Evaluating Music Mastering Quality Using Machine Learning

Mark Shtern
York University
4700 Keele Street
Toronto, ON
mark@cse.yorku.ca

Pedro Casas
York University
4700 Keele Street
Toronto, ON
pjcasas2@my.yorku.ca

Vassilios Tzerpos
York University
4700 Keele Street
Toronto, ON
bil@cse.yorku.ca

ABSTRACT

Machine learning has been applied in a vast array of applications in the recent years, including several qualitative problems in the arts. However, in the world of music production, including mixing and mastering, most tasks are still performed by music professionals with decades of experience. Aspiring mastering engineers typically have to apprentice with professionals to learn their craft. Access to professionals is a scarce resource though, as they are typically very busy.

In this paper, we present a method to evaluate the mastering quality of a piece of music automatically. We delegate the task of determining what we deem to be a subjectively well mastered song to professional mastering engineers. Using professionally mastered music, we derive datasets with varying degrees of deviation from the original music and train models to recognize the changes that have been made. This allows us to provide novice mastering engineers with an automatic rating of their work based on the magnitude of the deviation from the gold standard. We present experiments that demonstrate the accuracy of our approach, as well as a user study that shows how the results of our approach correlate to assessments made by human evaluators.

CCS CONCEPTS

• Computing methodologies → Supervised learning; **• Applied computing → Sound and music computing**;

KEYWORDS

machine learning, computational creativity, computer music

ACM Reference format:
Mark Shtern, Pedro Casas, and Vassilios Tzerpos. 2018. Evaluating Music Mastering Quality Using Machine Learning. In *Proceedings of CASCON, Markham, Ontario, Canada, October 2018 (CASCON '18),* 10 pages.
https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

In music production, the mastering of audio is the final step in the development of a song or piece of music which serves to make the audio sound as professional as possible. It is the mastering engineer who must tastefully edit a single sound file which may consist of many instruments, or very few instruments, or even just a voice depending on the music genre and song.

One of the main tools used by the mastering engineer is an equalizer. An equalizer changes an audio file by amplifying or reducing the volume of certain frequencies. A car typically has an equalizer that lets you change the amount of "bass" and "treble", which typically corresponds to affecting two large bands of frequencies centered at 250Hz and 4000Hz respectively. A mastering engineer performs similar equalization operations but has the ability, through the use of various parameters, to modify the center frequency, the width of the band around the center frequency, as well as the amount of amplification or reduction.

A mastering engineer's goal is two-fold when equalizing a song. First, she needs to ensure that all frequencies in the human hearing range (20Hz to 20kHz) are clearly audible, especially frequencies that correspond to the instrumentation in the music. For this purpose, she may accentuate frequencies that will bring out certain qualities, such as the attack of a snare drum. Second, she needs to ensure that the music can be enjoyed in a variety of sound systems and listening environments. For example, the noise of a car on the highway masks certain frequencies that may need to be amplified a bit. These two goals are often at odds with each other. Professional mastering engineers go through rigorous ear training and rely on years of experience to achieve the perfect balance for their final product.

The goal of the work presented in this paper is to develop approaches that can help aspiring mastering engineers learn from the craft of professional mastering engineers. For this purpose, we consider officially released music as the gold standard. We then apply different sets of equalization operations on the gold standard that we can use as input to machine learning algorithms to train models that can recognize the equalization operations that are needed to transform one piece of music to another. This way a mastering engineer in training can receive feedback on how many equalization operations separates their work from that of a professional.

This work was inspired by a similar approach in the image processing domain [5]. In that work, the authors start with professionally prepared photographs, apply various modifications to them, and train a model that can recognize these modifications. When these modifications are used to process new photographs, the results can in some cases be confused with professional work.

In our case, the goals of our work are two-fold:

(1) Develop a machine learning approach that can successfully recognize the equalization operations that it would take to transform one piece of audio to another.
(2) Develop a training tool for mastering engineers that can rate their work against a gold standard, as well as suggest equalization operations to improve it.

The structure of the remainder of this paper is as follows: Section 2 presents the necessary background on mastering, equalization, as well as the data representations of audio we used in our work. Section 3 describes a pilot study that informed the rest of our work. Our methodology is presented in Section 4. Section 5 presents the results of our experiments. A user study we conducted to establish the usefulness of our work is presented in Section 6. Related work is presented in Section 7. Finally, Section 8 concludes the paper and presents opportunities for further research.

