Identifying musics using NCD

Mestrado em Engenharia Informática

Universidade de Aveiro

DETI - Aveiro, Portugal

TAI - Teoria Algorítmica da Informação

Gonçalo Silva [103668], Guilherme Antunes [103600], Pedro Rasinhas [103541]

v2024-06-10

Índice

1 Introdução

A identificação de músicas a partir de excertos é uma ferramenta muito pedida e apreciada pelas pessoas desde que é possível reproduzir e gravar formas de áudio. Toda a gente já deu por si a ouvir uma música na televisão ou numa festa e perguntou-se que música seria aquela.

Uma ferramenta muito conhecida e bem-sucedida nesta área é o Shazam. De um modo resumido o Shazam grava a música que estamos a ouvir, suprime o ruído e outras interferências transformando o áudio em algo mais cru e retira características dele, como as frequências mais elevadas, para criar um “pedaço" de impressão digital, ou hash. Após criada essa impressão digital vários computadores comparam-na com outras impressões digitais das músicas registadas e cada um retorna o melhor resultado que obteve e de entre esses melhores resultados, novamente, é selecionado o melhor deles e está identificada a música que estamos a ouvir.

A escolha das frequências mais elevadas como característica relevante a ter em conta está relacionada com serem essas frequências que na visão de um computador permitem distinguir uma música de outra. A sua ocorrência e o timing em que o fazem permitem desenhar um indentificador único. Se optássemos por frequências mais medianas acabaríamos por ter muitas mais e muito mais parecidas com outras músicas, o que tornaria pior a decisão de escolher que características da música guardar.

Para este projeto foi-nos sugerido que implementássemos um modelo idêntico, mas utilizando NCD (Normalized Compression Distance) para procurar a que música pertence um determinado excerto. O objetivo é retirar características de várias músicas para construir o seu identificador, construir o identificador do excerto e ver qual dos identificadores das músicas permite comprimir melhor o identificador do excerto.

A escolha de gerar um identificador ao invés de comprimir diretamente as músicas deve-se ao facto de a compressão de áudio não ser tão eficiente como a compressão de dados de texto, pela representação e singularidade pouco habituais que os ficheiros de áudio constituem, e por bastar existir um pouco de ruído no

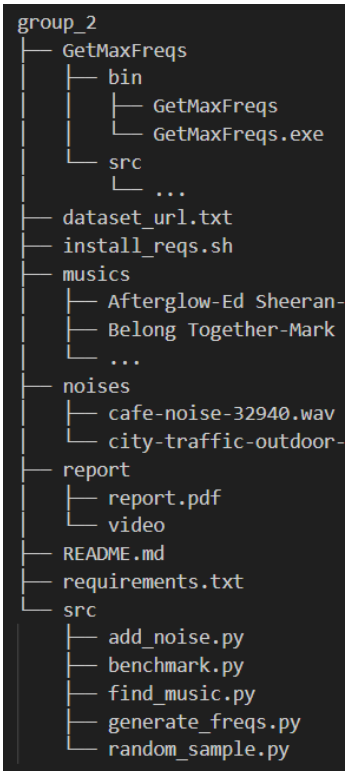
excerto a identificar que iria imediatamente impossibilitar a reutilização dos padrões anteriores para comprimir este novo.

2 Metodologia

O projeto é composto por vários programas para as diferentes fases de execução. O programa ***generate_freqs.py*** é o primeiro na linha de execução e é responsável por gerar a assinatura de cada uma das músicas do dataset utilizando o programa ***GetMaxFreq***, fornecido pelos docentes da unidade curricular, que gera uma assinatura da música tendo em conta frequências máximas presentes em excertos desta. O programa ***random_sample.py*** permite retirar uma amostra aleatória de tamanho ajustável de um ficheiro de áudio com as mesmas características dos presentes no dataset. O programa ***add_noise.py*** permite, partindo de uma amostra inicial, criar outras novas 5 amostras com diferentes tipos de alterações em cada uma. Por fim o programa ***find_music.py*** recebe uma amostra e tenta identificar a qual das músicas do dataset pertence.

