CLASSIFICADOR DE GÊNEROS MUSICAIS

João Paulo Vieira[1], Pedro Pordeus Santos[2]
[1-2]Departamento de Engenharia Elétrica e Eletrônica, Universidade Federal de Santa Catarina, UFSC, Florianópolis, Brasil,
joaopaulovieirajpv@gmail.com, pedropordeuss@gmail.com

I. Introdução

Este projeto visa desenvolver um classificador de gêneros musicais através das técnicas de aprendizado de máquina supervisionado apresentadas na disciplina EEL7513 - Tópico Avançado em Processamento Digital de Sinais. Para tanto, será utilizado o dataset GTZAN, que contém arquivos de áudio com músicas de 10 gêneros distintos e arquivos com features extraídas de cada faixa, além do seus espectrogramas de Mel. Para classificar os gêneros das faixas, serão feitas duas abordagens distintas, buscando-se o melhor resultado entre ambas. Na primeira abordagem, serão usadas características (features) extraídas das faixas de áudio como entrada de modelos classificatórios clássicos, sendo que tais features contêm informações espectrais, rítmicas, temporais e harmônicas das faixas. Na segunda abordagem, serão exploradas diferentes arquiteturas de redes convolucionais, tendo como entrada os espectrogramas das faixas. Objetiva-se avaliar o desempenho de diferentes modelos de machine learning para esse problema, de modo a encontrar uma solução ótima, realizando as extrações de atributos, transformações e otimizações necessárias. Para a abordagem de redes convolucionais, serão utilizadas técnicas de transfer learning, otimizando hiperparâmetros de diferente otimizadores como SGD e Adam, implementando técnicas de regularização e extração de representações de áudio no formato de imagens como espectrogramas da escala Mel e MFCCs.

Por fim, para verificar se o desempenho obtido no trabalho confere com o que já foi produzido academicamente, serão realizadas comparações com o trabalho "Music Genre Classification: A Review of Deep-Learning and Traditional Machine-Learning Approaches"[1] , que também estuda o desempenho de diferentes modelos de machine learning para esse problema de classificação, partindo do mesmo dataset. Para as redes convolucionais, serão comparados os resultados também com o trabalho "Convolutional Neural Networks Approach for Music Genre Classification" [2].

II. Dataset GTZAN

O dataset GTZAN, disponibilizado na plataforma Kaggle, contém um conjunto com 1000 trechos de músicas, dividido em 10 gêneros musicais: Blues, Classical, Jazz, Rock, Country, Disco, Hip-hop, Reggae, Pop e Metal. Cada trecho contém 30s de áudio com taxa de bits de 352kbps, com 1,26MB de tamanho na memória, totalizando 1,23GB de aúdio. Para facilitar o processo de treinamento, são fornecidas duas planilhas contendo diversas features extraídas do conjunto de áudios, de característica espectral, temporal, rítmica e harmônica. A primeira planilha contém uma seleção de features extraídas de cada faixa, enquanto, na segunda, são fornecidas informações sobre 10 trechos de cada faixa. Ou seja, no segundo caso, tem-se um dataset 10x maior. Por fim, o dataset também fornece Mel Spectograms, caso deseje-se utilizar redes convolucionais - ou outras arquiteturas de redes neurais - para classificar as faixas.

A. EDA Inicial

Realizando a Análise Exploratória dos Dados (EDA), foram identificado alguns problemas nos dados fornecidos (documento 'features_3_sec.csv'):

- Cada arquivo do dataset foi, teoricamente, segmentado em 10 partes, com a seguinte nomenclatura: 'jazz.00001.0.wav', 'jazz.00001.1.wav,...., 'jazz.00001.9.wav'. No entanto, foram fornecidas 9990 linhas na planilha, indicando que faltam informações referentes a 10 trechos de 3 segundos. Verificou-se que os trechos faltantes não estão associados a um mesmo arquivo, mas a 10 arquivos diferentes, o que não gerou uma perda considerável no desempenho.
- A variável 'length' tem um valor fixo de 66149 amostras. Nas informações do dataset, é dado que os arquivos de áudio possuem 30 segundos, mas os arquivos possuem tamanho variado entre si, de aproximadamente 30 segundos, o histograma da Figura 1 exibe os tamanhos em número de amostras. Não é informado como foram divididos esses trechos de 30 segundos, mas assumindo que fossem em dez partes, sem overlap ou zero-padding, não seria possível que todos os trechos de 3 segundos tivessem esse mesmo comprimento, considerando que há cerca de 307 arquivos no dataset com comprimento menor que 10*66149.
- Apesar de ter suas informações disponibilizadas no documento .csv, o arquivo 'jazz.00054.wav' está corrompido, não sendo possível conferir a validade das informações referentes a essa faixa.

III. Abordagem 1: Extração Manual das Features e Modelos de Classificação

Nessa abordagem, tem-se como dados de entrada dos modelos as features extraídas de cada faixa. Foi estudado o impacto de cada feature no desempenho dos modelos, tão como a necessidade de incluir outras. A seguir, estão dadas

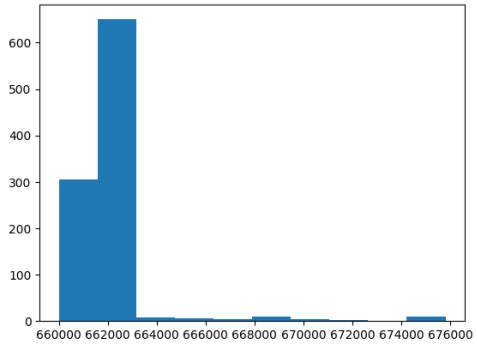

Figura 1. Histograma com o comprimento em amostras das faixas do dataset.

as descrições das features utilizadas na primeira fase desse trabalho, disponibilizadas pelo GTZAN:

- **RMS:** valor médio quadrático (energia) do sinal ao longo de vários frames.
- **Zero-crossing Rate:** a taxa no qual o sinal muda do sinal positivo para negativo, ou negativo para positivo.
- **Tempo (BPM):** número de batidas por minuto
- **Harmonics:** estimativa da frequência fundamental e harmônicas presentes no sinal
- **Componentes Percussivas:** elementos rítmicos do sinal de áudio. Representa instrumentos percussivos ou sons de curta duração.
- **Spectral Centroid:** representa o 'centro de massa' localizado no espectro, e é calculado como a média ponderada (pela energia) das frequências no espectro.
- **Spectral Rolloff:** é uma medida que representa a frequência na qual uma certa percentagem total da energia espectral está contida (por exemplo, 85% da energia do sinal). Provém informação sobre a forma e distribuição da energia espectral em um sinal.
- **Spectral Bandwidth:** representa a largura (ou distribuição) das frequências no espectro. Matematicamente, é o desvio padrão ponderado das frequências, em que os pesos são dados pela magnitude do espectro em cada valor de frequência. Provém informação sobre a faixa de frequências que o sinal cobre.
- **Mel-Frequency Cepstral Coefficients:** são coeficientes que descrevem o formato da densidade espectral de potência e da distribuição de energia de um sinal de áudio. Para calculá-los, é necessário uma série de processos: dividir a faixa em diversas janelas, extrair a FFT de cada janela, aplicar filtros triangulares espaçados na Escala de Mel (uma escala que descreve a percepção da audição humana sobre as frequências), realizar compressão logarítmica e, por fim, aplicar a DCT (Discrete Cosine Transform). Os coeficientes são dados numa matriz NxN, em que N é o número de coeficientes escolhidos. No GTZAN, foram extraídos 20x20 MFCCs.
- **Chroma Frequencies:** uma representação do espectro de um sinal com base nos 12 semi-tons (escala cromática). Provém uma representação concisa do conteúdo harmônica e tonalidade musical.

Todas as features citadas são extraídas em janelas do sinal de entrada, tendo como saída uma matriz Nx1 (ou NxN, no caso dos MFCCs). Portanto, extrai-se a média e variância associada a cada matriz (para os MFCCs, a média e variância de cada coluna), de modo a reduzir a dimensionalidade e, consequentemente, o processamento. A Tabela I exibe as features utilizadas e o número de dimensões referente a cada uma delas.

Nome	**Dimensões**
RMS (Média + Variância)	2
Zero-crossing Rate (Média + Variância)	2
Tempo	1
Harmônicas (Média + Variância)	2
Percussive (Média + Variância)	2
Spectral Centroid (Média + Variância)	2
Spectral Rolloff (Média + Variância)	2
Spectral Bandwidth (Média + Variância)	2
MFCCs (Média + Variância)	40
Chromagram (Média + Variância)	2
Nº total de dimensões:	57

Tabela I
FEATURES DO GTZAN E SUAS DIMENSÕES.

A. *Metodologia*

Essa abordagem visa classificar gêneros musicais com base em trechos de 3 segundos extraídos das músicas. Para tanto, cada arquivo presente no dataset é segmentado em dez partes, e são extraídas individualmente as features de cada trecho, o que aumenta em 10x o tamanho do dataset, em relação ao treinamento realizado com features retiradas dos trechos de 30 segundos. Observa-se que, ao separar cada trecho de 30 segundos em dez partes, torna-se necessário realizar uma divisão agrupada dos dados, para evitar o vazamento de informações.

