Music Genre Classification using CNN

A. Basile, A. R. Iacovazzi

November 2022 - University of Bari Aldo Moro

Abstract

The goal of this project is to use the deep learning approach to make classifications in relation to music genres and to make, the use of the final model, immediately available through a telegram bot that integrates it. CNNs are very powerful neural networks that, through convolution operators, are able to automatically extract useful features to make a distinction between observations, in this case, music tracks. The network was initially trained by extracting various types of features but in the end, it was decided to use the MFCCs of the songs (Mel-frequency cepstral coefficients) as they are more informative than the others. MFCCs are time-domain-based windows that collect a set of components (usually between 10 and 20) that can concisely describe the overall shape of a spectral envelope, thus, those areas of the spectrum with the highest energy. These features are extracted in various ways, i.e., by taking 13 and 15 components, and for both of them MFCCs (overlapped and not-overlapped) of 2, 3, 4 and 5 seconds are collected so as to explore which settings are most meaningful for solving this task. The obtained model is, finally, used to predict multiple time windows collected from a single track and based on them make a majority vote that will determine the most plausible class.

1 Introduction

The classification of musical genres is a complex task for humans for several reasons. First of all, a song almost never belongs distinctly to a specific genre but, usually, various shades of other genres are present. In addition, many genres are similar to each other such as rock and metal, consequently it is easy to confuse genres unless one knows the artist or musical group that usually come pre-labeled by humans as belonging to one genre for the sake of simplicity. However, despite the presence of sample-poor datasets, with this project it is interesting to show that CNN is able to learn how to distinguish between genres based on purely timbral characteristics of the songs. It should be premised on the fact that labeling songs to a particular genre is often a very subjective human practice, and thus this may spill over into interpretation by CNN. In fact, it will be possible to see that various interesting misclassifications of our network make sense since, some songs, despite labeled with a particular genre, tend to have, characteristics of other genres. The first experiments were carried out in order to conduct feature explorations by testing Chroma, Harmonica and MFCC individually relatively to Classical and Disco genres. The first two yielded very low results, so they were discarded immediately because they were not informative for the type of task, whereas, MFCC is very often used for speech recognition tasks in state-of-the-art projects, as it is capable of distinguishing timbral information. In this regard, we performed experiments that, in this case, yielded relatively good initial results. After coming to this conclusion, it was decided to deepen and improve the approach by integrating different types of feature extraction, modifying the structure of the model, and changing the classification system.

2 Related Works

Modern streaming services has increased a lot the demand for automatic music information retrivial such as the music genre recognition. Lots of works has been done in the past two decades in this domain using several tools that progressively the state of the art in machine lerarning has offered. One of the first seminal works was the one of Tzanetakis and Cook in 2002 [1] that proposes a diversified set of features for direct modeling of music signals and explores the use of those features for musical genre classification using K-Nearest Neighbors and Gaussian Mixture models. A similar approach was proposed also by Burred e Lerch in 2003 [2] using instead a hierarchical approach for the classification. More recently researchers have investigated the capabilities of deep learning techniques like convolutional neural network in the works of Zhang et al in 2016 [3] or Yang et al in 2020 [4] that have proposed a model where cnn and recurrent network are used in parallel. Even further, transformers model has been proposed like MusicBERT by Zeng et al [5] that is capable to cover several tasks in the MIR domain, included the genre classification.

3 Data-set Description

The dataset used in this project is the GTZAN dataset[6] which presents one hundred 30-second songs for 10 genres. This dataset is the most-used public dataset for evaluation in machine listening research for music genre recognition (MGR). The files were collected in 2000-2001 from a variety of sources including personal CDs, radio, microphone recordings, in order to represent a variety of recording conditions. Furthermore, the dataset was a team project done by three university students and not by musics experts then it is necessary to take into account also this aspect.

Music genres in the dataset are:

- Blues
- Classical
- Country
- Disco
- Hiphop
- Jazz
- Metal
- Pop
- Reggae
- Rock

This dataset was downloaded +29k times and is also quite famous in kaggle. s mentioned before, it has various flaws, motivations that where also treated in an interesting discussion present on [7] in which additional warnings are highlighted such as:

- The audio quality varies by samples (though it was intended) and it is not annotated
- There are heavy artist repetition, which are very often ignored during dataset split
- The labels don't seem to be 100% correct

4 Proposed method

4.1 Preprocessing

The preprocessing stage is critical if the goal is to obtain a model that can generalize predictions and focus only on the features fundamental to making distinctions between musical genres. For this purpose, two different pipelines have been realized. Before starting with the different pipelines, the dataset is splitted in train and validation set with a proportion respectively of 70% and 30%.