Também foi desenvolvido um programa ***benchmark.py*** que executa todos os programas anteriores de forma a retirar dados de precisão do modelo.

2.1 Organização



Relativamente à organização dos ficheiros do projeto podemos encontrar na pasta ***src/*** todos os ficheiros relacionados com código desenvolvido. É nesta pasta que encontramos o código fonte do ***generate_freqs.py***, ***random_sample.py***, ***add_noise.py***, ***find_music.py*** e ***benchmark.py***. Na figura ao lado a pasta ***musics/*** contém todas as músicas que são transferidas do url presente no ficheiro ***dataset_url.txt*** e na pasta ***report*** podemos encontrar este relatório que descreve o trabalho desenvolvido. Também está disponível um ficheiro ***README.md*** que descreve como correr os programas, o ***install_reqs.sh*** que permite instalar todos os compiladores e outros pacotes necessários para executar corretamente o projeto e o ***requirements.txt*** que lista os requisitos python necessários para correr todos os programas desenvolvidos.

Na pasta ***GetMaxFreqs*** está disponível todo o material providenciado pelos docentes da unidade curricular relativamente ao programa gerador de assinaturas e na pasta noises estão dois ficheiros de áudio que são adicionados às amostras quando executado o programa ***add_noise.py***.

2.2 Argumentos da Linha de Comandos

- **generate_freqs.py**

Nome	Flag	Descrição	Type
Input Directory	-d	Define o caminho até à coleção de músicas do dataset	String
Output Directory	-o	Define o caminho até onde as assinaturas das músicas devem ser guardadas	String
Verbose	-v	Define se o programa deve imprimir informação adicional sobre o gerar das assinaturas	Bool
Window Size	-ws	Define o tamanho da janela usada para análise de frequências	Positive Integer
Shift	-sh	Shift de uma janela para a seguinte	Positive Integer
Down Sampling	-ds	Redução da resolução do ficheiro original	Positive Integer
Number of Frequencies	-nF	Quantidade de componentes de frequência retidos por janela	Positive Integer

- **random_sample.py**

Nome	Flag	Descrição	Type
Sample Size	-s	Define o tamanho da amostra a retirar	Positive Integer
File Path	-f	Define o caminho até ao ficheiro do qual queremos tirar uma amostra	String
Output Name	-o	Define o caminho onde a nova amostra deve ser guardada	String

- **add_noise.py**

Nome	Flag	Descrição	Type
File Path	-f	Define o caminho até ao ficheiro ao qual queremos adicionar ruídos	String
Noise	-n	Define a proporção de ruído a adicionar no ficheiro de ruído constante	Float
Echo	-e	Define o atraso em milissegundos que o efeito de eco deve ter	Integer
Decay	-d	Define a proporção da amplitude do áudio original que o eco deve ter	Float

- **find_music.py**

Nome	Flag	Descrição	Type
Compressor	-c	Define o compressor e os argumentos a utilizar	String
Sample	-s	Define o caminho até à amostra que queremos identificar	String
Frequencies Folder	-f	Define o caminho até à pasta que contém as assinaturas geradas	String

3 Dataset

Para testar a implementação do modelo procurámos 30 músicas diferentes, mas de alguma maneira semelhantes aos grupos. Não tendo definido nenhum limite mínimo ou máximo de duração decidimos obter algumas músicas do mesmo autor, e por vezes do mesmo álbum, de diferentes estilos musicais e idiomas. Algumas escolhas, sendo de álbuns e autores diferentes, foram feitas tendo por base a semelhança auditiva entre elas, por exemplo as músicas exclusivamente

instrumentais ou só de piano. A lista das músicas que compõem o nosso *dataset* são as seguintes:

Música	Artista	Duração
Thunderstruck	ACDC	4:52
Bright Side of the Blue	Abby Anderson	2:57
Heart on Fire in Mexico	Abby Anderson	3:24
Senhora do Mar	anTUNiA	3:14
Thats the way it is	Daniel Lanois	4:08
Faroleiro	Dillaz	3:00
Habibi	Dillaz	3:24
Afterglow	Ed Sheeran	3:05
Nightrain	Guns N Roses	4:28
Sweet Child O Mine	Guns N Roses	5:56
Cornfield Chase	Hans Zimmer	2:07
Day One	Hans Zimmer	3:19
Os Vampiros	José Afonso	2:45
Venham Mais Cinco	José Afonso	4:40
CASTLE OF GLASS	Linkin Park	3:25
Numb	Linkin Park	3:07
The Night We Met	Lord Huron	3:28
Can You Hear The Music	Ludwig Göransson	1:50
Belong Together	Mark Ambor	2:28
Good To Be	Mark Ambor	2:27
HOPE	NF	4:24
RUNNING	NF	4:13
The Search	NF	4:08
When I Grow Up	NF	3:16
D-Day	Rui Massena	2:49
Estrada	Rui Massena	2:35
Could Have Been Me	The Struts	3:07
Música dAvó	Três bairros	4:14
Voltar Atrás	Tuna de Medicina da Universidade de Coimbra	4:35
Canção de Engate	Tuna Universitária da Madeira	4:47

4 Implementação

Como já referido anteriormente o objetivo deste modelo é identificar a que música pertence um determinado excerto utilizando a compressão das assinaturas e a NCD para ordenar todas as músicas do *dataset* por grau de similaridade. Para realizar a compressão das assinaturas geradas pelo ***GetMaxFreqs*** iremos utilizar compressores conhecidos como o gzip e o bzip2.

Após gerar a assinatura de todas as músicas do nosso conjunto de dados com o ***generate_freqs.py*** executamos o programa ***find_music.py*** para comparar o nosso excerto ou amostra de música com as existentes e ordená-las assim por similaridade entre assinaturas. Novamente, a amostra pode ser gerada com o ***random_sample.py*** e pode ser alterada com o ***add_noise.py***. Comparadas a assinatura da amostra com as assinaturas com as assinaturas das músicas completas, podemos ordená-las pelo valor da NCD e concluir que a que tem um menor valor de NCD é a que o modelo considera mais semelhante ao excerto que lhe fornecemos.

4.1 Compressores Utilizados

Para determinar o tamanho da compressão das assinaturas dos áudios em análise optamos por utilizar compressores já desenvolvidos e experimentados, como sugerido, nomeadamente o gzip, bzip2, zstd, xz, lzma e lzop. Todos estes compressores têm a opção de ser executados em modos mais rápidos e em modos mais eficientes, sendo que na maioria dos para o fazer adicionamos uma flag “-1” para compressões mais rápidas e incrementamo-la de 1 em 1 até 9 para obter modos de compressão cada vez mais eficientes.

Em termos práticos o programa irá comprimir cada um dos ficheiros e junções de ficheiros com compressor escolhido e o tamanho da compressão a utilizar no cálculo da NCD será o tamanho que o ficheiro resultante dessa compressão ocupa.

4.2 Adição de Ruído

Com o programa ***add_noise.py*** é possível, partindo de uma amostra, gerar 5 modificações desta. São essas modificações a adição de ruído de fundo de café; a adição de ruído de fundo de cidade, sendo que ambas as modificações consistem na adição do áudio de um ficheiro com esse ruído, igualar a amplitude do ruído à da amostra e sobreposição dos dois áudios; adição de eco, cujo atraso e o decay podem ser escolhidos via linha de comando; transformação num áudio de rádio antigo em que a taxa de amostragem é reduzida para 11025 Hz e as frequências acima de 4000Hz e abaixo de 200Hz são eliminadas e ainda é adicionado um ruído constante com de baixa amplitude; e a adição de ruído constante de intensidade ajustável, igualmente, via linha de comando.

Estas 5 diferentes amostras, juntamente com a original, irão ter um papel chave na avaliação da precisão do modelo para diferentes tipos de interferências sobre o sinal original.