2 BACKGROUND

Mastering is the last step in the process of preparing a song for public release[11]. This process starts with recording all instruments and vocals separately in individual audio files. These files are then typically edited to correct any errors, e.g. autotuning vocals. Next, all individual audio files are combined together to create one audio file during a process called mixing. Put simply, in this step the tracks are processed and their volume is adjusted so the tracks all sound like one cohesive song when put together. The mixed song is then sent to a mastering engineer.

During the mastering process, the audio can be equalized, its dynamic range compressed, and its volume adjusted for it to sound consistent across different sound systems as well as to artistically make the song sound more pleasing. Other processing, such as adding reverberation, may be performed for artistic reasons. The end result of the mastering process is referred to as the *master* of the song.

In this paper, we focus on the equalization part of the mastering process, as it is a technical rather than an artistic matter, and it is usually the hardest one to get right for aspiring mastering engineers. For this purpose, we introduce the concept of an *equalization operation*, defined in detail below.

Equalizing an electronic signal refers to balancing different frequency ranges in the signal. This can mean increasing or decreasing the energy of the signal in a certain frequency band. Musically, when speaking about equalizing a song the terms *boosting* and *cutting* volume are used instead of increasing or decreasing energy, respectively. Finding a good balance when equalizing involves finding the right amount of volume for each frequency band, so that the song can be perceived in a pleasing fashion in a variety of sound systems, e.g. both in high end stereo systems as well as on laptop speakers, as well as in a variety of environments, e.g. in a quiet room as well as in a car on the highway.

An equalization operation has three distinct parameters:

(1) Frequency. This is the center frequency that will be boosted or cut. Frequencies close to the center frequency will also be affected to a lesser degree as described below.
(2) Gain. The amount of boosting or cutting for the center frequency, measured in decibels, positive for boosting, negative for cutting.
(3) Width. This parameter designates the range of frequencies that will be affected by the equalization operation. In music production, this parameter is usually referred to as Q. Small values of Q correspond to wider equalization operations and vice-versa. In this paper, we measure width in the number of octaves[1] that span the frequencies that are affected by the equalization operation.

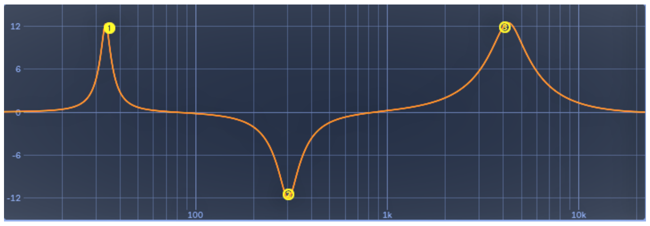

Figure 1: Equalization curve showing a 12dB boost at 33Hz with a half octave width, a 12dB cut at 300Hz with width of one octave, and a 12dB boost at 4.2kHz with a width of two octaves

Figure 1 shows a combination of three equalization operations with different center frequencies and different widths. As can be seen in Figure 1, the range of frequencies affected by an equalization operation follows a Gaussian distribution, i.e. the further away from the center frequency, the smaller the amount of boosting or cutting.

Boosting equalizations by definition increase the volume of a particular set of frequencies. That means that the clip overall becomes a little bit louder. When performing multiple equalizations on the same clip, this increase in volume can be significant, to the point where clipping occurs. To avoid this, all clips in our datasets were normalized to -12dBFS prior to applying any equalization on them. This ensured that none of the equalized clips were distorted due to clipping.

In the case of music mastering, it is rare to come across extreme equalization operations, such as ones using narrow widths or large amounts of gain, as such operations would affect the sound quite dramatically. Mastering engineers typically perform wide equalizations with a gain that is never more than 6dB. It is also unlikely that they would apply multiple equalizations whose center frequencies would be very close to each other, typically they would be at least an octave apart. We refer to equalization operations that conform to the description in this paragraph as *mastering-relevant equalizations*.

To achieve the goals outlined in the introduction, it is necessary to be able to determine what is the difference between two different masters of the same song, e.g. a master produced by a music professional and one produced by a novice mastering engineer. This way, the novice could study the equalization operations that would allow them to transform their master into the one produced by a professional.

