**SYSTEM ANALYSIS**

**EXISTING SYSTEM:**

Tssshe key existing systems were that they compared against:

* MFCC + Softmax Regression: Extract MFCC features, feed into softmax regression model for genre classification.
* CQT + Softmax Regression: Use Constant Q Transform instead of STFT to get spectrogram features, feed into softmax regression.
* FFT + Softmax Regression: Take FFT directly on audio, feed amplitude spectrum into softmax regression.
* MFCC + MLP: Use MFCC as input, feed into a multilayer perceptron (MLP) model with softmax output for classification.
* CQT + MLP: Use CQT spectrogram as input, feed into MLP model.
* FFT + MLP: Use FFT amplitude spectrum as input, feed into MLP.

So in summary, the key existing systems used:

* Different input audio representations: MFCC, CQT, FFT
* Simple linear models like softmax regression
* Non-linear MLP models

But they did not use convolutional neural networks or other deep learning approaches. The input features were hand-engineered rather than learned.

Let me know if you need any clarification on these existing systems! I tried to infer the details from the limited information provided in the paper.

Top of Form

**DISADVANTAGES OF EXISTING SYSTEM:**

Based on the typical audio feature extraction and classification approaches used in the existing systems described in the paper, some potential disadvantages or limitations could be:

* Hand-crafted audio features like MFCC may not capture all the relevant information for genre classification. They are engineered based on human assumptions rather than learned from data.
* Features like MFCC are extracted from short frames independently, without considering temporal context. This ignores useful temporal patterns in the audio.
* Simple linear models like softmax regression have limited modeling capacity to capture complex patterns in audio features.
* Non-linear MLPs are able to model complex patterns, but their performance still relies on the quality of input features.
* Most systems use a pipeline approach - feature engineering, feature selection, then classifier training. This is not end-to-end learning.
* Lack of shift/translation invariance - small variations in pitch or tempo can degrade accuracy of systems relying on fixed audio features.
* Unable to effectively learn from raw audio - most systems rely on engineered features rather than learning directly from spectrograms/waveforms.
* Inability to scale up - unlike deep learning approaches, traditional methods can't benefit from larger datasets.

In summary, key limitations are reliance on engineered features rather than end-to-end feature learning, lack of modeling temporal context, limited invariance properties, and disjoint training of feature extraction and classifier components. Deep learning approaches can help overcome some of these disadvantages.

**Algorithm:**

Here are some of the key existing algorithms and techniques that were used prior to this work:

* Using hand-crafted audio features like MFCCs, chroma features, spectral contrast, etc and feeding them into machine learning classifiers like SVM, KNN, Random Forests etc.
* Using aggregation and statistics of low-level features, e.g. mean, variance, histograms etc.
* Applying dimensionality reduction on hand-crafted features like PCA, ICA etc before classification.
* Using mid-level representations like bag-of-words on audio features.
* Combining multiple features at feature-level or decision-level via techniques like feature concatenation, early fusion, late fusion etc.
* Using deep neural networks like Deep Belief Networks (DBNs) and stacked autoencoders for unsupervised pre-training before classification.
* Applying recurrent neural networks like LSTMs on top of pre-extracted features for sequence modeling.
* Using 1D convolutional neural networks on raw waveform or spectrogram for feature learning.

So in summary, the key existing techniques relied heavily on hand-crafted audio features or 1D convolution, rather than 2D convolutional feature learning directly from spectrograms as proposed in this paper. The deep learning approaches focused more on unsupervised pre-training rather than end-to-end feature learning.

**PROPOSED SYSTEM:**

Here is a summary of the key points about the music genre classification paper:

* Motivation: Develop better feature representations directly from audio rather than using hand-crafted features like MFCCs for music genre classification.
* Approach: Use 2D convolutional neural network applied on spectrograms to learn features that capture timbral and temporal patterns.
* Input: 30-second audio clips converted to spectrograms using Short-time Fast Fourier Transform (STFT).
* Feature Learning: Designed 4 filters to detect patterns related to percussion, harmony, pitch slides etc. Convolved filters with spectrogram to obtain 4 feature maps.
* Subsampling: Applied 2x2 max pooling on feature maps for dimensionality reduction and translation invariance.
* Classification: Flattened feature maps and fed them into a Multilayer Perceptron (MLP) with softmax output for 10-way genre classification.
* Results: Achieved 72.4% accuracy on GTZAN dataset, outperforming MFCC+MLP (46.8%) and other baseline systems relying on hand-crafted features.
* Conclusion: Learned features from spectrograms using 2D CNNs capture more relevant information for genre classification than engineered MFCC features. End-to-end feature learning shows promise over pipeline systems.

In summary, the key ideas are - using 2D CNN on spectrograms for feature learning, end-to-end training, and demonstrating superior performance over traditional methods relying on MFCC and other hand-crafted audio features for music classification.

**ADVANTAGES OF PROPOSED SYSTEM:**

Some of the key problems this work is trying to address for music genre classification are:

1. Limitations of hand-crafted audio features like MFCCs:

* The paper mentions MFCCs lack dynamic analysis capability as they are extracted from single frames.
* MFCCs may not capture all the relevant information for genre classification.

1. Finding better representations from raw audio:

* Rather than using hand-crafted features, learn features directly from the spectrogram using convolutional neural nets.

1. Capturing temporal patterns:

* The 2D convolutional filters can capture patterns across both time and frequency dimensions of the spectrogram, unlike MFCCs.

1. Translation invariance:

* The max pooling provides some invariance to pitch shifting or tempo changes.

1. End-to-end learning:

* Compared to systems relying on engineered features, learn the feature extraction and classification together end-to-end.

So in summary, some of the key limitations the paper tries to address are:

* Finding better features from raw audio data rather than relying on hand-crafted features
* Learning features that capture temporal/spectral patterns
* Achieving some translation invariance
* End-to-end learning of features and classifier

The goal is to show convolutional neural networks can achieve better music genre classification from raw audio compared to approaches using traditional audio features.

**Algorithm:**

The proposed algorithm for music genre classification can be summarized as follows:

Input:

* Take 30-second audio clips
* Compute spectrogram using Short-time Fast Fourier Transform (STFT)
* Retain only magnitude values from spectrogram

Feature Extraction:

* Define 4 different 2D convolutional filters designed to capture different patterns in the spectrogram
* Convolve each filter with the input spectrogram to generate 4 feature maps
* This acts as a feature detector to extract useful representations

Subsampling:

* Apply 2x2 max pooling to each feature map
* Reduces dimensionality and provides translation invariance

Classification:

* Flatten the 4 subsampled feature maps into a vector
* Feed the feature vector into a Multilayer Perceptron (MLP)
* Use softmax activation in the output layer for predicting genre
* Train MLP in an end-to-end fashion via backpropagation

So in summary, the core proposed algorithm is:

1. Generate spectrogram from audio
2. Use 2D convolution to extract features
3. Max pool features
4. Feed into MLP for classification

The key aspects are using 2D convolutions on spectrograms for feature learning in an end-to-end model, rather than relying on engineered audio features like MFCCs used in prior work.