4.3 Cálculo do NCD e Precisão

Para ordenar as músicas quanto à sua similaridade ao excerto inserido no ***find_music.py*** é necessário calcular algum tipo de medida de similaridade. Como já referido a medida utilizada neste projeto é a NCD (Normalized Compression Distance). Para realizar o cálculo da NCD é necessário determinar o tamanho das compressões da assinatura amostra, da assinatura música e de ambas concatenadas. Este cálculo é dado pela fórmula da Fig. 1.

$$NCD(x,y) = \frac{C(x,y) - \min\{C(x),C(y)\}}{\max\{C(x),C(y)\}}$$

Figura 1 - Fórmula de cálculo da Normalized Compression Distance de uma amostra x e de uma música y

Dado pela fórmula da Fig. 1, em que x representa a assinatura da amostra e y a assinatura da música, C(x,y) corresponde ao tamanho da compressão das assinaturas concatenadas, C(x) ao tamanho da compressão da assinatura da amostra e C(y) ao tamanho da compressão da assinatura da música.

Para calcularmos a precisão do modelo implementado utilizamos o programa ***benchmark.py*** que irá gerar 10 amostras para cada ficheiro de música, criará ficheiros de ruído a partir de cada um deles e testa cada um dos ficheiros de ruído com cada um dos compressores disponíveis. Para um dataset de 30 músicas são 10800 execuções do programa ***find_music.py***. Quando o ***find_music.py*** calcula que a música original é a correta a combinação música/nº amostra/compressor/amostra é marcada positivamente. O número de hits será a quantidade de combinações marcadas positivamente e a quantidade de misses serão todas as outras. Posto isto, o cálculo da precisão é dado pela fórmula da Figura 2.

$$Precisão \approx Hits/(Hits + Misses)$$

Figura 2 - Fórmula de cálculo da quantidade da precisão do modelo

5 Resultados

Configurações

Em termos de configurações disponíveis é possível variar as configurações do compressor que utilizamos para calcular os tamanhos de compressão, as configurações do gerador de assinaturas e as configurações de alguns tipos de ruído, nomeadamente do ruído constante e do eco.

Em termos de configurações dos compressores é possível pedir que cada um seja mais rápido ou mais eficiente, sendo que quando uma destas características aumenta a outra diminui. Uma vez que para este modelo é mais relevante a capacidade de o compressor utilizar informação passada para comprimir com maior eficácia informação futura é do nosso interesse que os compressores seja as mais eficientes possíveis, já que estamos a tratar de ficheiros tão pequenos que a diferença de velocidade é residual. Para comprovar esta situação foram efetuados testes com configurações mais rápidas dos compressores, mas que acabaram por originar resultados maus ao nível da precisão, uma vez que os acertos foram

praticamente nulos, apenas começando a acertar quando a configuração do compressor estava mais perto do máximo de eficiência do que do máximo de velocidade. Posto isto concluímos que utilizar os compressores com uma configuração que não aquela que é a mais eficiente não traz vantagens à sua utilização e por isso não consideramos relevante de efetuar testes profundos como os que desenvolvemos para as configurações do gerador de assinaturas e do gerador de ruídos.

Para testar o impacto da alteração das configurações nos valores da precisão do modelo retirámos uma amostra de cada uma das 30 músicas e avaliámos a capacidade do modelo detetar corretamente cada amostra em função do ruído que lhe é aplicado para várias configurações.

	Normal Sample	Noisy Sample	Echo Sample	City Sample	Cafe Sample	Analog Sample
gzip	88.0	58.0	66.0	7.3	22.3	66.0
bzip2	50.3	27.0	21.7	3.6	4.0	19.3
xz	84.3	46.3	33.7	10.3	8.3	33.0
zstd	**99.7**	**75.0**	**95.3**	**20.0**	**40.6**	**84.0**
lzop	5.0	3.6	4.0	4.3	3.7	4.0
lzma	82.3	46.3	35.3	10.3	9.3	35.0

Tabela 1 – Precisão do modelo para cada par compressor/amostra em %
(WS: 1024, SH: 256, DS: 4, NF: 4, SS: 5, N: 0.5, E: 500, D: 0.5)

	Normal Sample	Noisy Sample	Echo Sample	City Sample	Cafe Sample	Analog Sample
gzip	91.9	66.8	75.8	9.7	25.8	73.6
bzip2	89.0	61.3	55.8	5.2	14.2	49.0
xz	89.0	59.4	50.0	14.8	12.3	49.4
zstd	**96.7**	**78.4**	**95.5**	**29.7**	**45.8**	**89.7**
lzop	5.5	5.8	4.2	5.8	7.7	5.5
lzma	92.9	62.6	55.5	17.4	12.6	51.3