B. *Modelos de classificação*

A seguir, na Tabela II, estão presentes os modelos de classificação utilizados nesse trabalho, tão como os hiperparâmetros avaliados para otimizá-los. Para os hiperparâmetros não citados nessa tabela, foram utilizados os valores padrões dados na biblioteca sklearn.

C. *Divisão dos conjuntos*

Realizou-se um split dos conjuntos nas proporções 60:20:20, representando, respectivamente, o conjunto de treino, validação e teste. Assim, foram otimizados os modelos, para obter os melhores hiperparâmetros, e treinou-se novamente (com a função refit) cada modelo incorporando o conjunto de validação ao de treino, tendo assim um split 80/20, o que provou melhorias consideráveis nos desempenhos. Serão apresentados resultados comparando o desempenho no teste para o split 60/20/20 e para o split 80/20, explicitando o ganho entre cada caso.

Modelos	Hiperparâmetros Fixos	Hiperparâmetros Otimizados
Regressão Softmax	max_iter=10000, solver='newton-cg', multi_class = 'multinomial'	C
SVM	max_iter=10000, kernel = 'rbf', random_state = 0	C, gamma
K-Nearest Neighbours	-	n_neighbours, weights, metric
Random Forest	random_state = 0, n_estimators = 250	max_depth, ccp_alpha
Gradient Booster	random_state = 0, n_iter_no_change = 200	n_estimators, learning_rate
Multi-layer Perceptron	random_state = 0, max_iter = 2000	activation, solver, learning_rate, hidden_layer_sizes, alpha

Tabela II
MODELOS DE CLASSIFICAÇÃO E HIPERPARÂMETROS

D. Métrica

Sabendo que o dataset é balanceado (possui uma quantidade igual de amostras para cada classe), optou-se por utilizar a acurácia como métrica de avaliação. Observa-se que, após a EDA, foi eliminada uma amostra da classe jazz, provocando um leve desbalanceamento, que não teve impacto o suficiente para justificar a utilização da acurácia balanceada. Para a segunda abordagem desse trabalho, com redes convolucionais, manteve-se a acurácia como métrica.

E. Otimização de Hiperparâmetros

Para otimizar os hiperparâmetros, utilizou-se a função GridSearchCV, da biblioteca sklearn. Nessa função, tem-se como entrada entrada os conjuntos de treino e validação, em que são testados diferentes hiperparâmetros no conjunto de validação, e se retorna o melhor resultado encontrado. Dentro da função GridSearch, foi habilitada a opção 'refit', que, após determinados os melhores hiperparâmetros, une o conjunto de treino com o de validação e retreina o modelo, melhorando consideravelmente seu desempenho e reduzindo overfitting. A seguir, nas Figuras 2 e 3, são dados dois resultados encontrados num ajuste fino para otimização dos hiperparâmetros da SVM, que se mantiveram os mesmos para a maioria dos testes realizados neste trabalho.

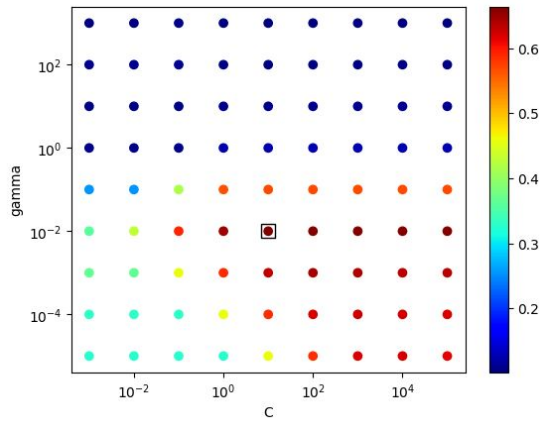

Figura 2. Otimização dos hiperparâmetros C e alpha da SVM - Ajuste grosso.

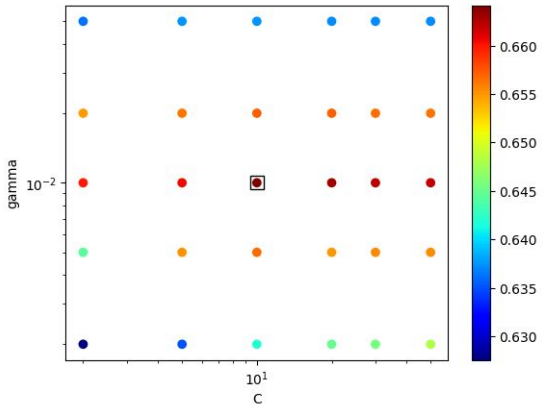

Figura 3. Otimização dos hiperparâmetros C e alpha da SVM - Ajuste fino.

F. Resultados da Literatura

Deseja-se comparar os resultados desse trabalho com os do artigo "Music Genre Classification: A Review of Deep-Learning and Traditional Machine-Learning Approaches". Esse artigo é um estudo dividido em três fases, em que, na terceira fase, denominada 'Fase C', utiliza as features dadas no dataset GTZAN. A Figura 4 exibe os resultados disponibilizados nesse artigo para diferentes modelos de classificação, além dos melhores hiperparâmetros obtidos.

Classifier	Accuracy	Training Time (s)	Hyperparameters
k-Nearest Neighbours	92.69%	0.0780	nearest neighbours=1
Multilayer Perceptron	81.73%	60.620	activation=ReLu solver lbfgs
Random Forests	80.28%	52.890	number of trees=1000, max depth=10, $\alpha = e^{-5}$, and hidden layer sizes=(5000,10)
Support Vector Machines	74.72%	3.8720	decision function shape=ovo
Logistic Regression	67.52%	3.6720	penaty=12, multi class=multinomial

Figura 4. Resultados de classificação durante a Fase C do artigo [1].

Observa-se uma alta acurácia para a maioria dos modelos, o que indica um possível vazamento de dados. Para verificar a validade dessa hipótese, treinaram-se os modelos com os dados do documento features_3_sec.csv, realizando uma divisão estratificada por classe, sem considerar que os dados estivessem agrupados. Os desempenhos obtidos estão dados na Tabela III, em que a acurácia da KNN confere com a encontrada no artigo.

Conjunto	SVM	Softmax Regression	KNN	Random Forest	Gradient Booster	MLP
Validação	89,40%	71,80%	91,60%	86,38%	81%	89,30%
Teste	87,50%	71,90%	90,50%	85,50%	78,60%	88,90%
Teste (refit)	89,10%	72,50%	92,10%	87,30%	80,80%	90,70%

Tabela III
DESEMPENHOS EM MODELOS DE CLASSIFICAÇÃO - COM VAZAMENTO DE DADOS

G. Resultados com a divisão agrupada dos dados

Na divisão realizada anteriormente, o conjunto de teste recebeu trechos de músicas que estão presentes no treino, então o modelo aprendeu a identificar quais eram tais músicas, em vez

de classificar os gêneros. Considerando esse detalhe, passou-se a realizar uma divisão estratificada agrupando os dados, de modo a garantir que não fossem vazadas informações de um conjunto para outro. A Tabela IV exibe os resultados obtidos ao corrigir a divisão dos conjuntos, utilizando como entrada os dados de 'features_3_sec.csv'. Na Tabela V, estão dados os melhores hiperparâmetros encontrados para os modelos da Tabela IV.

Conjunto	SVM	Softmax Regression	KNN	Random Forest	Gradient Booster	MLP
Validação	66,40%	61,70%	60,69%	64,38%	65,60%	66,70%
Teste (sem refit)	65,20%	61,70%	61,30%	65,40%	66,30%	66,00%
Teste (refit)	74,20%	65,90%	68,00%	72,10%	70,70%	74,00%

Tabela IV
DESEMPENHOS FEATURES INICIAIS - SEM VAZAMENTO DE DADOS - CONJUNTO CSV DATASET

Modelos	Hiperparâmetros
Regressão Softmax	C = 100
SVM	C = 10, alpha = 0.01
K-Nearest Neighbours	metric = 'manhattan', n_neighbours = 3, weights = 'distance'
Random Forest	ccp_alpha = 0.0001, max_depth = 50, n_estimators = 250
Gradient Booster	learning_rate = 0.001, n_estimators = 100
Multi-layer Perceptron	hidden_layer_sizes = (1000,500,200), activation = 'relu', solver = 'adam', learning_rate = 'adaptive'

Tabela V
MELHORES HIPERPARÂMETROS ENCONTRADOS PARA OS MODELOS DA TABELA IV

Observa-se uma queda considerável no desempenho em relação ao caso com vazamento de dados, porém agora há maior confiabilidade na capacidade do modelo em prever o estilo musical.

H. Extração Manual de Features

Considerando os resultados obtidos na Tabela IV, optou-se por adicionar mais features no modelo, tão como variar o tamanho dos MFCCs. Como não há informações sobre como foram segmentados os arquivos de 30s, extraíram-se as features manualmente, de modo a garantir maior confiabilidade nos dados, e que as novas features extraídas (na fase seguinte deste trabalho) estivessem associadas ao mesmo trecho que as features utilizadas na fase inicial. As features foram extraídas da seguinte forma:

- Para cada arquivo de 30 segundos, divide-se o seu comprimento em 10 e, descartam-se as amostras restantes, caso hajam. Por exemplo, um arquivo 661151 amostras pode ser dividido em 10 trechos de 66150 amostras. Para esse caso, é eliminada a primeira amostra e utilizam-se dez trechos de tamanho igual.

