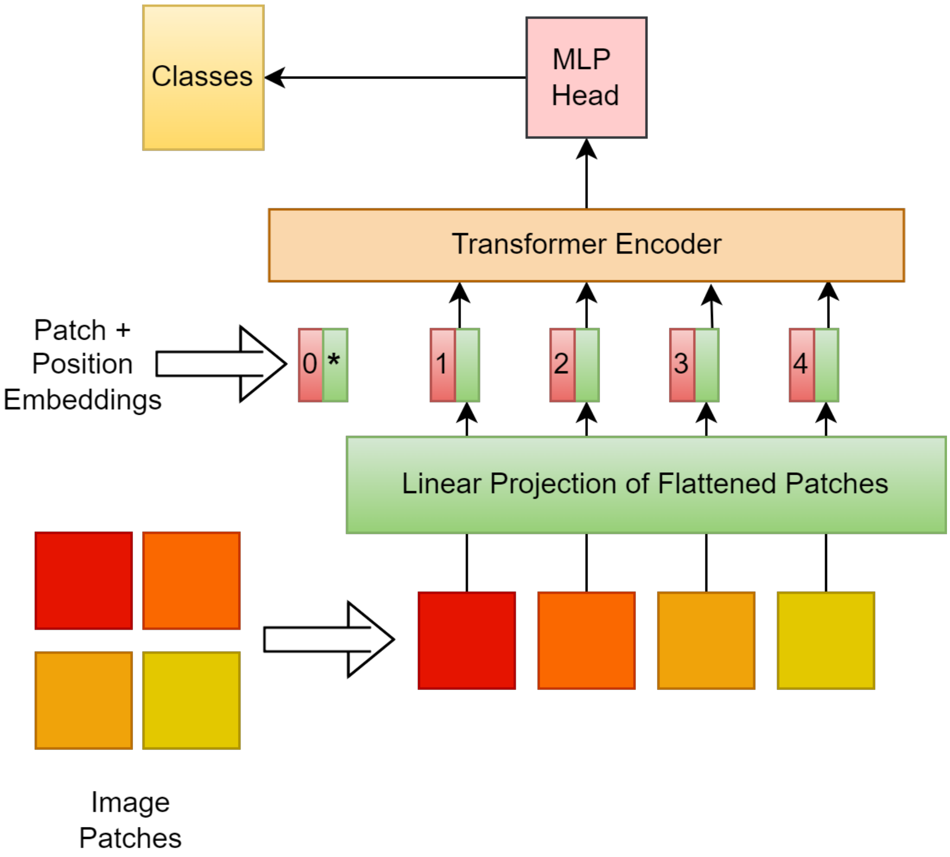
Classes
MLP Head
Transformer Encoder
Patch + Position Embeddings
0
*
1
2
3
4
Linear Projection of Flattened Patches
Image Patches

Figure 4. ViT Architecture: Image Partitioning with a Classification Token "*" and four distinct patches, each denoted by a unique color [55].

3.5. SWIN Transformer

The Shifted-Window (SWIN) Transformer [34] is a recent image processing architecture used for tasks such as image classification and object detection. It addresses two challenges faced by transformer-based models in image processing: the inefficiency of fixed-size patches for the processing of large images and the loss of contextual information when processing patches independently.

The SWIN transformer introduces a novel shifted-window technique to overcome these challenges. Instead of using fixed-size patches, the shifted-window technique divides the input image into overlapping patches and shifts each patch by a specific amount in both the horizontal and vertical directions. This approach maintains contextual information between neighboring patches, improving the model's effectiveness in capturing spatial relationships.

In the SWIN transformer, each patch is treated as a discrete "token", and its feature representation is obtained by concatenating the RGB color channel values of the pixels within the patch. The raw feature representation of each patch with dimensions of $4 \times 4 \times 3$ for a patch size of 4×4 is mapped to a target dimensionality using a linear embedding layer.

The SWIN transformer block consists of two key modules: the multihead self-attention (MSA) module and the multilayer perceptron (MLP) module, as depicted in Figure 5. Layer normalization is applied before each MSA and MLP module to address internal covariate shift, and residual connections ensure the fusion of input and processed information [34]. Additionally, the SWIN transformer utilizes a shifted-window partitioning strategy, whereby successive blocks switch between different window configurations, enhancing the model's ability to capture diverse spatial dependencies.