Tabela 2 – Precisão do modelo para cada par compressor/amostra em %
(WS: 1024, SH: 256, DS: 4, NF: 4, SS: 10, N: 0.5, E: 500, D: 0.5)

Observando as tabelas 1 e 2 em que é variado o tamanho das amostras em teste podemos observar que para todos os tipos de amostra o compressor com maior precisão dos testados é o zstd e o pior disparado é o lzop. Naturalmente verificamos que a precisão em média aumenta com o aumento do tamanho das amostras, o que é um comportamento esperado já que com mais informação para comprimir mais fácil é estabelecer diferenças entre resultados de compressão. O compressor que parece ganhar mais com o aumento do tamanho das amostras é o bzip2 já que em alguns casos consegue mais que duplicar a precisão, apesar de continuar abaixo do zstd.

Como era expectável as melhores precisões são alcançadas para as amostras sem alterações e para as amostras com eco, uma vez que são as que conferem menores ou nenhumas alterações relevantes nas suas frequências. A precisão sobre as amostras com ruído e com efeito analógico também não varia muito já que o ruído que lhes é acrescentado é constante e não superior em termos de amplitude às amplitudes máximas da amostra original e, no caso analógico, as frequências que são filtradas não são aparentemente tantas que impeçam a identificação da música.

	Normal Sample	Noisy Sample	Echo Sample	City Sample	Cafe Sample	Analog Sample
gzip	73.3	36.6	56.6	10.0	10.0	50.0
bzip2	66.7	30.0	33.3	6.7	10.0	23.3
xz	83.3	36.7	30.0	10.0	13.3	23.3
zstd	**93.3**	**66.7**	**83.3**	**23.3**	**36.7**	**76.7**
lzop	10.0	6.7	6.7	3.3	0.0	13.3
lzma	83.3	33.3	30.0	10.0	10.0	26.7

Tabela 3 – Precisão do modelo para cada par compressor/amostra em %
(WS: 1024, SH: 256, DS: 4, NF: 5, SS: 5, N: 0.5, E: 500, D: 0.5)

	Normal Sample	Noisy Sample	Echo Sample	City Sample	Cafe Sample	Analog Sample
gzip	93.3	66.6	86.7	20.0	33.3	83.3
bzip2	20.0	13.3	10.0	3.3	3.3	10.0
xz	80.0	33.3	33.3	13.3	16.7	36.7
zstd	**96.7**	**73.3**	**96.7**	**26.7**	**26.7**	**90.0**
lzop	20.0	3.3	10.0	6.6	3.3	6.7
lzma	80.0	36.7	36.7	16.6	20.0	36.7

Tabela 4 – Precisão do modelo para cada par compressor/amostra em %
(WS: 1024, SH: 256, DS: 4, NF: 3, SS: 5, N: 0.5, E: 500, D: 0.5)

Comparando os resultados das tabelas 1, 3 e 4, onde variamos o valor do número de frequências (NF), podemos verificar um ligeiro aumento da precisão do modelo à medida que o número de frequências diminui. A diminuição do número de frequências a registar pode tornar mais objetiva a seleção de frequências, e isso pode justificar as assinaturas com mais registos de frequência serem mais facilmente confundíveis, por terem mais variedade, tirando identidade. No entanto valores de NF muito baixos, como por exemplo 1, podem originar uma igual confusão do modelo novamente pela falta de identidade, já que com menos frequências também pode ser mais fácil confundir músicas com frequências semelhantes uma vez que não existem mais para servir de critério de desempate.