4.1.1 Pipeline one

In this pipeline every track is splitted in a 3 seconds long fragments and each fragment is detached from the previous one (without overlapping). With this approach is possible to generate $samples = l_s/l_w$. Every fragment is transformed applying two effects:

- Pitch-shift: the shifting is applied choosing a random value between 4 and 6. These values indicate how many seminotes are added to the original pitch.
- White-noise: add a background noise scaled by a factor of 0.005.

Subsequently, the MFCC is computed for all these fragments and the output is saved in a .json file using a fixed number of component of 13.

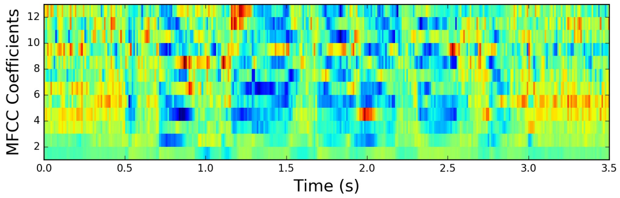


Figure 1: An example of MFCC

4.1.2 Pipeline two

Only one effect is applied in order to make songs different each other. More precisely, the effects applied are:

- Pitch-shift: is applied three times using as semitones shift the values 2, 4 and 6.
- Time-stretch: is applied two times using two values 0.5, that double the length of the track, and 2, that halve the length of the track.
- White-noise: add a background noise scaled by a factor of 0.005.

Each transformation generate a whole new track. So, every track exists in this augmented dataset in 7 different versions[1]. After these phase, the extraction of MFCCs is performed using a 3-seconds overlapped windows and a number of components equal to 13. The overlapping is made in the way that one MFCC overlaps the other by 50%. Example: if a window captures a 3-second interval then each window overlaps the previous one every 1.5 seconds. With this approach is possible to generate $samples = l_s/(l_w/2)$. The two models that have been produced by these two pipelines gave different results in term of accuracy using the CNN architecture explained in the section 4.2

[1] The 6 transformed versions plus the original one.

"CNN Architecture". The first one gave an accuracy score of 67% while the second one gave an accuracy around 72%. These results make evident that the second pipeline is better than the first one. To further investigates on the potential of the second preprocessing pipeline, several experiments have been conducted using different settings for time-length window and the number of MFCC components that are:

- Time-length window: 2, 3, 4 and 5 seconds length.
- Number of MFCC components: 13 and 15 components.

At the end, 8 different .json files have been produced using all the combination of the previous seen parameters.

4.2 CNN Architecture

The Neural Network used in this project is a Convolutional one. Convolutional Neural Network are particular network which are based on Convolutional Operators called Kernels. "Convolutional Neural Network has had ground breaking results over the past decade in a variety of fields related to pattern recognition; from image processing to voice recognition. The most beneficial aspect of CNNs is reducing the number of parameters in ANN [...] Another important aspect of CNN, is to obtain abstract features when input propagates toward the deeper layers" [3].

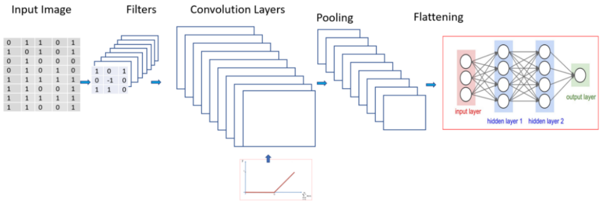
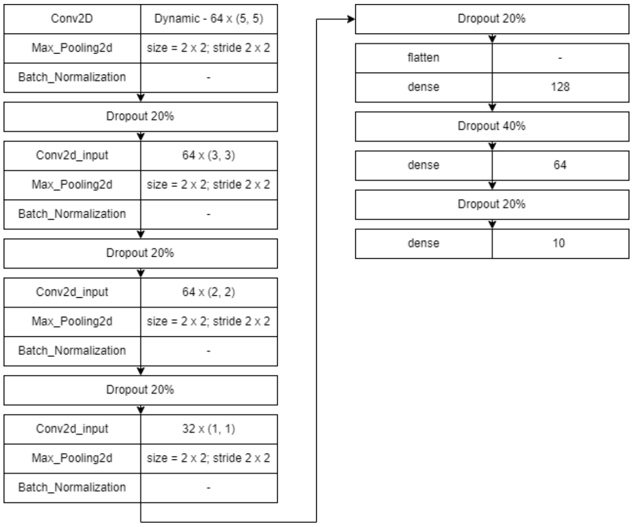