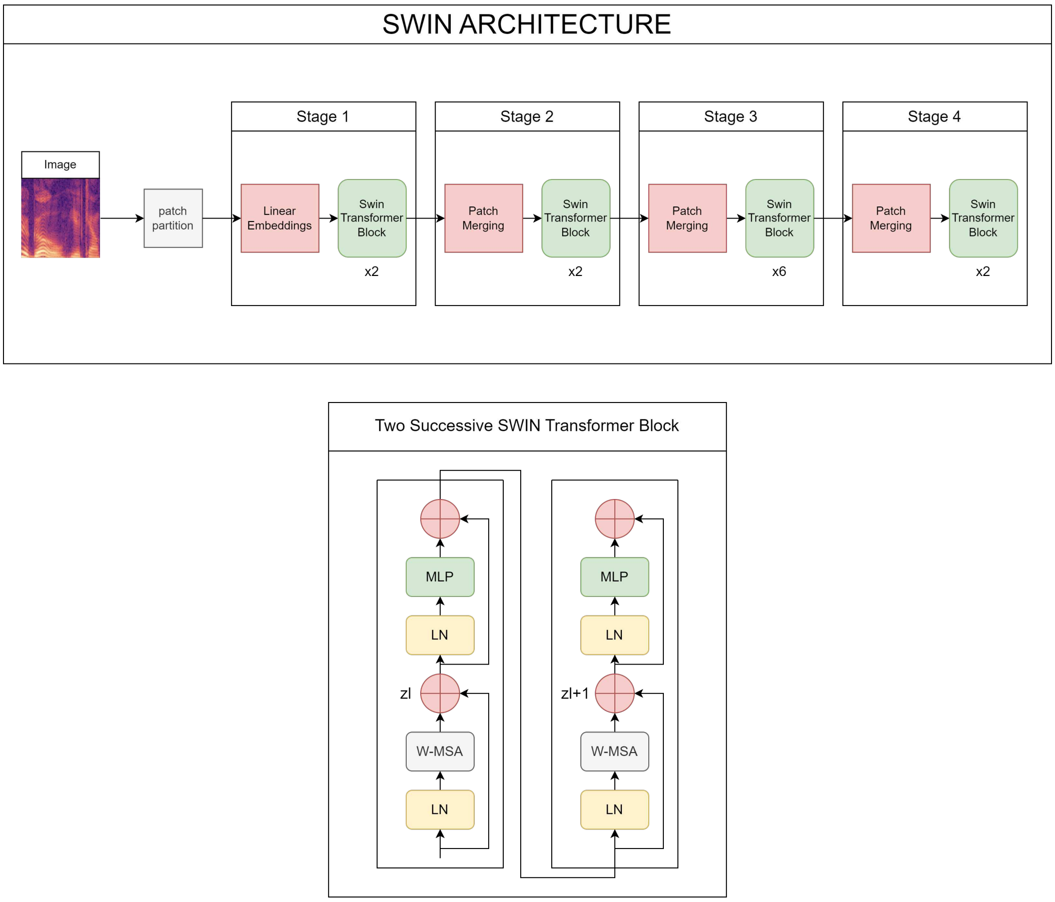

Figure 5. SWIN transformer architecture [34].

To optimize batch calculation efficiency, shifted-window partitioning may generate additional windows. However, a cyclic shifting approach towards the top-left direction is proposed, maintaining the same number of batched windows as traditional window partitioning while achieving improved efficiency [34]. Furthermore, the consideration of relative position bias incorporated during similarity calculation demonstrates better performance compared to the use of absolute position embedding or no bias term at all. Absolute position embedding, when introduced to the input, slightly degrades performance.

Figure 5 provides a visual representation of the SWIN transformer architecture, illustrating the shifted-window mechanism and the overall flow of the model. In our research, we utilized the SWIN transformer model, which includes two initial layers for patch extraction and embedding. Within the SWIN transformer framework, there are 2 instances, each integrating multiple sublayers, such as normalization, attention, dropout, and MLP layers. Additionally, the model incorporates a patch-merging layer for spatial consolidation, a global average pooling layer for dimensionality reduction, and a final dense output layer for classification. The SWIN transformer has shown promising results in various image processing tasks, highlighting its effectiveness in handling large-scale images with improved contextual understanding.