Observa-se que não houve perda considerável de informação ao eliminar tal amostra, considerando que representa 1/22050 segundos de aúdio = 45us. Como as faixas estão sendo divididas em dez partes, a máxima perda de informação pode ser de 9 amostras = 408us.

A Tabela VI exibe os resultados obtidos da extração manual das features, utilizando as mesmas dadas no arquivo 'features_3_sec.csv'.

Conjunto	SVM	Softmax Regression	KNN	Random Forest	Gradient Booster	MLP
Validação	64,84%	61,12%	59,49%	63,05%	63,50%	65,80%
Teste (sem refit)	64,70%	60,80%	59,10%	63,90%	64,30%	64,40%
Teste (refit)	73,80%	66%	68,10%	72,00%	70,00%	72,50%

Tabela VI
DESEMPENHOS FEATURES INICIAIS - SEM VAZAMENTO DE DADOS - FEATURES EXTRAÍDAS MANUALMENTE

Observam-se resultados muito próximos ao que foi obtido anteriormente (Tabela IV), o que indica que as informações extraídas conferem com o que foi fornecido no dataset, e agora há maior certeza sobre o processo realizado para extrair as informações. Mediu-se, também, o erro relativo para cada feature extraída, obtendo erros baixíssimos (na ordem de 0,1%). No entanto, observaram-se discrepâncias consideráveis nas médias e variâncias do MFCCs, indicando que a divisão realizada difere da que foi realizada no dataset, ou que foram utilizados parâmetros diferentes na extração dos coeficientes.

I. Adição de features extra

Nessa fase do trabalho, foram adicionadas cerca de 16 features no modelo, baseando-se na Fase B do artigo [1]. A seguir, estão presentes as features de característica espectral escolhidas, e uma breve descrição de seu significado físico/matemático.

- **Spectral contrast:** diferenças em magnitude entre picos e vales no conteúdo espectral de uma faixa de áudio. Provém informações sobre o contraste entre regiões do espectro, ou seja, sobre zonas distintas entre si.
- **Spectral flux:** a taxa de variação do conteúdo espectral ao longo do tempo. Provém informações sobre como varia a energia espectral. Um alto fluxo espectral indica que o espectro da faixa varia rapidamente, ou seja, há mais dinâmica nessa faixa.
- **Spectral crest:** feature que mede a proeminência de bandas de frequência dentro de um sinal. Indica a magnitude relativa dos máximos locais no espectro comparados à energia total.
- **Spectral flatness:** feature que mede regularidade no formato do espectro. Provém informações sobre o equilíbrio ou uniformidade de componentes espectrais. Uma alta spectral flatness indica que há uma maior uniformidade no espectro, ou seja, as componentes de frequência possuem um balanço de energia.
- **Spectral decrease:** indica o decaimento da energia espectral ao longo do tempo em uma faixa.
- **Spectral slope:** uma estimativa da derivada do espectro, obtida realizando regressão linear das componentes de magnitude em frequência.

- **Spectral skewness:** feature que mede assimetrias no formato do espectro. Provém informação sobre o equilíbrio entre componentes de baixa e alta frequência.
- **Spectral spread:** mede a separação entre bandas de frequência no espectro, ou seja, se o espectro está mais concentrado em uma banda, ou se abrange a maior parte das frequências.
- **Spectral variability:** uma medida da variância da magnitude nas componentes em frequência ao longo do tempo.
- **Peak smoothness:** indica a suavidade entre transições de frequências no espectro. Provém informações sobre a continuidade nas componentes.

Para essas features, que em maior parte possuem mais de uma dimensão (pois são retiradas de janelas do sinal), foram também retiradas médias e variâncias, de modo a reduzir a dimensionalidade e obter uma estimativa estatística para o valor de cada característica ao longo do sinal. A Tabela VII exibe o número de dimensões de cada feature adicionada, e o novo número total de features no modelo.

Nome	**Dimensões**
Spectral Contrast (Média + Variância)	2
Spectral Flux (Média + Variância)	2
Spectral Crest (Média + Variância)	2
Spectral Flatness (Média + Variância)	2
Spectral Decrease (Média + Variância)	2
Spectral Slope	1
Spectral Skewness	1
Spectral Spread	1
Spectral Entropy	1
Spectral Variability	1
Peak Smoothness	1
Features GTZAN	57
Nº total de dimensões:	73

Tabela VII
NOVAS FEATURES EXTRAÍDAS E SUAS DIMENSÕES.

Realizando, novamente, o treinamento e otimização de hiperparâmetros, houve ganho de desempenho em todos os modelos. A Tabela VIII exibe os resultados na validação, teste e teste com refit (split 80/20) para a SVM e a MLP, os dois modelos que apresentaram melhor desempenho na fase inicial de testes.

Conjunto	**SVM**	**MLP**
Val	68,20%	69,01%
Teste (sem refit)	68,20%	69%
Teste (refit)	76,70%	75,92%

Tabela VIII
DESEMPENHOS NA SVM E MLP: FEATURES MANUAIS + EXTRAS

Na otimização, mantiveram-se os melhores hiperparâmetros dados na Tabela V. Observou-se um ganho de 2,9% na SVM, enquanto um ganho de 3,42% na MLP com o adição das features, indicando que possuíram um impacto positivo no modelo. Além disso, o ganho em desempenho com o refit foi de 8,6% para a SVM, e 6,72% para a MLP, indicando que retreinar o modelo incorporando o conjunto de validação tem impacto considerável. Considerando a variação estatística do desempenho da MLP, realizaram-se 5 treinamentos, de modo a se considerar a média entre os resultados o valor que seja representativo do modelo. A Tabela IX exibe os resultados para esses testes, sendo a média entre eles o desempenho da MLP apresentado na Tabela VIII.

Desempenhos [%]	76,88	76,18	75,43	74,67	75,42
Média		75,92%	**Desvio Padrão**		0,46%

Tabela IX
VARIAÇÃO ESTATÍSTICA DO DESEMPENHO DA MLP - FEATURES EXTRA

J. Data Augmentation

Em todas as fases do teste, obtiveram-se resultados no treino próximos a 100% e uma grande disparidade entre o desempenho de treino e validação, indicando alta capacidade dos modelos e overffiting. Como o conjunto de dados é limitado, optou-se por realizar data augmentation, manipulando os arquivos de áudio presentes no dataset. Fizeram-se as seguintes modificações no conjunto de treino:

- Normalização do ganho;
- Pitch shift (aumento de 5%);
- Speed (aumento de 5%);
- Reverb;
- Echo;
- Filtro passa-baixas em 8kHz;
- Adição de contraste.

Todos esses efeitos foram implementados através da biblioteca TorchAudio, que possui uma gama de efeitos disponíveis. Após verificar os arquivos de áudio gerado, e analisá-los espectralmente, conferiu-se que ainda mantinham os elementos associados a cada estilo, e realizou-se a extração de suas features, adicionando-as ao modelo. Para ambos os modelos (SVM e MLP), foram observadas quedas de desempenho, indicando que as modificações escolhidas não os favoreceram. Na SVM, o desempenho no teste caiu para 75%, e na MLP, para 73.46%.

K. Variando o número de MFCCs

Utilizando o melhor modelo obtido na Tabela VIII (SVM com adição de features), variou-se o número dos MFCCs, extraindo novamente suas médias e variâncias. A Tabela X exibe os resultados obtidos para 15, 20, 25, 30 e 40 MFCCs, em que o melhor resultado obtido foi para 25, com ganho percentual de 0,3%. Assim, o número total de features do modelo passou a valer 83.

Nº MFCCs	15	20	25	30	40
Desempenho Teste (refit)	73,20%	76,70%	77%	76,00%	75,10%

Tabela X
DESEMPENHOS PARA O CONJUNTO DE FEATURES MANUAIS + EXTRAS - VARIANDO NÚMERO DOS MFCCS (SVM)

L. Testes com o Histograma de Batidas

Foi verificado, também, se adicionar o histograma de batidas como feature poderia melhorar o modelo, sendo esta uma feature com largo comprimento (cerca de 109). No entanto, não houve um ganho que justificasse a incluir nos modelos, considerando que impactou negativamente o tempo de treino.

M. Features mais importantes para o modelo

Para avaliar quais foram as features mais importantes para os modelos, computaram-se os Shap Values, uma abordagem que utiliza a teoria de jogos para determinar as variáveis mais importantes para um modelo, estimando seus pesos. Através da função DeepExplainer, que extrai um valor aproximado para o Shap Values de uma rede neural / profunda em formato TensorFlow, foi obtida uma estimativa das 30 features mais importantes para o modelo, dadas na Figura 18, disponível no apêndice. Observa-se que dentre as 30 features mais importantes, 9 delas pertencem ao grupo de features extra adicionados, 13 são médias e variâncias dos MFCCs e as 8 restantes pertencentes ao grupo de inicial.