Figure 2: Classic Structure of a CNN

This kind of NN are largely used on images to find pattern with the goal to distinguish certain classes. The peculiarity of this work is that, the convolutional neural network is not applied to images but time windows concerning audio timbre. The intuition was to imagine audio not as a signal but as something visual related to it. The overall structure of our model is composed by four convolutional layer containing:

- 2D Convolution Operator Layer using ReLu activation function
- 2D MaxPooling Layer
- Batch Normalization Layer

each of this Convolutional Layer is followed by a Dropout Layer used as regularizer. At the end, the output of these layers is flatten and sent to a dense layer that will carry out the final decision using an softmax activation function taking into account the number of classes to predict.



Conv2D	Dynamic - 64 x (5, 5)
Max_Pooling2d	size = 2 x 2; stride 2 x 2
Batch_Normalization	-
Dropout 20%	
Conv2d_input	64 x (3, 3)
Max_Pooling2d	size = 2 x 2; stride 2 x 2
Batch_Normalization	-
Dropout 20%	
Conv2d_input	64 x (2, 2)
Max_Pooling2d	size = 2 x 2; stride 2 x 2
Batch_Normalization	-
Dropout 20%	
Conv2d_input	32 x (1, 1)
Max_Pooling2d	size = 2 x 2; stride 2 x 2
Batch_Normalization	-
Dropout 20%	
flatten	-
dense	128
Dropout 40%	
dense	64
Dropout 20%	
dense	10

Figure 3: The CNN structure used in this project

Dynamic means that the input shape of Conv2D is adapted to the given MFCC. The parameters used in the layers have been chosen after a lot of experiments in order to find the best ones that was light in term of parameters number without sacrificing the accuracy that are:

- First three 2D Convolution Operator Layer with 64 parameters respectively with filter size of 5×5, 3×3 and 2×2.
- Last one 2D Convolution Operator Layer with 32 parameters with filter size of 1×1
- All 2D MaxPooling Layer using 2×2 kernel and 2×2 strides.
- First fours Dropout Layers with 0.2 factor.
- Last one Dropout Layer with 0.4 factor.
- First two Dense Layer respectively with 128 and 62 parameters with ReLu as activation functions.
- Last one Dense Layer with 10 parameters and Softmax activation function.

4.3 Training Phase

The model was trained with the following settings for the hyper-parameters:

- Optimizer: Adam with learning rate = 0.0001
- Loss Function: Sparse Categorical Crossentropy

- Number of epochs: 300
- Batch size: 32
- Early stopping: patience = 20 and restore best weights set to True

These hyper-parameters have been tuned empirically within a great number of experiments to reach the best observable trade-off between time-complexity and accuracy. The metric used to apply the early stopping is the validation loss. The big number of epochs is due to the necessity of have enough epochs to allows the early stopping to apply; the biggest number of epochs seen in actual training never exceed the number of 110. Furthermore, the training is executed several times taking into account different combinations of MFCC components and seconds for the MFCC windows as explained in section 4.1.2.

5 Experimental results

5.1 Results on the validation set

All the experiment was done using a Google Colab environment with GPU acceleration and 25 GB of RAM. The following table resume the results obtained on the validation set with the different feature configuration.

N. MFCC	Window Length	Accuracy	F1-Score	Learning Time
13	2	0.72	0.73	∼ 40 min
13	3	0.72	0.73	∼ 20 min
13	4	0.75	0.75	∼ 21 min
13	5	0.70	0.70	∼ 9 min
15	2	0.71	0.71	∼ 26 min
15	3	0.72	0.72	∼ 17 min
15	4	0.74	0.74	∼ 14 min
15	5	0.73	0.73	∼ 13 min

Table 1: Results on prediction of slice of tracks

As stated by the caption these results are about the prediction of the single MFCCs slices derived by the preprocessing pipeline. However, the aim of this classifier is to predict the genre of a whole track. In order to have a reliable prediction of a track its necessary to detect the most representative class predicted taking into account all the sliced derived by the preproccessing. Two different approach has been used that are:

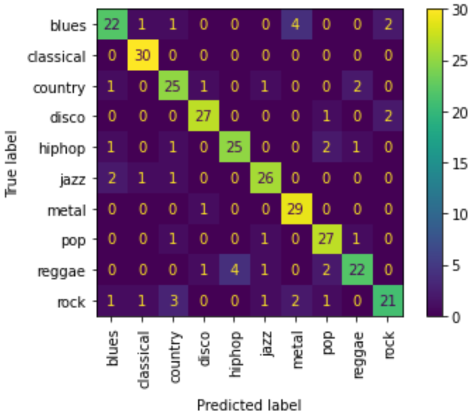
- Vote: take the most common prediction.
- Max: summing all the prediction and the taking the classes corresponding to the biggest value.

The following table resume the results obtained using this two technique to derive the class of the whole track.

N. MFCC	Window Length	Aggregation type	Accuracy	F1-Score
13	2	Vote	0.85	0.84
13	2	Max	0.85	0.84
13	3	Vote	0.81	0.81
13	3	Max	0.81	0.81
13	4	Vote	0.81	0.80
13	4	Max	0.82	0.82
13	5	Vote	0.77	0.77
13	5	Max	0.77	0.77
15	2	Vote	0.80	0.80
15	2	Max	0.82	0.82
15	3	Vote	0.80	0.80
15	3	Max	0.81	0.81
15	4	Vote	0.82	0.82
15	4	Max	0.82	0.82
15	5	Vote	0.82	0.82
15	5	Max	0.81	0.81

Table 2: Results on prediction of the whole tracks

The results suggests that despite the worse performance on the singles slices, the configuration with 13 MFCCs component and 2 seconds length time window performs better, this probably because is the configuration with more slices involved so the actual class has most chances to appear and obtain the majority of votes. Regarding the aggregation algorithm there are no sufficient evidence that clearly suggest that one is better than the other since the results are similar in almost all the cases.



	blues	classical	country	disco	hiphop	jazz	metal	pop	reggae	rock
blues	22	1	1	0	0	0	4	0	0	2
classical	0	30	0	0	0	0	0	0	0	0
country	1	0	25	1	0	1	0	0	2	0
disco	0	0	0	27	0	0	0	1	0	2
hiphop	1	0	1	0	25	0	0	2	1	0
jazz	2	1	1	0	0	26	0	0	0	0
metal	0	0	0	1	0	0	29	0	0	0
pop	0	0	1	0	0	1	0	27	1	0
reggae	0	0	0	1	4	1	0	2	22	0
rock	1	1	3	0	0	1	2	1	0	21

Figure 4: Confusion matrix for the classifications of the whole tracks using the best combination of feature parameters (13,2) and the aggregation type "vote".

Genre	Precision	Recall	F1-Score
blues	0.81	0.73	0.77
classical	0.91	1.00	0.95
country	0.78	0.83	0.81
disco	0.90	0.90	0.90
hiphop	0.86	0.83	0.85
jazz	0.87	0.87	0.87
metal	0.83	0.97	0.89
pop	0.82	0.90	0.86
reggae	0.85	0.73	0.79
rock	0.84	0.70	0.76
macro avg	0.85	0.85	0.84
accuracy	0.85		

Table 3: Details about the results for the classifications of the whole tracks using the best combination of feature parameters (13,2) and the aggregation type "vote".

From the confusion matrix is possible to see which kind of genres are less recognized, it's possible to see that blues, reggae and rock are the most difficult to classify, in particular rock is the genre that has the worst recall. A valid hypothesis about this behaviour is that the rock genre covers a lot of different variation in terms of timbre and style that could be easily confused with other genres even by humans.

5.2 Results on the test set

The aim of this work is to test the capabilities of the model to generalize the concept of music genre in the most wide way, in order to test this capability the tests performed on the validation set are not sufficient since this set was used for choosing hyperparameters, those results could heavily depends on the similarities between the tracks since the approach chosen to collect the dataset is not known and could be biased in same way. Furthermore, as stated before in Section 3, it contains only quite popular tracks until the year of publication (2001) so certain genres that has has evolved a lot during time, or that has several sub-genres, could be not recognized. In order to test the capabilities of the model on unknown tracks, we have collected several playlists from YouTube and have extracted from them 30 tracks of 30 second for each of the genres involved paying also attention to include recent songs and some variation of the same genres. This test has also the aim of analyze the quality of the model in the real life usage, where songs of every kind will be given as input. On this new dataset only the model resulted with the best performance in predicting whole tracks on the validation set has been tested. Before performing the test, the model has been re-trained on the full augmented dataset following the second preprocessing pipeline and using the feature parameters that maximize the accuracy on the full track genre prediction. Since here there is no validation set available to apply the early stopping, a proportional number of epoch was chosen[2]. After that the prediction on the test dataset has been performed obtaining the following results.