4. Results

In this section, we present the outcomes of our empirical analysis, which aimed to evaluate the performance and efficiency of several state-of-the-art deep learning architectures in the context of image classification. A rigorous 4:1 train–test split was employed to ensure fair training and validation. Each model underwent training for a carefully determined number of epochs, with early stopping callbacks to identify the best-performing models. The following key hyperparameters were used across all experiments: a learning rate of 0.001, a batch size of 32, and a total of 50 training epochs.

4.1. Performance Metrics

The models were evaluated based on a range of performance metrics:

- Accuracy: The percentage of correctly classified images;
- Precision: The percentage of true-positive predictions among all positive predictions;
- Recall: The percentage of true-positive predictions among all actual positives;
- F1 score: The harmonic mean of precision and recall, providing a balanced measure of a model's accuracy;
- Log loss: A measure of how well the model's predicted probabilities align with the actual class labels.

Table 2 summarizes the results in terms of these performance metrics.

Table 2. Performance metrics of the evaluated models.

Model	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)	Log Loss
DenseNet	91.64	91.89	92.11	91.99	0.46
ResNet	86.87	88.12	87.23	87.67	0.63
SWIN Transformer	84.64	85.87	84.91	85.38	1.08
ViT	85.22	86.44	85.28	85.85	0.88

DenseNet emerged as the best-performing model, achieving the highest F1 score of 91.99%. The SWIN transformer and Vision Transformer achieved F1 scores of 85.38% and 85.85%, respectively, showing a slight decrease in performance compared to the baseline ResNet.

4.2. Training and Testing Efficiency

Efficiency in terms of both training and testing times, as well as the number of model parameters, was assessed. Training and testing times were measured in minutes, while the number of parameters was expressed in units of 10^5. The results are presented in Table 3.

Table 3. Efficiency metrics of the evaluated models.

Model	Training Time (min)	Testing Time (min)	Training Parameters (in 10^5)
DenseNet	0.031	0.0068	69.72
ResNet	0.022	0.0068	235.77
SWIN Transformer	0.023	0.0084	2.01
ViT	0.021	0.0059	13.1

The best-performing model was DenseNet, achieving the highest F1 score of 91.99%, surpassing the baseline ResNet model by 4.32%. The SWIN transformer and Vision Transformer achieved F1 scores of 85.38% and 85.85%, respectively, exhibiting a slight decrease in performance compared to the baseline ResNet.

Table 3 provides insights into the training and testing times for each model, as well as the number of training parameters of each model. DenseNet had the longest training time, and the SWIN transformer exhibited the longest testing time among the models.

4.3. Discussion

The empirical analysis of various deep learning architectures and methodologies presented in the previous section provides valuable insights into their performance and efficiency for image classification tasks. The results highlight the strengths and weaknesses of each model and offer a basis for further discussion.

DenseNet emerged as the top-performing model, achieving the highest F1 score of 91.99%. This result is particularly noteworthy, as it surpassed the baseline ResNet model by 4.32%. DenseNet's success can be attributed to its ability to leverage dense connections between layers and efficiently reuse features. This not only enhances its predictive power but also underscores its high parameter efficiency and reduced redundancy, making it a compelling choice for image classification tasks, where both accuracy and model size matter.

The results also shed light on the inherent tradeoff between model efficiency and performance. In this specific evaluation, the models with fewer parameters did not fare well, suggesting that for complex image classification tasks, sacrificing model size may not be advisable.

The Vision Transformer (ViT) and SWIN transformer models, which leverage self-attention mechanisms for image analysis, demonstrated competitive performance. However, their F1 scores showed a slight decrease compared to the baseline ResNet model. While transformer-based models hold promise for image classification tasks, further optimization and exploration of hyperparameters may be required to unlock their full potential in this context.

Efficiency in terms of training and testing times, as well as the number of model parameters, was also assessed. DenseNet showed longer training times, which are attributed to its deeper architecture and dense connectivity. The SWIN transformer exhibited longer testing times, likely due to its shifted-window mechanism and the need to consider positional biases during similarity calculations.