	Normal Sample	Noisy Sample	Echo Sample	City Sample	Cafe Sample	Analog Sample
gzip	20.0	10.0	6.7	3.3	3.3	3.3
bzip2	76.7	66.7	40.0	3.3	6.7	26.7
xz	80.0	66.7	26.7	0	10.0	26.7
zstd	**90.0**	**70.0**	**53.3**	**13.3**	**13.3**	**36.7**
lzop	3.3	3.3	6.7	3.3	0.0	10.0
lzma	80.0	66.7	40.0	6.7	10.0	26.7

Tabela 5 – Precisão do modelo para cada par compressor/amostra em %
(WS: 1024, SH: 256, DS: 2, NF: 4, SS: 5, N: 0.5, E: 500, D: 0.5)

	Normal Sample	Noisy Sample	Echo Sample	City Sample	Cafe Sample	Analog Sample
gzip	30.0	3.3	23.3	0.0	6.7	20.0
bzip2	**93.3**	**70.0**	**80.0**	**23.3**	**13.3**	**66.7**
xz	73.3	30.0	26.7	6.7	10.0	16.7
zstd	6.7	6.7	6.7	6.7	6.7	3.3
lzop	3.3	3.3	3.3	3.3	3.3	3.3
lzma	73.3	46.7	33.3	6.7	**13.3**	23.3

Tabela 6 – Precisão do modelo para cada par compressor/amostra em %
(WS: 1024, SH: 256, DS: 1, NF: 4, SS: 5, N: 0.5, E: 500, D: 0.5)

Comparando os resultados das tabelas 1, 5 e 6, onde variamos o valor de downsampling (DS), verificamos uma alteração do compressor mais eficaz, para DS=1 ou para quando o número de amostras permanece inalterado, que passa a ser o bzip2, caindo inesperadamente a pique a precisão do zstd. No entanto os resultados de precisão obtidos pelo bzip2 para DS=1 não são melhores que os

resultados obtidos pelo zstd para DS=4, não correspondendo assim a uma melhoria da configuração do sistema.

	Normal Sample	Noisy Sample	Echo Sample	City Sample	Cafe Sample	Analog Sample
gzip	76.7	50.0	46.7	3.3	3.3	36.6
bzip2	16.7	23.3	16.7	3.3	3.3	13.3
xz	50.0	30.0	16.7	6.7	3.3	23.3
zstd	**93.3**	**63.3**	**83.3**	**23.3**	**23.3**	**73.3**
lzop	6.7	6.7	6.6	3.3	3.3	0.0
lzma	46.7	30.0	16.7	6.7	3.3	26.6

Tabela 7 – Precisão do modelo para cada par compressor/amostra em %
(WS: 512, SH: 256, DS: 4, NF: 4, SS: 5, N: 0.5, E: 500, D: 0.5)

	Normal Sample	Noisy Sample	Echo Sample	City Sample	Cafe Sample	Analog Sample
gzip	50.0	33.3	16.6	10.0	10.0	23.3
bzip2	56.7	26.7	30.0	3.3	3.3	26.7
xz	86.7	56.7	33.3	6.7	10.0	36.7
zstd	**100.0**	**80.0**	**93.3**	**26.7**	**53.3**	**86.7**
lzop	6.7	10.0	3.3	3.3	6.7	3.3
lzma	86.7	56.7	30.0	13.3	10.0	40.0

Tabela 8 – Precisão do modelo para cada par compressor/amostra em %
(WS: 2048, SH: 256, DS: 4, NF: 4, SS: 5, N: 0.5, E: 500, D: 0.5)

Variando o tamanho da janela WS vemos os resultados nas tabelas 1, 7 e 8 onde, tendo novamente o zstd com os melhores resultados, temos uma precisão de 100% para a run efetuada com WS = 2048 para amostras sem alterações. É possível verificar inclusive que o aumento da precisão está diretamente relacionado com o aumento do tamanho da janela. Uma possível justificação para este comportamento pode explicar-se da mesma maneira que se explicou o aumento da precisão estar relacionado com a diminuição de NF. Com o aumento do tamanho da janela WS e a manutenção de NF seria semelhante a manter WS e diminuir NF, logo seria possível construir uma identidade mais vincada de cada música.

Numa tentativa de conjugar fatores que alterados singularmente resultaram num aumento da precisão do modelo a tabela 9 apresentada a seguir é o resultado de um aumento da janela WS acompanhado do aumento do tamanho das amostras.