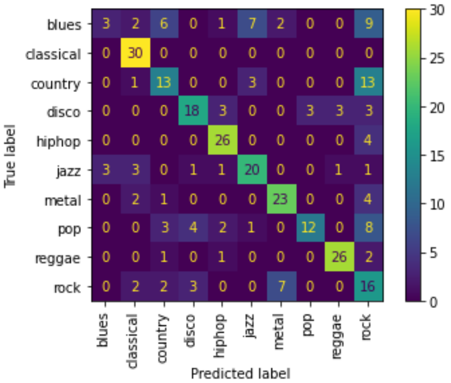


Figure 5: Confusion matrix for the classifications of the whole tracks of the test set.

Genre	Precision	Recall	F1-Score
blues	0.50	0.10	0.17
classical	0.75	1.00	0.86
country	0.50	0.43	0.46
disco	0.69	0.60	0.64
hiphop	0.76	0.87	0.81
jazz	0.65	0.67	0.66
metal	0.72	0.77	0.74
pop	0.80	0.40	0.53
reggae	0.87	0.87	0.87
rock	0.27	0.53	0.36
macro avg	0.65	0.62	0.61
accuracy	0.62		

Table 4: Details about the results for the classifications of the whole tracks of the test set.

On these new tracks the results are decisely more low than the ones obtained on the validation set, in fact it ends up with an accuracy of 0.62. The value, computed used a total of 300 samples is in a confidence interval of $[0.56, 0.67]$ with a 95% probability. This results are not so surprising, the GTZAN dataset

[2] Since the training with the 70% of the dataset took 90 epoch (110 - 20 of early-stopping patience), for the 100% of data a proportional value of 130 epoch was thought as sufficient.

contains too few tracks to have a meaningful representation of the timbral variation of complex genre like pop or rock. Quite interesting is the case of the blues genre that is systematically confused as country, jazz or rock, genres that share several instruments with the blues. Probably this is due to the fact that rely only on the timbral and rhythmical information is not sufficient when several genres share the same instruments.

5.3 Analysis of the mistakes

Before concluding would be interesting try to understand the origin of the misclassification. For this reason every misclassified track has been given in output with the respective predicted genre, this in order to see if there is a kind of pattern in such errors.

- The misclassification of a track as blues doesn't seem to have a particular pattern.

- The misclassification of a track as classical happen when the track has no drum part or when is quite ethereal.

- The misclassification of a track as country happen often when an acoustic guitar or an high pitched voice is involved.

- The misclassification of a track as disco happen often when the track has 4/4 bassdrum or the bass has a predominant role in the rhythmic section.

- The misclassification of a track as hiphop happen often when the track has a vocal part that is not melodic or quite monotonous, this probably because resemble the rap vocals.

- The misclassification of a track as jazz happen often when a track is slow and with a predominant bass part, this could explain why several blues track has been misclassified as jazz.

- The misclassification of a track as metal happen when it's loud and has a distorted guitar or when presents an high pitched solo or voice. This explain why several rock songs has been misclassified as metal.

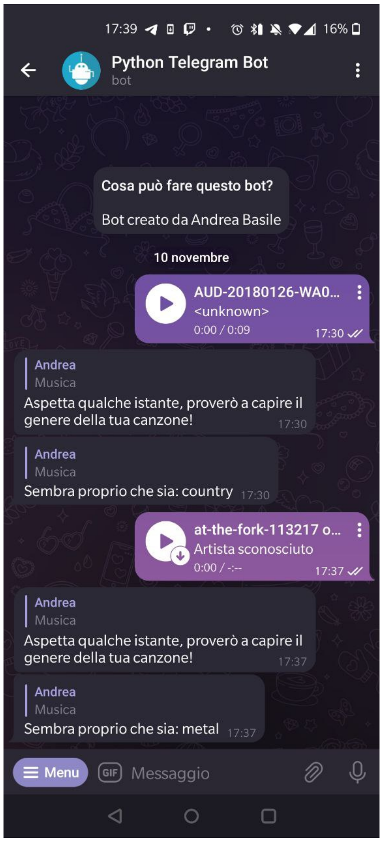
- The misclassification of a track as pop happens only three times on disco songs with female singers, is quite typical that pop songs has a 4/4 beat and a female singer but there is no sufficient example to state that was a pattern of misclassification.