In conclusion, the results of this analysis underscore the importance of selecting an appropriate deep learning architecture based on the specific requirements of the image classification task. While DenseNet showcased superior performance in this evaluation, there may be scenarios in which other models such as ViT or the SWIN transformer could provide more suitable solutions, especially when a balance between accuracy and model size is crucial. Ultimately, the choice of model should align with the specific constraints and objectives of the given image classification problem.

5. Conclusions and Future Work

In this research study, we conducted a comprehensive comparative analysis of deep learning model architectures for estimation of musical key. Our study encompassed traditional deep learning architectures and prominent vision transformer methodologies executed using the GTZAN dataset.

Among the evaluated models, DenseNet emerged as the standout architecture, achieving an impressive F1 score of 91.99%. This exceptional accuracy underscores the effectiveness of DenseNet in estimating musical key. In contrast, the vision transformer architectures, while promising in other domains, displayed suboptimal performance for this specific task. Notably, the SWIN transformer performed worse than Vision Transformer in terms of both accuracy and training time, indicating that vision transformers might be more suitable for music-related tasks.

Taking into account both accuracy and time efficiency, vision transformers hold promise as the most suitable deep learning architectures for estimation of musical key. This is especially true in scenarios where computational resources are constrained.

The findings of this study have substantial implications for the development of automated music analysis tools and can benefit researchers and practitioners in music-related fields. The utilization of vision transformers in music analysis tasks shows promise and merits further exploration.

Our future work will encompass several key areas of focus. First, we will undertake architectural exploration. This involves investigating a broader range of deep learning architectures—both traditional and transformer-based—in various musical domains. We aim to assess their applicability and performance on tasks beyond musical key estimation. Specifically, we plan to delve into the potential of deeper transformer models and assess their ability to improve both accuracy and prediction time compared to the current state-of-the-art DenseNet architecture [57–60].

Secondly, we will concentrate on refining audio signal transformation techniques. Our objective is to reduce information loss during the transformation process and, in turn, enhance classification accuracy. We will explore alternative transformation methods to achieve this goal, seeking innovative ways to effectively represent audio data.

Lastly, we will develop a unified model capable of predicting not only the musical key but also other relevant audio characteristics, such as genre and tempo. This involves adopting a multilabel classification approach, enabling a single model to identify multiple audio attributes simultaneously. This approach has the potential to streamline the classification process and reduce computational complexity.

By pursuing these future directions in our research, we anticipate advancing the field of automated music analysis. Our aim is to develop more efficient and accurate models not only for estimating musical key but also to address other pertinent audio characteristics. This research contributes to the continued evolution of automated music analysis tools, benefiting a wide range of applications in the music industry and beyond.