	Normal Sample	Noisy Sample	Echo Sample	City Sample	Cafe Sample	Analog Sample
gzip	96.7	90.0	76.7	33.3	46.7	80.0
bzip2	**100.0**	83.3	56.7	6.7	23.3	50.0
xz	**100.0**	**96.7**	46.6	20.0	30.0	46.7
zstd	**100.0**	**96.7**	**96.7**	**50.0**	**63.3**	**93.3**
lzop	16.7	13.3	10.0	10.0	6.7	3.3
lzma	**100.0**	90.0	53.3	23.3	36.7	50.0

Tabela 9 – Precisão do modelo para cada par compressor/amostra em %
(WS: 2048, SH: 256, DS: 4, NF: 4, SS: 10, N: 0.5, E: 500, D: 0.5)

Como esperado o aumento do tamanho das amostras resultou num aumento da precisão do modelo, tendo inclusive atingido 100% de precisão para 4 compressores para amostras não adulteradas. Também podemos ver que com esta configuração alcançamos o primeiro resultado minimamente interessante para a precisão na avaliação de amostras com ruído de cidade.

Numa última tentativa de melhoramento dos resultados de precisão reduzimos NF uma vez que já tínhamos verificado que essa diminuição tinha ajudado a aumentar os valores de precisão do modelo.

	Normal Sample	Noisy Sample	Echo Sample	City Sample	Cafe Sample	Analog Sample
gzip	93.5	80.0	83.9	**29.0**	29.0	77.4
bzip2	74.2	58.1	32.3	6.5	12.9	29.0
xz	90.3	74.2	45.1	**29.0**	29.0	51.6
zstd	**93.5**	87.1	**90.3**	**29.0**	**35.4**	**87.1**
lzop	16.1	12.9	13.0	6.5	6.5	9.7
lzma	**93.5**	**90.3**	71.0	25.8	29.0	61.3

Tabela 10 – Precisão do modelo para cada par compressor/amostra em %
(WS: 2048, SH: 256, DS: 4, NF: 3, SS: 10, N: 0.5, E: 500, D: 0.5)

Como teorizado anteriormente a descida de NF para 3 com WS a 2048 resultou numa perda de identidade por parte de algumas assinaturas tendo diminuído no geral a precisão do modelo para praticamente todos os compressores. Apesar de não ser uma queda abrupta como já verificado anteriormente é o suficiente para concluir que já ultrapassámos o NF ideal ao descê-lo para 3.

Relativamente às configurações das amostras de ruído podemos verificar, sem uma análise detalhada como as anteriores e como já era expectável, que o aumento da intensidade do ruído ou o aumento do desfasamento do eco e o aumento da sua intensidade contribuem para a distorção do áudio original uma vez que acabam por sobrepor e substituir frequências chave na identificação de uma música.

6 Conclusão

Os resultados obtidos foram no geral interessantes uma vez que conseguimos valores de precisão relativamente bons tendo em conta a abordagem simplificada que foi feita ao problema em questão. Obviamente seria bastante interessante colocar o programa à prova num contexto de mundo real para perceber se a eficácia se mantém para outros áudios, até mesmo gravados com o telemóvel, para perceber se os testes que realizámos ao modelo representam bem as dificuldades que é uma gravação banal de microfone ou se mesmo aplicando diferentes tipos de ruído digitalmente não conseguimos simular um ambiente real.

Os resultados acabam por permitir-nos concluir que a extração das frequências certas e a amplitude do ruído presente no áudio são fatores determinantes na capacidade deste modelo especifico conseguir ou não identificar a música em questão. Também observámos que quanto maior for a amostra mais facilmente conseguimos identificar qual a música a que pertence uma vez que iremos ter mais contexto e os compressores serão capazes de provocar variações no NCD maiores.

Referências

[1] A. J. Pinho e D. Pratas, “GetMaxFreqs.zip,” [Online]. Available: https://elearning.ua.pt/mod/resource/view.php?id=290156. [Acedido em Maio 2024].

[2] A. J. Pinho e D. Pratas, “trab3.pdf,” [Online]. Available: https://elearning.ua.pt/pluginfile.php/376169/mod_resource/content/7/trab3.pdf. [Acedido em Maio 2024].