[1]While a musical term, *octave* has a distinct meaning in the frequency domain. An octave is the distance between a frequency f and frequency $2f$. As a result, a linear increase in octaves is an exponential increase in frequency. For this reason, all graphs in this paper express frequency using a logarithmic scale.

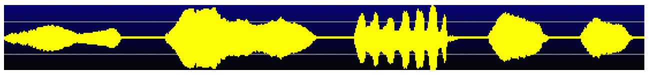

Figure 2: A waveform of a birdsong showing 5 bird calls [2]

To compute this difference, we need to carefully consider the data representation we will use for the audio signals in our work. Digital audio signals are stored using the pulse-code modulation (PCM) method (.wav files contain PCM data). While this format is great for storage and reproduction, it only stores the amplitude of the signal at every sampling point. As a result, its visual representation, called a waveform (see Figure 2), only provides information about the loudness of the signal at any given time.

However, subtracting the two waveforms would produce very poor results for our purposes. The reason for this is because the various processing steps applied during the mastering process modify the phase of the signal in a complicated manner (different frequencies are shifted by various amounts of phase). This is in part because digital equalizers attempt to replicate the process that analog equalizers use to boost or cut frequencies which involves shifting phase. While humans cannot perceive such phase shifts, things are different at the signal level. Therefore, if one simply subtracted two different masters, the result would not be just the audio that corresponds to the various cuts and boosts, but it would also include noise due to phase interference.

As a result, it is apparent that we need to represent our audio signals in the frequency domain. For this purpose, we experimented with two different data representations: spectrograms, and the average Short-Time Fourier Transform (STFT) of each signal. We discuss both below.

A *spectrogram* of an audio signal is a two dimensional plot with the y axis corresponding to frequency and the x axis to time. For time intervals of a specified length, the Fourier transform of the audio signal at that time is calculated and plotted on the graph. Brighter values correspond to louder volume for that frequency for that portion of the signal.

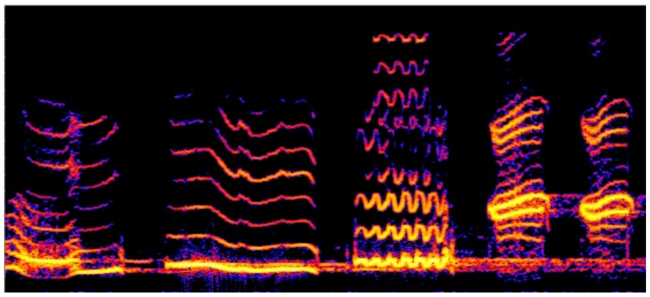

Figure 3: A spectrogram of a birdsong showing the distinct harmonic spectra of different bird calls [2]

Figure 3 presents the spectrogram for the birdsong whose waveform was shown in Figure 2. It also showcases the fact that real world signals have multiple frequencies present at any given time. In fact, even if a single note is played on a piano, the resulting signal will contain several frequencies, typically multiples of the lowest present frequency, called the fundamental frequency. The distribution and magnitudes of these frequencies is what gives each instrument its distinct sound, its timbre.

Spectrograms, such as the one in Figure 3, may choose to depict phase information through the use of colour. As phase can actually be problematic for our work, as described above, we remove it, which is why the remaining spectrograms in the paper are in grayscale.

To subtract spectrograms, we simply compute the absolute value of the difference in pixel intensity between corresponding pixels in the spectrograms of two different masters. We refer to the resulting spectrogram as a *difference spectrogram.*

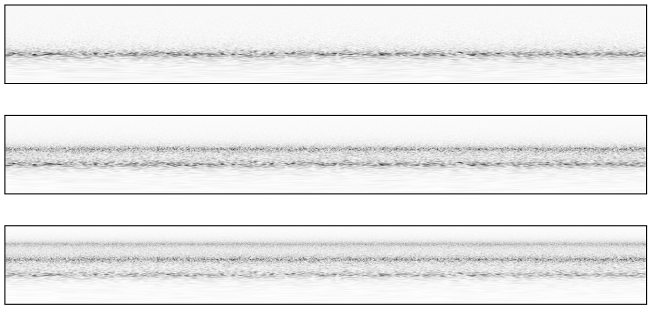

Figure 4: Three examples of pink noise difference spectrograms. The top one has had a boost at 261Hz, the middle one has had two boosts at 261Hz and 1046Hz, and the bottom one has had three boosts at 261Hz, 1046Hz, and 4186Hz. All boosts had a gain of 6dB and a width of 2 octaves. All frequencies correspond to C notes two octaves apart from each other