Author Contributions: Conceptualization, M.G., P.G., P.S. and M.S.; Data Curation, M.G., P.G., P.S., M.S. and B.A.; Writing—Original Draft, M.G., P.G., P.S., M.S. and B.A.; Methodology, M.G., P.G., P.S., M.S., B.A., V.C.G. and A.K.; Review and Editing, M.S., B.A., V.C.G. and A.K.; Supervision, B.A., V.C.G. and A.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Princess Nourah bint Abdulrahman University Researchers Supporting Project (PNURSP2023R138), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia. The authors extend their appreciation to the Deanship of Scientific Research at King Khalid University (KKU) for funding this research through the Research Group Program under grant number R.G.P.2/513/44.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Humphrey, E.J.; Bello, J.P. Rethinking Automatic Chord Recognition with Convolutional Neural Networks. In Proceedings of the 11th IEEE International Conference on Machine Learning and Applications (ICMLA), Boca Raton, FL, USA, 12–15 December 2012; pp. 357–362.
2. Mauch, M.; Dixon, S. Approximate Note Transcription for the Improved Identification of Difficult Chords. In Proceedings of the 11th International Society for Music Information Retrieval Conference (ISMIR), Utrecht, The Netherlands, 9–13 August 2010; pp. 135–140.
3. Temperley, D. *The Cognition of Basic Musical Structures*; MIT Press: Cambridge, MA, USA, 2004.
4. Krumhansl, C.L.; Kessler, E.J. Tracing the Dynamic Changes in Perceived Tonal Organization in a Spatial Representation of Musical Keys. *Psychol. Rev.* **1982**, *89*, 334. [CrossRef] [PubMed]
5. Faraldo, Á.; Gómez, E.; Jordà, S.; Herrera, P. Key Estimation in Electronic Dance Music. In *Advances in Information Retrieval, Proceedings of the 38th European Conference on IR Research (ECIR), Padua, Italy, 20–23 March 2016*; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2016; Volume 9626, pp. 335–347.
6. Noland, K.; Sandler, M. Signal Processing Parameters for Tonality Estimation. In Proceedings of the Audio Engineering Society Convention 122, Vienna, Austria, 5–8 May 2007.
7. Pauws, S. Musical Key Extraction from Audio. In Proceedings of the 5th International Conference on Music Information Retrieval (ISMIR), Barcelona, Spain, 10–14 October 2004.
8. Temperley, D. WWhat's Key for Key? The Krumhansl-Schmuckler Key-Finding Algorithm Reconsidered. *Music Percept.* **1999**, *17*, 65–100. [CrossRef]
9. Giorgi, B.D.; Zanoni, M.; Sarti, A.; Tubaro, S. Automatic Chord Recognition based on the Probabilistic Modeling of Diatonic Modal Harmony. In Proceedings of the 8th International Workshop on Multidimensional Systems, Erlangen, Germany, 9–11 September 2013; pp. 1–6.
10. Mauch, M.; Dixon, S. Simultaneous Estimation of Chords and Musical Context From Audio. *IEEE Trans. Audio Speech Lang. Process.* **2010**, *18*, 1280–1289. [CrossRef]
11. Ni, Y.; McVicar, M.; Santos-Rodriguez, R.; Bie, T.D. An End-to-End Machine Learning System for Harmonic Analysis of Music. *IEEE Trans. Audio Speech Lang. Process.* **2012**, *20*, 1771–1783. [CrossRef]
12. Pauwels, J.; Martens, J.P. Combining Musicological Knowledge About Chords and Keys in a Simultaneous Chord and Local Key Estimation System. *J. New Music Res.* **2014**, *43*, 318–330. [CrossRef]