N. Redução de dimensionalidade

Para reduzir impacto no processamento, ao mesmo tempo mantendo o desempenho do modelo, testou-se um número distinto das features mais impactantes. Para as 30 features mais importantes, houve queda de 5,3%, na SVM. Para as 40 features mais importantes, uma queda de 1,6%. Por fim, utilizando as 50 features mais importantes, obteve-se um ganho de 0.1%, passando a ter um desempenho máximo alcançado de 77,1% na SVM. Desse modo, reduziu-se o número total de features de 83 para 50.

O. Ensemble

Da forma como foram segmentados os dados de entrada, o modelo prevê 10 saídas para cada música, avaliando individualmente cada trecho de 3 segundos. Buscando melhorar sua acurácia, realizou-se um ensemble da saída do conjunto de teste para o melhor modelo encontrado (SVM, 77,1%), tirando a média aritmética entre as probabilidades. Nesse caso, para cada 10 previsões do modelo, tem-se apenas uma saída, agora ponderada. Com essa abordagem, houve melhoria impactante, com um novo desempenho de 86,4% (ganho de 9,4%), agora prevendo as saídas para o conjunto de teste com 199 amostras, que antes possuía 1990 amostras, sendo cada grupo de 10 amostras consecutivas associadas a uma mesma música. A seguir, nas Figuras 5 e 6, estão as matrizes de confusão para a saída do modelo com e sem o ensemble.

Observa-se, pela Figura 5, que as classes em que o modelo menos acertou foram 'reggae', com 8 erros, 'rock', com 7 erros, e hip-hop, com 5 erros. As classes 'metal', 'classical' e 'country' tiveram 0 erros, enquanto jazz, blues, disco e pop tiveram entre 1 - 3 erros. Com isso, é possível que as features escolhidas não sejam adequadas o suficiente para classificar com precisão os gêneros em que houveram mais erros, considerando que esses correspondem a 74,7% do número total de casos em que o modelo errou. Nos resultados da Figura 6,

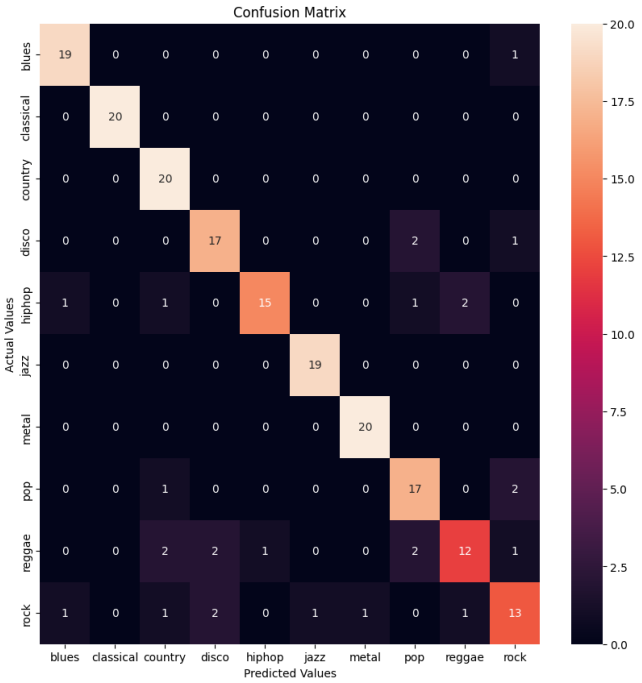

Figura 5. Matriz de confusão da SVM com ensemble.

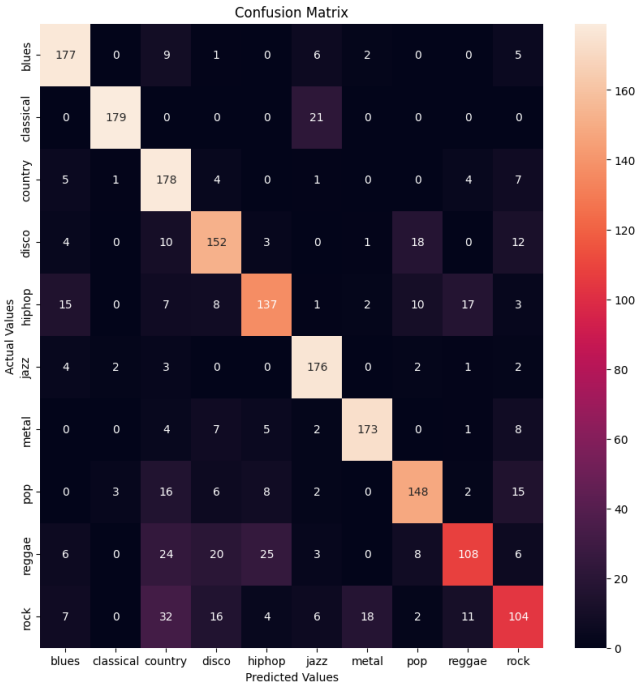

Figura 6. Matriz de confusão da SVM sem ensemble.

observa-se que as três classes que o modelo mais errou foram as mesmas.

P. Possíveis melhorias para o modelo

Considerando que estão disponíveis 30 segundos de áudio, uma possibilidade de melhoria para o modelo seria, em vez de aumentar 10x o tamanho do dataset, aumentar 10x (ou mais) o tamanho das features. Ou seja, para cada arquivo, haveriam features associadas a cada trecho de 3s, e features associadas ao arquivo completo. Isso forneceria ao modelo informações localizadas e gerais do áudio ao mesmo tempo, o que poderia aumentar sua capacidade de classificar o gênero. Dado a redução de dimensionalidade realizada anteriormente, pode-se, trabalhando com 50 features, facilitar tal procedimento. Foi feito um teste final dividindo os arquivos em 10 faixas de 3 segundos, 5 faixas de 6 segundos e 2 faixas de 15 segundos. Desse modo, foram incorporadas informações em trechos de 3, 6, 15 e 30 segundos de cada faixa. O melhor desempenho obtido no teste para a SVM foi de 79,5%, e para a MLP, 81,5%. Ou seja, a melhor abordagem encontrada foi treinar um modelo que realize predições para segmentos da entrada original (de 30s) e, após isso, faça o ensemble. Outra possibilidade, a ser testada, seria dividir as faixas, também, em 5 trechos de 6 segundos e 2 trechos de 15s, o que aumentaria em 70% o conjunto de dados. Observa-se que, para esse caso proposto, ainda seria necessária a divisão agrupada dos dados.

IV. Abordagem 2: Redes Neurais Convolucionais

Devido a natureza da classificação de gêneros musicais ser embasada nas características de harmonia, rítmo, tonalidade, distorção, etc., espera-se que uma arquitetura de rede neural convolucional treine filtros que destacarão as caracterísitcas musicais mais importantes para a classificação. A fim de obter o melhor modelo para a aplicação, foram testadas duas arquiteturas de redes neurais convolucionais propostas em artigos acadêmicos e também comparados os modelos pré-treinados disponíveis na biblioteca Torchvision. Foi escolhido o melhor modelo baseado na acurácia de validação e em seguida aplicadas otimizações de hiperparâmetros, diversas técnicas de regularização e inferência do modelo. Inicialmente, todos os modelos foram treinados e testados com o conjunto de espectrogramas de Mel disponibilizados pelo dataset GTZAN. É importante levar em consideração que essa é uma decisão a fim de tomar um ponto de partida para a avaliação e escolha do modelo a ser melhorado posteriormente, tendo em vista que diversos modelos aplicados poderiam ter desempenhos melhores em dados no formato MFCC.

A. Implementação de modelos da literatura

O artigo de Ndou, Ajooda e Jadhav (2021), propõe uma arquitetura de rede convolucional construida com 5 blocos convolucionais compostos por: Uma cama convolucional de kernel (3,3), stride (1,1), padding ”same”; Função de ativação ReLU; Max Pooling com stride e janela (2,2); Regularização dropout de razão 0.2. Ao fim da arquitetura se encontra uma camada conectada implementando a função de ativação Softmax. Como o artigo não menciona quantos filtros por camada convolucional, se implementou 32, 64, 128, 256 e 512 filtros, nesta ordem para os 5 blocos convolucionais. O split utilizado também não é mencionado pelo artigo, assim, testou-se o modelo com dois splits diferentes: 60% de conjunto de treino, 20% de conjunto de validação e 20% de conjunto de teste; e 80% para treino, 10% para validação e 10% para teste. O otimizador utilizado foi o SGD e batchsize de 64. O learning rate que apresentou melhores resultados foi o de 0.01, sendo testados de 0.1, 0,001 e 0.0001 onde nenhum além do learning rate de 0.01 apresentou um comportamento aparente de convergência para 120 épocas

Para o split 60-20-20 foram testados os seguintes valores de momento e penalidade L2 (weight decay), as épocas foram ajustadas baseado no valor de momento buscando atingir a máxima acurácia de treino e por fim obter o valor máximo de acurácia de validação.