- The misclassification of a track as reggae happens only four times on track with a rhythmic element in upbeat or a distinguishable sound of snare rim, quite common in reggae, however there is no sufficient example to state that was a patter of misclassification.

- The misclassification of a track as rock is the most common one, it happens often with blues and country tracks that have a lot of similarities with rock tracks and probably would be confusing also for humans. However also other misclassification as rock tracks have occurred for tracks that simply present a solid 4/4 beats and little else that would resemble a rock track as in the case of the four hiphop tracks misclassified as rock. So the kind of rhythm could have a role in this kind of errors.

6 Telegram Bot using CNN

In order to make the model available to everyone who want to try it, we decided to build up a telegram bot using python. This bot can handle file audio that last at least 2 seconds and can predict the class using the final model that is the one with the best accuracy score.



Cosa può fare questo bot?

Bot creato da Andrea Basile

10 novembre

AUD-20180126-WA0...
<unknown>
0:00 / 0:09 17:30

Andrea
Musica
Aspetta qualche istante, proverò a capire il genere della tua canzone! 17:30

Andrea
Musica
Sembra proprio che sia: country 17:30

at-the-fork-113217 o...
Artista sconosciuto
0:00 / -:-- 17:37

Andrea
Musica
Aspetta qualche istante, proverò a capire il genere della tua canzone! 17:37

Andrea
Musica
Sembra proprio che sia: metal 17:37

Figure 6: Telegram Bot predicting some songs

Note that the Telegram Bot is not hosted in any server since it was build only for didactic purpose. Furthermore is essential that the audio sent respect the following rules:

- The file is a .wav file.
- The file last at least 2 seconds.
- The file size is not over 15 MB.

7 Conclusions and future works

As results suggest, the MFCCs is informative to understand the genre but it's not sufficient to obtain very accurate predictions for all the genres. For this reason, it would be necessary introduce other kind of features that convey information about harmony or melody. Furthermore, the dataset is still too small for the task especially if the model has to predict nowadays songs that are really shaded by other genres. Would be also interesting try to distinguish genres in a hierarchical way that would also helps to differentiate even between similar genres. However this approach need a serious musicological investigation. The last aspect that can change the whole game could be using different models as more deep and complex CNN, RNN or even Transformers.

References

[1] Tzanetakis Geoge; Cook Perry. "Musical genre classification of audio signals". In: (2002).

[2] Burred Juan José ; Lerch Alexander. "A Hierarchical Approach To Automatic Musical Genre Classification". In: (2003).

[3] Zhang Weibin; Lei Wenkang; Xu Xiangmin; Xing Xiaofeng. "Improved Music Genre Classification with Convolutional Neural Networks". In: (2016).

[4] Yang Rui; Feng Lin; Wang Huibing; Yao Jianing; Luo Sen. "Parallel Recurrent Convolutional Neural Networks-Based Music Genre Classification Method for Mobile Devices". In: (2020).

[5] Zeng Mingliang; TanXu; Wang Rui; Ju Zeqian; Qin Tao; Liu Tie-Yan. "MusicBERT: Symbolic Music Understanding with Large-Scale Pre-Training". In: (2021).

[6] Andrada Olteanu. *GTZAN Dataset - Music Genre Classification.* 2019. URL: `https://www.kaggle.com/datasets/andradaolteanu/gtzan-dataset-music-genre-classification`.

[7] Keunwoo Choi Minz Won Janne Spijkervet. *Music Classification: Beyond Supervised Learning, Towards Real-world Applications.* 2021. URL: `https://music-classification.github.io/tutorial/part2_basics/dataset.html`.

[8] Saad Albawi, Tareq Abed Mohammed, and Saad Al-Zawi. "Understanding of a convolutional neural network". In: *2017 International Conference on Engineering and Technology (ICET).* 2017, pp. 1–6. DOI: `10.1109/ICEngTechnol.2017.8308186`.