13. Krumhansl, C.L. *Cognitive Foundations of Musical Pitch*; Oxford University Press: Oxford, UK, 2001; Volume 17.
14. Harte, C. Towards Automatic Extraction of Harmony Information from Music Signals. Ph.D. Thesis, Queen Mary University of London, London, UK, 2010.
15. Fujishima, T. Realtime Chord Recognition of Musical Sound: A System using Common Lisp Music. In Proceedings of the International Computer Music Conference, Beijing, China, 22–28 October 1999.
16. Juslin, P.N.; Sloboda, J. *Handbook of Music and Emotion: Theory, Research, Applications*; Oxford University Press: Oxford, UK, 2011.
17. Dowling, W.J.; Harwood, D.L. *Music Cognition*; Academic Press: Cambridge, MA, USA, 1986.
18. Hatten, R.S. *Musical Meaning in Beethoven: Markedness, Correlation, and Interpretation*; Indiana University Press: Bloomington, IN, USA, 2004.
19. Gómez, E. Tonal Description of Music Audio Signals. Ph.D. Thesis, Universitat Pompeu Fabra, Barcelona, Spain, 2006.
20. Tzanetakis, G.; Cook, P.R. Musical Genre Classification of Audio Signals. *IEEE Trans. Speech Audio Process.* **2002**, *10*, 293–302. [CrossRef]
21. Greener, J.G.; Kandathil, S.M.; Moffat, L.; Jones, D.T. A Guide to Machine Learning for Biologists. *Nat. Rev. Mol. Cell Biol.* **2022**, *23*, 40–55. [CrossRef]
22. Mehta, N.; Shah, P.; Gajjar, P.; Ukani, V. Ocean Surface Pollution Detection: Applicability Analysis of V-Net with Data Augmentation for Oil Spill and Other Related Ocean Surface Feature Monitoring. In *Communication and Intelligent Systems*; Springer: Singapore, 2022; pp. 11–25.
23. Senjaliya, H.; Gajjar, P.; Vaghasiya, B.; Shah, P.; Gujarati, P. Optimization of Rocker-Bogie Mechanism using Heuristic Approaches. *arXiv* **2022**, arXiv:2209.06927.
24. Whalen, S.; Schreiber, J.; Noble, W.S.; Pollard, K.S. Navigating the Pitfalls of Applying Machine Learning in Genomics. *Nat. Rev. Genet.* **2022**, *23*, 169–181. [CrossRef]
25. Gajjar, P.; Dodia, V.; Mandaliya, S.; Shah, P.; Ukani, V.; Shukla, M. Path Planning and Static Obstacle Avoidance for Unmanned Aerial Systems. In Proceedings of the International Conference on Advancements in Smart Computing and Information Security, Rajkot, India, 24–26 November 2022; pp. 262–270.
26. Bender, A.; Schneider, N.; Segler, M.; Walters, W.P.; Engkvist, O.; Rodrigues, T. Evaluation Guidelines for Machine Learning Tools in the Chemical Sciences. *Nat. Rev. Chem.* **2022**, *6*, 428–442. [CrossRef]
27. Martins, R.M.; Wangenheim, C.G.V. Findings on Teaching Machine Learning in High School: A Ten-Year Systematic Literature Review. *Inform. Educ.* **2022**, *22*, 421–440. [CrossRef]
28. Gajjar, P.; Mehta, N.; Shah, P. Quadruplet Loss and SqueezeNets for Covid-19 Detection from Chest-X Rays. *Comput. Sci.* **2022**, *30*, 89. [CrossRef]
29. Li, X. Information Retrieval Method of Professional Music Teaching Based on Hidden Markov Model. In Proceedings of the 14th IEEE International Conference on Measuring Technology and Mechatronics Automation (ICMTMA), Changsha, China, 15–16 January 2022; pp. 1072–1075.
30. Murthy, Y.V. Content-based Music Information Retrieval (CB-MIR) and its Applications Towards Music Recommender System. Ph.D. Thesis, National Institute of Technology Karnataka, Surathkal, India, 2019.
31. Ostermann, F.; Vatolkin, I.; Ebeling, M. AAM: A Dataset of Artificial Audio Multitracks for Diverse Music Information Retrieval Tasks. *EURASIP J. Audio Speech Music Process.* **2023**, *2023*, 13. [CrossRef]
32. Khan, S.H.; Naseer, M.; Hayat, M.; Zamir, S.W.; Khan, F.S.; Shah, M. Transformers in Vision: A Survey. *ACM Comput. Surv.* **2022**, *54*, 200:1–200:41. [CrossRef]
33. Liu, Z.; Wang, Y.; Han, K.; Zhang, W.; Ma, S.; Gao, W. Post-Training Quantization for Vision Transformer. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 28092–28103.
34. Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; Guo, B. Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, BC, Canada, 11–17 October 2021; pp. 9992–10002.
35. Mao, X.; Qi, G.; Chen, Y.; Li, X.; Duan, R.; Ye, S.; He, Y.; Xue, H. Towards Robust Vision Transformer. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022; pp. 12032–12041.
36. Gajjar, P.; Shah, P.; Sanghvi, H. E-Mixup and Siamese Networks for Musical Key Estimation. In *International Conference on Ubiquitous Computing and Intelligent Information Systems*; Springer: Singapore 2021; pp. 343–350.
37. Raphael, C. Music Plus One and Machine Learning. In Proceedings of the 27th International Conference on Machine Learning (ICML), Haifa, Israel, 21–24 June 2010; pp. 21–28.
38. Purwins, H.; Li, B.; Virtanen, T.; Schlüter, J.; Chang, S.; Sainath, T.N. Deep Learning for Audio Signal Processing. *IEEE J. Sel. Top. Signal Process.* **2019**, *13*, 206–219. [CrossRef]
39. Parulian, N.N.; Dubnicek, R.; Worthey, G.; Evans, D.J.; Walsh, J.A.; Downie, J.S. Uncovering Black Fantastic: Piloting A Word Feature Analysis and Machine Learning Approach for Genre Classification. *Proc. Assoc. Inf. Sci. Technol.* **2022**, *59*, 242–250. [CrossRef]
40. Ghatas, Y.; Fayek, M.; Hadhoud, M. A Hybrid Deep Learning Approach for Musical Difficulty Estimation of Piano Symbolic Music. *Alex. Eng. J.* **2022**, *61*, 10183–10196. [CrossRef]