Momentum	**Weight_decay**	**Épocas**	**Acurácia de Validação**
0.9	0.00001	200	52%
0.9	0.0001	200	47%
0.9	0.001	200	58%
0.8	0.00001	200	52.5%
0.8	0.0001	200	49.5%
0.8	0.001	200	52.5%
0.7	0.00001	300	49%
0.7	0.0001	300	53.5%
0.7	0.001	300	45.5%
0.6	0.00001	500	46.5%

Tabela XI
Otimização de Hiperparâmetros do modelo de Ndou, Ajoodha e Jadhav(2021) para o split 60-20-20

O melhor resultado obtido foi de 58% de acurácia no conjunto de validação. Observou-se um considerável desvio de desempenho se comparado com o tabelado no artigo, 66,5%. Porém, a probabilidade maior é de que os parâmetros como o número de filtros e hiperparâmetros de otimizador escolhidos tenham sido diferentes, assim como o random state das funções utilizadas, que também influencia na reprodutibilidade dos resultados. Na Figura 7 é dado o gráfico de acurácia de treinamento do modelo para o split 60-20-20 com learning rate de 0.01, weight decay de 0.001 e 200 épocas.

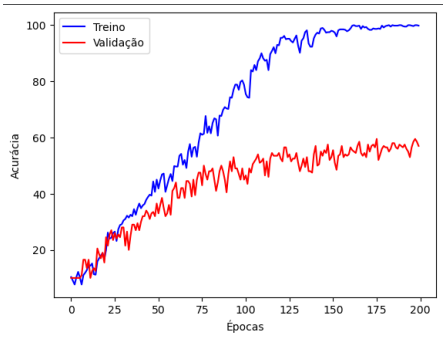

Figura 7. Melhor resultado de treinamento do modelo de Ndou, Ajoodha e Jadhav para o split 60-20-20.

Para o split 80-10-10 foram testados os seguintes hiperparâmetros gerando seus respectivos resultados de acurácia de validação, dados na Tabela XII.

Momentum	Weight_decay	Épocas	Acurácia de Validação
0.8	0.001	250	62%
0.8	0.0001	250	61%
0.7	0.01	500	60%
0.9	0.01	200	63.1%

Tabela XII
OTIMIZAÇÃO DE HIPERPARÂMETROS PARA O MODELO DE NDOU, ADOODHA E JADHAV(2021) PARA O SPLIT 80-10-10

Modificar o split ajudou consideravelmente o modelo a convergir para uma acurácia por volta dos 60% para vários hiperparâmetros diferentes. Sendo uma hipótese possível para os resultados de 66% obtidos no artigo em questão, um split mais favorável ao conjunto de treino como é o caso do 80-10-10. Contudo, pelo fato do valor reduzido de amostras do conjunto de testes trazer uma confiança menor nos resultados, foi utilizado o split 60-20-20 para testar e decidir qual modelo desenvolver os estudos, de modo a obter uma métrica mais coerente com a aplicação.

Os hiperparâmetros que proporcionaram o melhor resultado na acurácia de validação para o split a ser utilizado foi o de momentum = 0.9, weight decay = 0.001 e learning rate = 0.01. Utilizou-se estes mesmos hiperparâmetros para comparar os desempenhos dos modelos seguintes.

O artigo de Cheng, Chang e Kuo (2020), implementa a mesma arquitetura do modelo anterior, porém com algumas modificações de hiperparâmetros: As camadas de dropout são de probabilidade 0.5, o otimizador utilizado é o Adam e o batch size de treino é de 32. Em relação aos outros hiperparâmetros como learning rate e regularização L2, estes não são especificados, dessa forma repetiu-se os mesmos valores que obtiveram os melhores resultados para o modelo anterior.

A acurácia de validação obtida foi de 41.5%, sendo bastante diferente dos 77% apresentado pelos autores. Novamente, a falta de informação em relação à quantidade de filtros e hiperparâmetros utilizados influenciam na diferença de resultados e possivelmente, se aplicados a outros modos de representação dos audios de modo visual como MFCC's os resultados poderiam se aproximar.

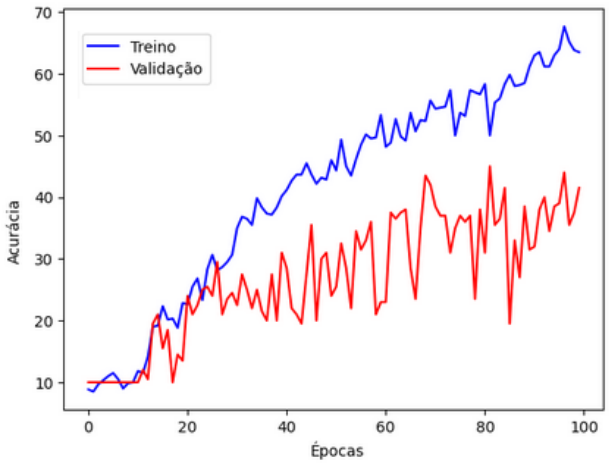

Figura 8. Treinamento do modelo de Cheng, Chang e Kuo (2020) para os hiperparâmetros descritos no artigo

Em seguida, o modelo foi impelementado de acordo com o otimizador e os melhores hiperparâmetros encontrados para o modelo de Ndou, Ajoodha e Jadhav (2021), observando um ganho considerável na acurácia de validação de 58.5%.

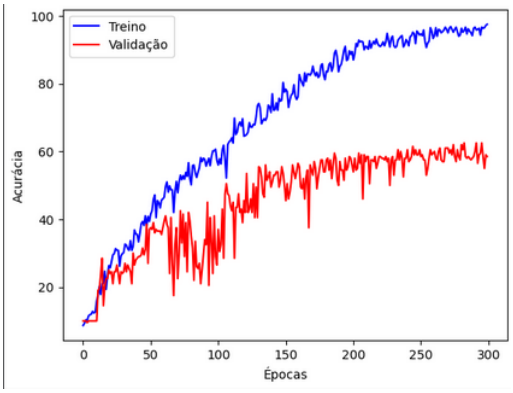

Figura 9. Gráfico de treinamento do modelo de Cheng, Chang e Kuo (2020) para os hiperparâmetros otimizados

B. Avaliação de modelos pré-treinados

A fim de encontrar o melhor modelo para a aplicação, foram testados modelos pré-treinados no dataset ImageNet disponibilizados pela biblioteca torchvision que apresentaram convergência para o GTZAN. Todos foram inicializados com os pesos da IMAGENET1K_V1 e a camada de saída devidamente alterada para 10 saídas de classificação. O otimizador aplicado e seus hiperparâmetros foi o que apresentou as melhores respostas para os modelos anteriores: SGD com learning rate = 0.01, momentum = 0.9, weigh decay = 0.001 e treinados em 100 épocas pois devido à inicialização dos pesos da ImageNet, o tempo de treinamento foi reduzido se comparado aos modelos próprios dos artigos testados anteriormente.

Modelo	**Acurácia de Validação**	**Número de Parâmetros**
Resnet18	68,0%	11.7M
AlexNet	63,0%	61.1M
VGG16	70,5%	138.4M
DenseNet161	71,0%	28.7M
GoogleNet	65,0%	6.6M
ShuffleNet_v2_x1_0	64,5%	1.4M
MobileNet_v2	66,0%	3.5M
ResNext50_32x4d	63,0%	25.0M
Wide_resnet50_2	63,5%	68.9M

Tabela XIII
ACURÁCIA DE VALIDAÇÃO E NÚMERO DE PARÂMETROS PARA MODELOS PRÉ TREINADOS DA BIBLIOTECA TORCHVISION

Comparando todos os modelos treinados e testados até o momento, observa-se que os que obtiveram melhor desempenho na acurácia de validação são a ResNet18, VGG16 e DenseNet161. Levando em consideração o número de parâmetros do modelo que funciona como indicativo do tempo de treinamento e necessidade de poder computacional, o modelo final optado para os estudos e aprimoramentos seguintes foi a DenseNet161 pelo seu desempenho de 71% de acurácia de validação e por ter uma quantidade razoável de parâmetros treináveis.

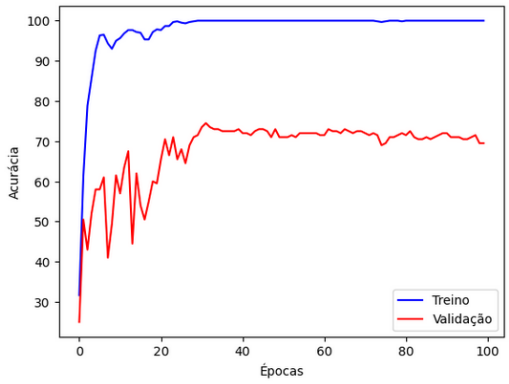

Figura 10. Gráfico de treinamento do modelo DenseNet161

C. Aprimoramentos do modelo escolhido: Densenet161

1) Extração própria de espectrogramas: Para conseguir melhor resolução no tempo e frequência dos espectrogramas e obter mais informações de cada música por imagem, foi implementada a extração dos espectrogramas pela biblioteca torchaudio. A extração foi feita com 2048 FFTs por janela e 256 bancos de filtro de escala Mel, a imagem final foi dimensionada para 128x128, como entrada para o modelo. Seu desempenho foi aferido com os mesmos hiperparâmetros aplicados do conjunto de imagens original. Foi evidente o ganho considerável na acurácia de validação obtida de 75.5% em 100 épocas, cerca de 5% a mais que os espectrogramas nativos do dataset. Pode-se observar também, pelo gráfico da acurácia de validação, que o modelo se beneficiaria da técnica de early stopping, salvando os pesos do modelo cuja acurácia de validação foi máxima já que existem picos de acurácia maior no meio do gráfico que no ponto final. Pelo gráfico de aprendizado do modelo, também optou-se por reduzir o número de épocas para 70 para agilizar os tempos de treino, visto que não há muita variação após 70 épocas.