Figure 4 presents three difference spectrograms of equalized pink noise (pink noise contains the same amount of energy in each octave so it is usually more appropriate for musical applications than white noise which contains the same amount of energy per frequency). It is easy to see that each equalization operation produces a black band around its center frequency. This indicates that this data representation can be used to determine the number of equalization operations that separate two masters of the same song. We present experiments to that effect in Section 3.

The second data representation we use in this paper is that of the *average STFT* of each audio signal. The STFT of an audio signal gives us the amplitude for each frequency at a resolution of every few milliseconds depending on the parameters used. Since equalization operations are applied consistently throughout the duration of a master, we average these amplitudes across the duration of the audio signal and obtain a vector with the average amplitude for each frequency. This approach can still give a clear image of the amount of energy for each frequency, so it is well suited for recognizing frequency-based operations, such as equalization. To compute the difference between two average STFT vectors, we simply compute the absolute value of the difference between the two vectors.

We utilize the average STFT representation in Section 4 to successfully recognize the number and parameters of equalization operations that can be performed to transform a given master of a song to a different master.

3 PILOT STUDY

Our first attempt at addressing the problem of identifying the equalization operations that separate two versions of a clip made use of the difference spectrograms presented in the previous section.

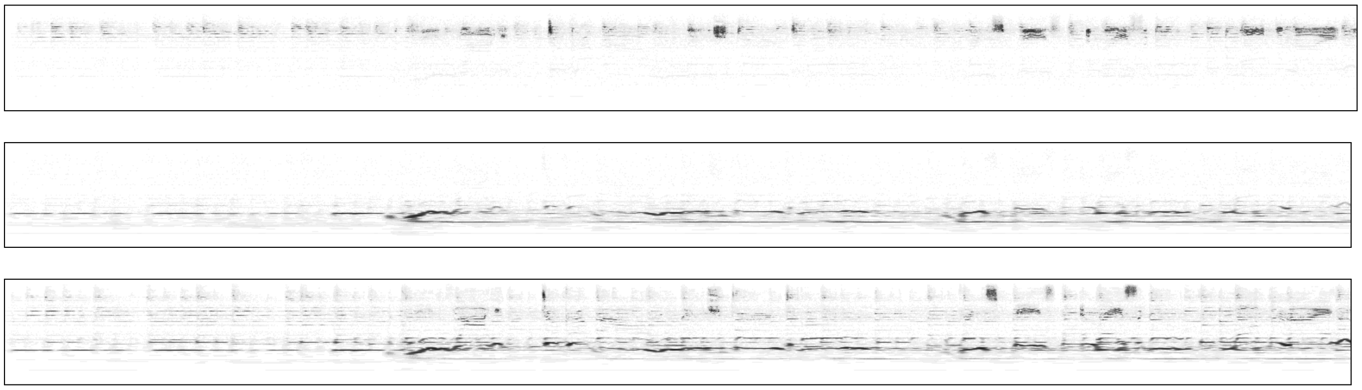

Figure 5: Three difference spectrograms from our pilot experiment (explained in the text)

We utilized a transfer learning approach that has shown promising results for audio classification tasks in our previous research [16].

Transfer learning is typically used when the number of available labeled data is not sufficient to effectively train a neural network for a particular classification task. In that case, it is still possible to use a network that has been trained for a task in a related domain. In our case, we utilized Inception v3 [9], an image classifier that has been trained with real world images from the ImageNet dataset [17]. We fixed the weights of all its layers except for the last one, and re-trained the last layer using difference spectrograms as the input vector [19]. The output classes reflected the number of equalization operations that corresponded to each spectrogram.

In particular, as shown in Figure 6, both the original and the equalized audio clips were converted into spectrograms using the asperes tool [1]. The resulting spectrograms from the datasets were then subtracted from their corresponding original spectrograms to generate difference spectrograms that become the input for Inception v3.