41. Nagarajan, S.K.; Narasimhan, G.; Mishra, A.; Kumar, R. Long Short-Term Memory-Based Neural Networks in an AI Music Generation Platform. In *Deep Learning Research Applications for Natural Language Processing*; IGI Global: Hershey, PA, USA, 2023; pp. 89–112.
42. Huang, H.; Zhou, X.; He, R. Orthogonal Transformer: An Efficient Vision Transformer Backbone with Token Orthogonalization. In Proceedings of the NeurIPS, New Orleans, LA, USA, 28 November–9 December 2022.
43. Arnab, A.; Dehghani, M.; Heigold, G.; Sun, C.; Lucic, M.; Schmid, C. ViViT: A Video Vision Transformer. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Virtual, 11–17 October 2021; pp. 6816–6826.
44. Miranda, E.R.; Shaji, H. Generative Music with Partitioned Quantum Cellular Automata. *Appl. Sci.* **2023**, *13*, 2401. [CrossRef]
45. Kaliakatsos-Papakostas, M.; Velenis, K.; Pasias, L.; Alexandraki, C.; Cambouropoulos, E. An HMM-Based Approach for Cross-Harmonization of Jazz Standards. *Appl. Sci.* **2023**, *13*, 1338. [CrossRef]
46. Ramírez, J.; Flores, M.J. Machine Learning for Music Genre: Multifaceted Review and Experimentation with Audioset. *J. Intell. Inf. Syst.* **2020**, *55*, 469–499. [CrossRef]
47. Briot, J.; Pachet, F. Deep Learning for Music Generation: Challenges and Directions. *Neural Comput. Appl.* **2020**, *32*, 981–993. [CrossRef]
48. Mao, H.H.; Shin, T.; Cottrell, G.W. DeepJ: Style-Specific Music Generation. In Proceedings of the 12th IEEE International Conference on Semantic Computing (ICSC), Laguna Hills, CA, USA, 31 January–2 February 2018; pp. 377–382.
49. Schreiber, H.; Urbano, J.; Müller, M. Music Tempo Estimation: Are We Done Yet? *Trans. Int. Soc. Music Inf. Retr.* **2020**, *3*, 111. [CrossRef]
50. George, A.; Mary, X.A.; George, S.T. Development of an Intelligent Model for Musical Key Estimation using Machine Learning Techniques. *Multimed. Tools Appl.* **2022**, *81*, 19945–19964. [CrossRef]
51. Prabhakar, S.K.; Lee, S. Holistic Approaches to Music Genre Classification using Efficient Transfer and Deep Learning Techniques. *Expert Syst. Appl.* **2023**, *211*, 118636. [CrossRef]
52. GTZAN Key Dataset. Available online: https://github.com/alexanderlerch/gtzan_key (accessed on 9 July 2023).
53. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. CoRR. abs/1512.03385. 2016. Available online: https://openaccess.thecvf.com/content_cvpr_2016/html/He_Deep_Residual_Learning_CVPR_2016_paper.html (accessed on 20 September 2023)
54. Huang, G.; Liu, Z.; van der Maaten, L.; Weinberger, K.Q. Densely Connected Convolutional Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 2261–2269.
55. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An Image is Worth 16 × 16 Words: Transformers for Image Recognition at Scale. *arXiv* **2020**, arXiv:2010.11929.
56. Mehta, S.; Rastegari, M. MobileViT: Light-weight, General-purpose, and Mobile-friendly Vision Transformer. *arXiv* **2021**, arXiv:2110.02178.
57. Kanavos, A.; Kounelis, F.; Iliadis, L.; Makris, C. Deep learning models for forecasting aviation demand time series. *Neural Comput. Appl.* **2021**, *33*, 16329–16343. [CrossRef]
58. Lyras, A.; Vernikou, S.; Kanavos, A.; Sioutas, S.; Mylonas, P. Modeling Credibility in Social Big Data using LSTM Neural Networks. In Proceedings of the 17th International Conference on Web Information Systems and Technologies (WEBIST), Online, 26–28 October 2021; pp. 599–606.
59. Savvopoulos, A.; Kanavos, A.; Mylonas, P.; Sioutas, S. LSTM Accelerator for Convolutional Object Identification. *Algorithms* **2018**, *11*, 157. [CrossRef]
60. Vernikou, S.; Lyras, A.; Kanavos, A. Multiclass sentiment analysis on COVID-19-related tweets using deep learning models. *Neural Comput. Appl.* **2022**, *34*, 19615–19627. [CrossRef]