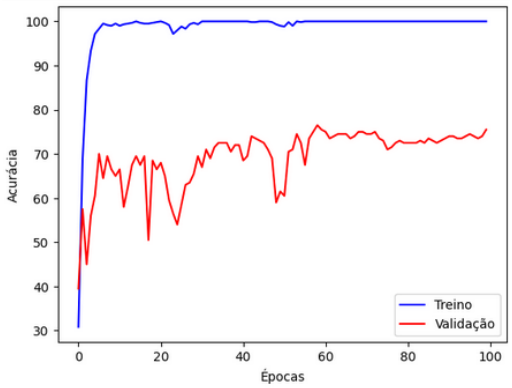

Figura 11. Gráfico de treinamento do modelo DenseNet161 para os espectrogramas extraídos

2) Implementação de EarlyStopping: No treinamento do modelo foi implementada a técnica de Early Stopping, onde o modelo salva os pesos cuja acurácia de validação resultante foi mais alta dentre as épocas decorridas. Além disso, ele interrompe o treinamento caso após um número específico de épocas, denominado paciência, não ocorra melhora na acurácia máxima de validação. Aplicado o Early Stopping foi possível obter uma acurácia de validação de 79.5% na época 35 (20 épocas de paciência), um ganho espressivo.

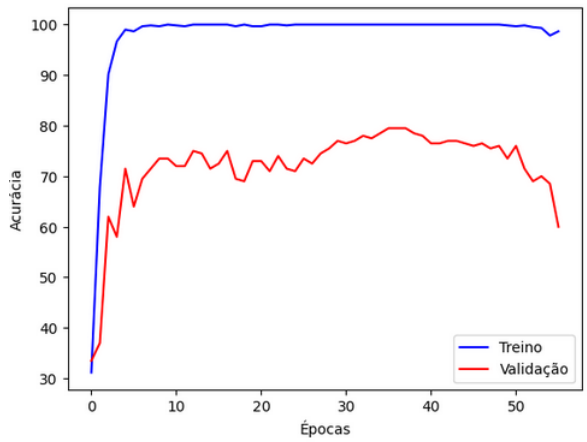

Figura 12. Gráfico de treinamento do modelo DenseNet161 aplicado EarlyStopping

Para os testes em seguida, todos foram aplicados a técnica de early stopping com suas respectivas épocas máximas e paciências.

3) Otimização de Hiperparâmetros para espectrogramas de 30 segundos: Buscando encontrar os melhores hiperparâmetros para o otimizador utilizado, foram testadas diversas combinações de hiperparâmetros a fim de mapear as acurácias de validação. Em seguida, o otimizador foi alterado para o Adam e novamente os testes a fim de encontrar o melhor desempenho foram aplicados.

Momentum	Learning Rate	Weight Decay	Época Limite	Época de Max. Acc	Paciência	Acurácia de Validação
0.9	0.01	0.001	70	35	20	79.5%
0.9	0.01	0.0001	70	35	30	76.0%
0.9	0.01	0.00001	70	47	30	76.0%
0.9	0.001	0.001	100	6	40	68%%
0.9	0.001	0.0001	100	14	40	70.0%
0.9	0.001	0.00001	100	70	40	70.5%
0.9	0.0001	0.001	120	109	50	65.0%
0.9	0.0001	0.0001	120	80	50	67.5%
0.9	0.0001	0.00001	120	88	50	67.5%
0.8	0.01	0.001	70	11	30	72.0%
0.8	0.01	0.0001	70	13	30	73.5%
0.8	0.01	0.00001	70	22	30	72.5%
0.8	0.001	0.001	100	57	40	70.5%
0.8	0.001	0.0001	100	20	40	70.0%
0.8	0.001	0.00001	100	51	40	69.5%
0.8	0.0001	0.001	120	116	50	65.5%
0.8	0.0001	0.0001	120	73	50	66.5%
0.8	0.0001	0.00001	120	118	50	67%

Tabela XIV
TABELA DE OTIMIZAÇÃO DE HIPERPARÂMETROS PARA O MODELO DENSENET161 COM OTIMIZADOR SGD

Learning Rate	Weight Decay	Época Limite	Época de Max. Acc	Paciência	Acurácia de Validação
0.01	0.001	70	26	30	52.5%
0.0005	0.0005	100	49	30	79.0%
0.0005	0.0001	100	42	30	78.0%
0.0001	0.01	70	42	30	75.5%
0.0001	0.001	70	14	30	75.5%
0.0001	0.0001	100	16	30	76.5%
0.0001	0.0005	100	60	30	79.0%
0.00001	0.001	70	51	30	73.5%

Tabela XV
TABELA DE OTIMIZAÇÃO DE HIPERPARÂMETROS PARA O MODELO DENSENET161 COM OTIMIZADOR ADAM

Observa-se que ambos os otimizadores chegaram em um máximo de acurácia de validação de cerca de 79% a 80%. Porém, levando em consideração a velocidade de convergência, onde a época de máxima acurácia do otimizador SGD foi de 35 enquanto do Adam foi de 49 e 60, é possível concluir que o otimizador SGD, dentro do escopo de hiperparâmetros analisados, converge mais rapidamente para uma acurácia de validação maior. Os melhores hiperparâmetros encontrados continuam sendo o otimizador SGD com learning rate = 0.01, weight decay = 0.001 e 70 épocas.

4) Data Augmentation: Buscando reduzir o overfitting elevando a acurácia de validação, decidiu-se aumentar o conjunto de treinamento através de técnicas de data augmentation para espectrogramas. Foram aplicadas duas funções de data augmentation da biblioteca torchaudio, nas imagens de treino e validação: TimeMasking e FrequencyMasking, que consistem em retirar componentes de uma faixa de frequência ou de tempo do espectrograma. Implementadas as funções com parâmetros time_mask_param e freq_mask_param iguais a 80, realizou-se a inferência no conjunto de testes treinando o modelo no conjunto de treino + validação e o resultado de 35.5% de acurácia de teste foi obtido. A queda tão espressiva do desempenho do modelo é um indicativo de que as funções de data augmentation escolhidas não são compatíveis com a aplicação em questão. Uma possível estratégia para obter novos espectrogramas por data augmentation que venham a melhorar o desempenho do modelo, seria obter os espectrogramas dos áudios .wav que sofreram modificações por funções de data augmentation para audio, já que as funções para as imagens não se mostraram efetivas.

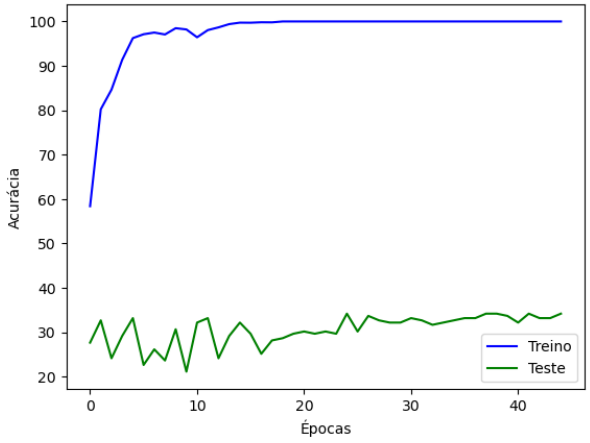

Figura 13. Gráfico de treinamento do modelo DenseNet161 aplicado Data Augmentation

5) Extração de espectrogramas para 3 segundos de áudio: Outra possibilidade além do data augmentation para aumentar o conjunto de treinamento buscando reduzir o overfitting, é extrair os espectrogramas de faixas de tempo menores das músicas. Neste caso realizamos a extração para períodos de 3 segundos de música para cada áudio do dataset. Foram aplicados os mesmos parâmetros de extração dos espectrogramas de 30s (número de FFTs por janela = 2048, número de bancos de filtro de mel = 256). Porém o resize da imagem final foi de 64x64 em contrapartida com as dimensões de 128x128 dos espectrogramas de 30 segundos devido a limitações de tempo de computação. Ao fazer a divisão de 3 segundos e retirar novamente os espectrogramas, multiplica-se em 10 vezes o conjunto de imagens, aumentando seu tamanho e tempo de treinamento do modelo na mesma razão. Reduzindo a imagem para 64x64, pode-se agilizar o processo consideravelmente. Um leve ganho de acurácia de validação é observado, porém uma estabilidade ao decorrer das épocas após a convergência para cerca de 80% é muito considerável. O resultado obtido foi de 80.85% de acurácia máxima na época 60.