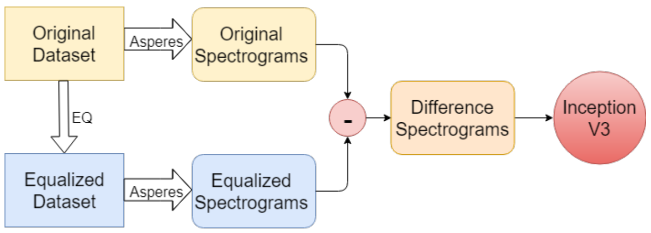

Figure 6: Pilot study data flow

For our pilot study, we used the following parameters for our equalization operations: the frequencies of the different equalization operations were at least one octave apart, while the width of each equalization operation is one quarter of an octave. As a result, there is no frequency overlap between any of the equalization operations. An approach that cannot classify successfully this set of difference spectrograms would have a much harder time with spectrograms that contain overlapping equalization operations.

We run the classifier for 4000 iterations (further iterations did not show meaningful improvement) with a learning rate of 0.01. The training batch size was 100 clips, and the test set was 10% of the images.

Despite the fact that our pilot experiment used non-overlapping equalization operations, we were able to achieve only an accuracy of 41.2%. In other words, only for 41.2% of the difference spectrograms was the classifier able to place them in the correct class with regard to the number of equalization operations performed.

Figure 5 presents sample difference spectrograms that illustrate some of the issues the classifier may have had (the actual images provided to the classifier are many times larger than shown here).

The top difference spectrogram corresponds to a single equalization operation at 4.2kHz. A black band is indeed visible at that frequency (the band is not continuous as in the case of the pink noise spectrograms in Figure 4 since in a song a particular frequency is not always present). This spectrogram happened to be classified correctly.

However, the model had a much harder time with difference spectrograms corresponding to multiple equalization operations. The middle spectrogram in Figure 5 corresponded to three equalization operations, and the bottom one to four. They were both misclassified. While it is possible to see black bands in both of these spectrograms, their number is not obvious to a human evaluator, and it appears that Inception v3 had similar problems as well.

These experiments indicated that, while it appears possible to detect the number of equalization operations on a pink noise difference spectrogram, the situation is quite different when it comes to music. Music does not have a uniform distribution of energy across the frequency spectrum (thankfully), so the patterns corresponding to equalization operations are not easy to pick up by either humans or machines.

4 METHODOLOGY

The fact that an otherwise successful classifier like Inception was unable to detect the number of equalization operations in difference spectrograms indicated to us that a change in our approach was necessary. We decided to instead detect the exact equalization operations directly. Determining their number, then, becomes a simple post-processing step. In this section, we describe the methodology we used for all our experiments.

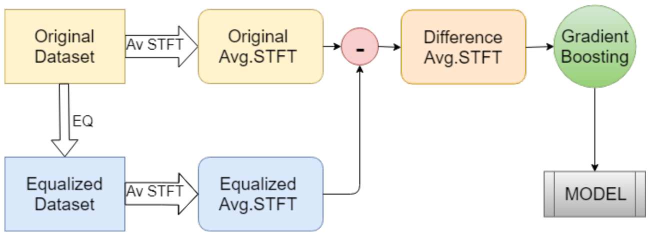

Figure 7: Data flow to produce a trained model that detects sets of equalization operations

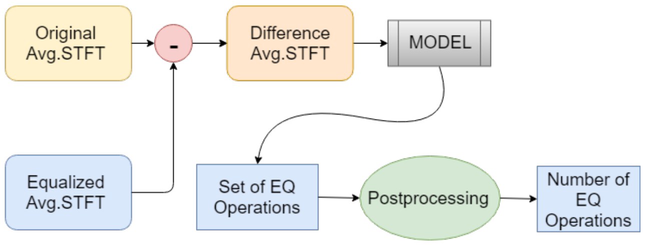

Figure 8: Data flow to determine the set and number of equalization operations in the testing phase

Assume that we start with a set O of m original masters (professionally mastered songs). The first step of our methodology is to produce n equalized versions for each professionally mastered song O_i in our dataset, where $i \in [1, m]$. We refer to each such equalized version as $E_j(O_i)$, where $j \in [1, n]$. We select n sets of equalization operations that are applied to the original masters. Each set s_j may be a singleton, e.g. one equalization operation with a center frequency of 200Hz, a gain of -6dB, and a width of 1 octave, or it may contain several equalization operations, e.g. 6 equalization operations with center frequencies 100Hz apart, ranging from 300 Hz to 800Hz, all with a gain of 9dB, and a width of a quarter octave. Each set s_j of equalization operations corresponds to a class c_j for the output of our classifier.

In order to produce appropriate input vectors for our classifier, we deployed the average STFT data representation described in Section 2. We first compute $A(O_i)$ and $A(E_j(O_i))$, where A represents the average STFT function, whose output is a vector of size 1024. Each element in $A(O_i)$ represents the average magnitude for a particular frequency in the human hearing range. Next, we obtain the difference $D_{ij} = |A(O_i) - A(E_j(O_i))|$. Vectors D_{ij} are the inputs to our classifier.

The next step in our methodology is to employ a classifier that can be used to train a model that recognizes equalization operations. We found *gradient boosting* to work exceptionally well for our purposes. Gradient boosting is a machine learning technique which produces a prediction model in the form of an ensemble of weak prediction models [20]. It is a boosting technique, which means that it is based on sequentially improving weak predictors using gradient descent to come to an accurate predictor. The weak predictors are typically decision trees which are sequentially built using a greedy method. Each new predictor is built in sequence by summing the result of the previous predictor with some other learned parameter, thus its basis on gradient descent. At each step of boosting, the newly learned parameter learns of the distribution of the residuals of the previous model [6].

Our methodology proceeds by training a model M that attempts to classify each input vector D_{ij} into the correct class $c_j = M(D_{ij})$. Figure 7 shows the process described so far in graphical form.

After training, the model is saved, and can be used to classify any difference vector $d_j = |A(o) - A(E_j(o))|$, where o is a professionally mastered song that was not part of our original set O (see Figure 8). If the classifier assigns vector d_j in class c_j, then it has correctly identified the exact set of equalization operations that transformed o into $E_j(o)$.

If the goal is to identify the number of equalization operations that transformed o into $E_j(o)$ rather than the exact operations, then a simple post-processing step produces output equal to the cardinality of set s_j that corresponds to class c_j (see Figure 8).

In summary, our methodology ensures that the two goals of this paper set out in the introduction are met:

(1) It produces a trained model that can identify the sets of equalization operations that have been used to transform a piece of audio into another (set s_j that corresponds to output class c_j)
(2) It can be used as a training tool for mastering engineers. Consider the difference vector $d = |A(p) - A(t)|$, where p is the professional master of a song, and t is the master provided by a trainee. If d is the input to our model, the output of the post-processing step can be used as a rating of the quality of t (the larger the output, the poorer the quality).

A couple of threats to the validity of our approach must be mentioned:

(1) The same equalization operation will have a different effect depending on the key of the song being equalized. For example, a song in C major is likely to contain many C notes. If the equalization operation is centered at the fundamental frequency of a C note, it will have a much larger effect than if it is centered on a note not in the C major scale.
While we believe this is an important issue worthy of further study, it does not alter the results presented later in this paper as the songs in our datasets were of unknown key, and equally as likely to match the equalization operations selected as not.
(2) We applied the same equalization operation across the duration of each clip. In practice, a mastering engineer may apply different equalization operations for different parts of a song, e.g. apply more dramatic equalization for the chorus as opposed to the verse.
While the above is true, the validity of our approach remains. In a practical setting, it is still possible to apply our approach only to parts of a song by isolating the relevant part, or duplicating a part that is shorter than the duration our model is trained with.

In the next section. we present experiments that apply the above methodology using different sets of equalization operations to assess the accuracy of our approach.

5 EXPERIMENT RESULTS

5.1 Classifier implementation

An important step that has to be completed before we can run experiments is to choose an appropriate implementation of gradient boosting. We chose the *catboost* implementation of gradient boosting [22]. Catboost uses binary decision trees, where the depth at which the trees will terminate can be specified in order to avoid overfitting. It also addresses the problem of prediction shift (where the new learner to be added at each step to the previous predictor causes a shift with respect to the previously learned set of residuals) by creating a model with unshifted residuals using an approach called ordered boosting [14]. Catboost has been used by many winning entries in Kaggle competitions in the last few years [8].