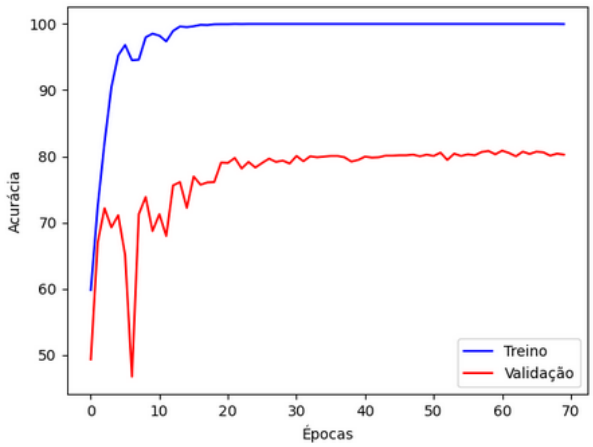

Figura 14. Gráfico de treinamento do modelo DenseNet161 para espectrogramas de 3 segundos

Por causa do aumento do dataset, o tempo de treinamento foi muito mais elevado que o de 30 segundos, tendo em vista que estamos tratando de 10000 imagens, mesmo reduzindo suas dimensões para 64x64.

6) Otimização de hiperparâmetros para espectrogramas de 3 segundos: Novamente, testou-se ambos os otimizadores SGD e Adam com diferentes hiperparâmetros para buscar a melhor combinação que entregasse o melhor desempenho na acurácia de validação agora para o conjunto de espectrogramas de 3 segundos.

Momentum	Learning Rate	Weight Decay	Época Limite	Época de Max. Acc	Paciência	Acurácia de Validação
0.9	0.1	0.001	70	42	30	61.35%
0.9	0.01	0.01	70	15	30	72.35%
0.9	0.01	0.001	70	60	30	80.85%
0.9	0.01	0.0001	70	43	30	76.6%
0.9	0.01	0.00001	70	22	30	79.0%
0.9	0.001	0.001	100	64	40	73.75%

Tabela XVI
OTIMIZAÇÃO DE HIPERPARÂMETROS PARA O MODELO DENSENET161 COM OTIMIZADOR SGD PARA ESPECTROGRAMAS DE 3 SEGUNDOS

Learning Rate	Weight Decay	Época Limite	Época de Max. Acc	Paciência	Acurácia de Validação
0.0001	0.0001	70	59	30	80.15%
0.0001	0.0005	70	30	30	78.50%
0.0005	0.0001	70	28	30	80.65%
0.0005	0.0005	70	40	30	80.05%

Tabela XVII
OTIMIZAÇÃO DE HIPERPARÂMETROS PARA O MODELO DENSENET161 COM OTIMIZADOR ADAM PARA ESPECTROGRAMAS DE 3 SEGUNDOS

Os melhores hiperparâmetros continuam sendo com o otimizador SGD, learning rate = 0.01, weight decay = 0.001 e 70 épocas.

7) Inferência no conjunto de Testes com Ensemble: Finalmente, para a inferência no conjunto de testes, é possível implementar o ensemble de previsões do modelo como feito na abordagem 1. O modelo é treinado no conjunto de treino+validação de 3 segundos dos espectrogramas, porém na inferência do conjunto de testes são realizadas as previsões para os 10 trechos de 3 segundos de cada música e salvas as saídas de probabilidade. Para classificar o gênero musical do áudio de 30 segundos em questão, retira-se a média das previsões dos 10 trechos e opta-se pelo gênero de maior probabilidade. Observa-se um ganho considerável de acurácia de validação (83.9%) demonstrando o quão eficiente é o método para também para redes convolucionais e conjuntos de imagem.

8) Ensemble no treinamento do modelo: Outra maneira possível de aplicar o ensemble seria utilizá-lo também no treinamento, e não somente na inferência do conjunto de teste. Para calcular a perda e consequentemente o gradiente,

o modelo realizaria um ensemble de previsões em cada áudio de 30 segundos pela média das previsões dos trechos de 3 segundos da música. Esta média de previsões alimentaria então a função perda, incorporando assim o ensemble durante o treinamento do modelo.

Implementado o ensemble no treinamento, obteve-se um resultado de 82.9% de acurácia de teste. O resultado foi 1% abaixo do ensemble aplicado somente na inferência de teste, porém as aleatoriedades do modelo podem afetar sendo inconclusivo qual método seria o mais benéfico para o desempenho do modelo, porém ambos impactam positivamente na acurácia.

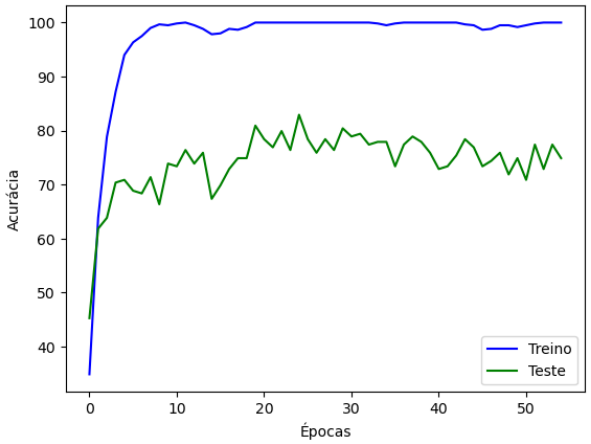

Figura 15. Gráfico de treinamento do modelo DenseNet161 com ensemble no treino

9) Extração própria de MFCCs: Outra forma de representar os áudios no formato de imagem é através dos Mel-frequency cepstral coefficients (MFFCs). Através da biblioteca librosa foram retirados MFCCs dos trechos de 3 segundos de cada música com 20 coeficientes Mel. Em seguida testou-se uma faixa de diversos hiperparâmetros para o otimizador SGD e Adam obtendo os seguintes resultados.

Momentum	Learning Rate	Weight Decay	Época Limite	Época de Max. Acc	Paciência	Acurácia de Validação
0.9	0.01	0.001	70	67	35	64.7%
0.9	0.001	0.001	70	2	35	56.8%
0.9	0.0005	0.001	701	17	35	55.0%
0.9	0.01	0.01	70	17	35	57.1%
0.9	0.01	0.0005	70	50	35	63.0%
0.9	0.01	0.0001	70	48	35	60.2%
0.8	0.01	0.001	70	51	35	62.25%
0.8	0.001	0.001	120	95	60	55.1%

Tabela XVIII
OTIMIZAÇÃO DE HIPERPARÂMETROS PARA O MODELO DENSENET161 COM O OTIMIZADOR SGD PARA DADOS DE ENTRADA NO FORMATO MFCC DE 3 SEGUNDOS

Learning Rate	Weight Decay	Época Limite	Época de Max. Acc	Paciência	Acurácia de Validação
0.0001	0.0001	100	98	50	62.7%
0.0001	0.0005	100	72	50	62.35%
0.0005	0.0001	100	85	50	65.70%
0.0005	0.0005	100	28	50	63.25%

Tabela XIX
OTIMIZAÇÃO DE HIPERPARÂMETROS PARA O MODELO DENSENET161 COM OTIMIZADOR ADAM PARA DADOS DE ENTRADA NO FORMATO MFCC DE 3 SEGUNDOS

O melhor resultado obtido foi com o otimizador Adam de 65.70% de acurácia de validação, porém os resultados foram consideravelmente abaixo dos obtidos com os espectrogramas de Mel.

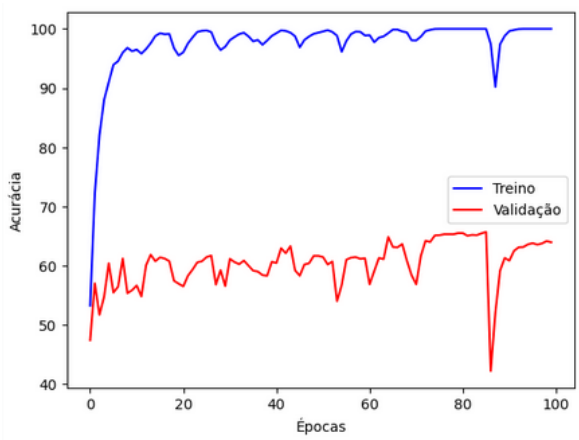

Figura 16. Gráfico de treinamento do modelo DenseNet161 para dados de entrada no formato MFCC de 3 segundos

Treinando o modelo no conjunto de treino+validação e realizando a inferência no conjunto de testes com ensemble (abordagem que gerou os melhores resultados), obteve-se uma acurácia de teste de 76.88%, um ganho de acurácia muito considerável (11.18%) porém o desempenho final ainda se encontra abaixo do treinamento com espectrogramas de Mel.

10) Análise dos resultados e possíveis melhorias: O modelo final para a arquitetura de redes neurais convolucionais, que obteve o melhor desempenho, foi a DenseNet161, com otimizador SGD de hiperparâmetros: learning rate = 0.01, weight decay = 0.001, momentum = 0.9, batch size de 64, implementação de early stopping com 30 épocas de paciência e ensemble na inferência de teste. Por se utilizar de um modelo para classificação de imagens treinados no conjunto ImageNet, demonstrou-se como o transfer learning pode ser aplicado com muita eficiência para a aplicação de classificação de gêneros musicais.

Pelas grandes variações de acurácia durante o treinamento, a técnica de early stopping se mostrou crucial para aquisição dos melhores pesos, além de agilizar o processo de otimização de hiperparâmetros pela interrupção do treinamento.

O ensemble na inferência de teste obteve o melhor resultado,

se mostrando muito eficiente para a aplicação, aproveitando a oportunidade desses tipos de dados (áudios de música) de particionar em diferentes tempos e aumentar o conjunto de treino. Porém ainda é incerto se o ensemble durante o treinamento poderia se mostrar melhor devido as aleatoriedades envolvidas no processo, sendo necessário testes mais completos da estratégia para uma conclusão definitiva.

Obtendo a matriz de confusão das previsões do melhor modelo, pode-se observar que as previsões erradas são condizentes com a proximidade de gênero. Por exemplo, o modelo se confundir com músicas do gênero discoteca com pop é bastante condizente devido às similaridades de ritmo, instrumentos, timbre etc. Em comparação com a abordagem 1, é muito interessante como as duas abordagens obtiveram matrizes de confusão muito próximas, demonstrando como mesmo para dados de entrada e estratégias diferentes, ambos os modelos conseguem convergir para um aprendizado similar.

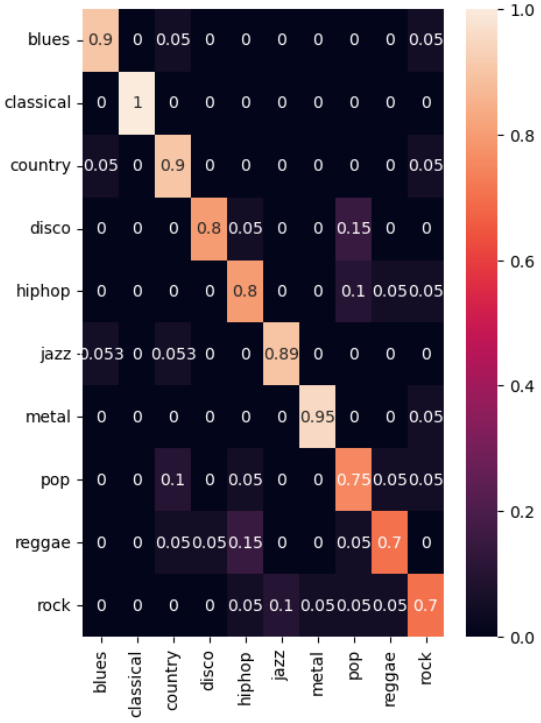

Figura 17. Matriz de confusão para o melhor modelo encontrado de arquitetura de redes neurais convolucionais - DenseNet161

Por fim, dentre as possíveis melhorias para o modelo, e futuros testes e otimizações se encontram:

- Aquisição de mais testes de combinação de hiperparâmetros para os diferentes otimizadores, estratégias e tipos de dados de entrada.
- Otimização de dimensões da imagem de entrada, tanto para espectrogramas quanto MFCCs para melhor acurácia de validação.
- Otimização dos hiperparâmetros de extração dos espectrogramas e MFCCs para melhor acurácia de validação.
- Testes com diferentes tempos para as partições dos áudios a serem aplicados o ensemble.
- Implementação de data augmentation nos áudios e em seguida a extração dos espectrogramas, tendo em vista que o data augmentation diretamente nos espectrogramas se mostrou contraprodutivo.

V. Comparações entre os resultados de cada abordagem

Por fim, na Tabela XX estão presentes os melhores resultados para cada abordagem desse trabalho, comparando os desempenhos obtidos utilizando modelos clássicos e CNN, tendo como entradas features ou espectrogramas de 30 segundos e 3 segundos (com ensemble).

Modelo	**Desempenho**
SVM - Features extra (3s) + Ensemble	86,40%
SVM - Features extra (30 segundos)	80,50%
Densenet161 - Espectrogramas de 30 segundos	79,50%
Densenet161 - Espectrogramas de 3s + Ensemble no teste	83,90%

Tabela XX
Modelos de Classificação e Hiperparâmetros

Observa-se, em ambos os casos, que houve ganho em trabalhar extraindo características de trechos menores (3s), o que pode ocorrer por diversas razões. Tem-se como hipóteses o fato de que, com um aumento considerável do conjunto de treino ao trabalhar-se com features de 3 segundos, aumenta-se a capacidade do modelo. Outra hipótese, a ser verificada, é de que a extração de características de forma mais localizada facilita a classificação de gêneros musicais. Por fim, tem-se que o melhor modelo resultante desse trabalho foi a SVM, utilizando as 50 melhores features obtidas na Abordagem 1, e realizando um ensemble da saída do teste.

Referências

[1] N. Ndou, R. Ajoodha and A. Jadhav, "Music Genre Classification: A Review of Deep-Learning and Traditional Machine-Learning Approaches,"2021 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS), Toronto, ON, Canada, 2021, pp. 1-6, doi: 10.1109/IEMTRONICS52119.2021.9422487.

[2] Y. -H. Cheng, P. -C. Chang and C. -N. Kuo, "Convolutional Neural Networks Approach for Music Genre Classification,"2020 International Symposium on Computer, Consumer and Control (IS3C), Taichung City, Taiwan, 2020, pp. 399-403, doi: 10.1109/IS3C50286.2020.00109.

VI. Apêndice

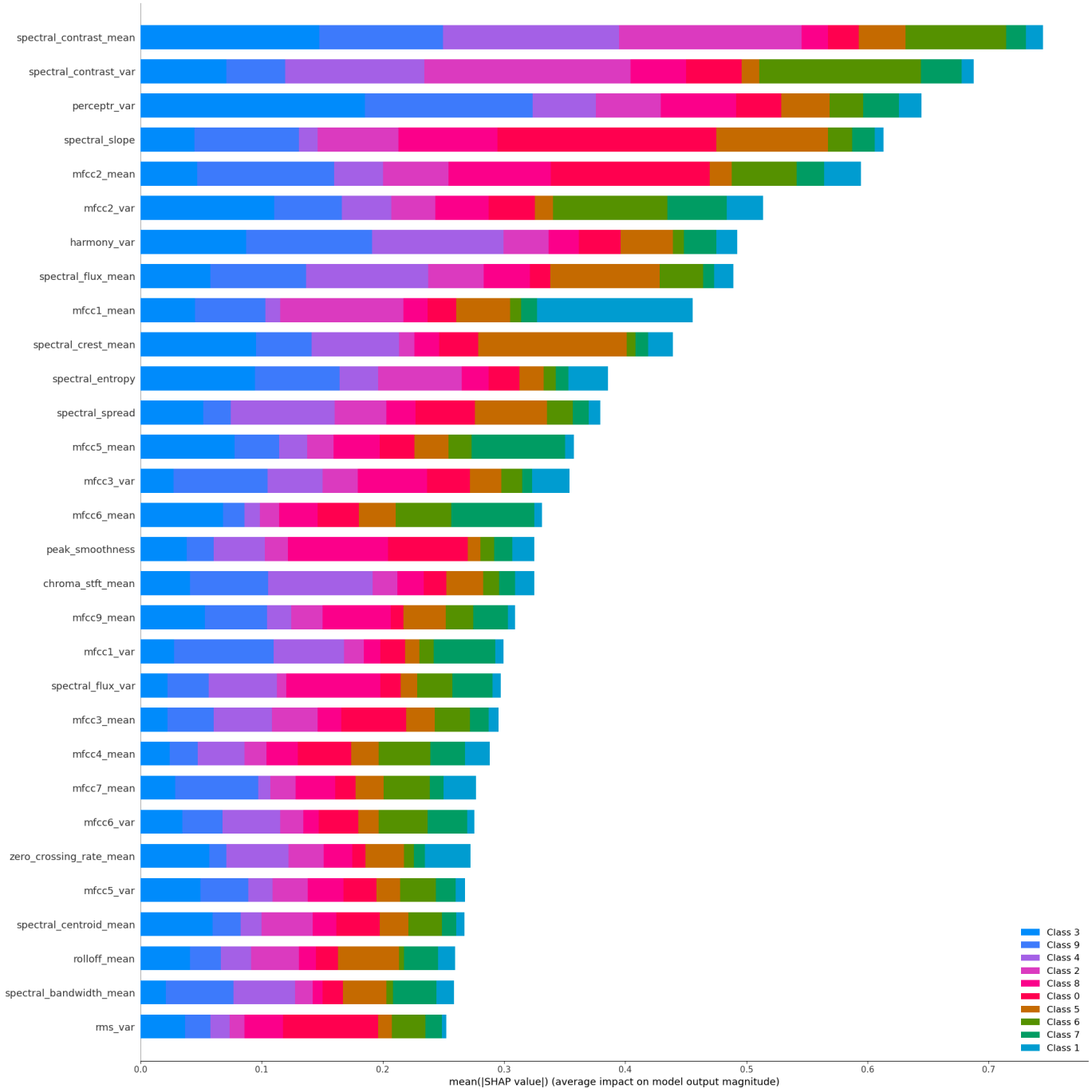

Figura 18. 30 melhores features da MLP estimadas pelos Shap Values.