For each one of our experiments, the catboost algorithm was run for 1000 iterations. We chose a tree depth of 5, and the loss function used was MultiClass [23]. The STFTs were obtained using the librosa library [18]. The STFT window size used was a standard size of 2048 samples with a hop of 512 samples, producing overlap as to not lose information.

5.2 Initial experiments

To begin with, we performed several experiments to determine the importance of various factors in constructing our datasets.

The first factor we considered was the length of the audio clips used in our approach. We conducted several pilot experiments that clearly demonstrated that having more than 30 seconds of audio has no effect on the accuracy of our results. As a result, all clips in all our experiments are 30 seconds long. This length of the audio was chosen both for efficiency reasons (it is much faster to process 30 seconds of audio as opposed to a full song), as well as consistency reasons (the original songs we used all have different lengths). To ensure meaningful content, we omitted the first 30 seconds of each song (to avoid initial silence or fade-ins) and used the next 30 seconds for our experiments.

The next factor we experimented with is the genre of the music in our audio clips. The experiments presented later in this section are based on a song corpus of approx. 800 songs in the rock genre. We performed experiments using song corpora using other genres, such as classical music, heavy metal and a mix of all three genres. These experiments yielded insignificant differences in the results, so the rest of this section focuses on the results we obtained for the rock song corpus[2].

Finally, we conducted experiments relating to the parameters of the equalization operations. We found that once the value for gain becomes larger than 6dB, accuracy does not improve no matter how high the gain. Whether the gain was positive or negative was also not a factor, so we fixed the gain value for all our experiments at +6dB. We experimented with smaller gains (3dB) that proved hard for our models to detect. In Section 6, we show that even music professionals have a hard time identifying equalization operations with a gain of 3dB.

[2]The music in the aforementioned corpora is copyrighted, so they are unfortunately not publically available

5.3 Dataset creation

We created 9 equalized datasets whose goal was to model possible equalization moves that a mastering engineer would have to make to master a song. To test to what precision the equalizations could be detected by a model, the datasets varied in the widths of the equalization operations performed, as well as the intervals between frequencies (in the case of multiple simultaneous equalization operations).

We used three different widths (three different Q values) to create our datasets: two octaves, one octave, and half an octave (Q values of 0.7, 1.4, and 2.8, respectively).

We also used three different minimum intervals between the frequencies of simultaneous equalization operations. Due to the bell curve nature of equalization, mastering engineers do not perform equalizations whose center frequencies are very close. As a result, we chose minimum intervals of 1 octave, half an octave (a music interval of an augmented fourth), and one fourth of an octave (a music interval of a minor third). We chose the exact frequencies so that the notes at the center of the equalization operations were respectively: all C notes in the human hearing range, all C and F# notes, and C, D#, F#, and A notes.

A mastering engineer does not perform more than a few equalization operations for a given song. As a result, we performed up to 7 equalization operations per clip, the center frequencies of which were randomly chosen from the set of notes described above. This set of equalizations corresponds to the target label for each clip.

For each equalized dataset we created, we combined one of the width values with one of the minimum frequency intervals, resulting in 9 datasets. Each dataset was named based on these values, e.g. dataset F1/2W2 contains equalized versions of the original rock clips, where the minimum interval between the frequencies of simultaneous equalization operations is half an octave, and the width of every equalization operation is two octaves.

The reason behind the different datasets is to stress the limits of our approach. More specifically, some of the datasets will have much more easily discernible equalizations than others where equalizations might overlap. If the width used is very wide, and the frequency interval is small, the widths will overlap when multiple equalizations are applied. This should make the accurate detection of the exact equalization operations much harder.

5.4 Accuracy results

Table 1 presents the accuracy values we obtained for our 9 datasets. Correct recognition for each clip meant that our approach correctly identified both the number of equalization operations as well as the center frequency for each equalization operation.

		Frequency Interval		
		1	1/2	1/4
Width	1/2	96.4%	97.6%	97.9%
	1	93.9%	93.6%	89.7%
	2	88.7%	85.8%	77.4%

Table 1: Accuracy results for equalization recognition

As can be seen from the results, accuracy is very high when the equalization operations do not overlap with each other as is the case in the top left part of the table. As overlap starts to become a factor, our accuracy slightly decreases. The only comparatively poor result is when the equalization operations may be as close as 1/4 of an octave while their width is two octaves. Considering the significant amount of overlap (something unrealistic for music mastering), our accuracy result of 77.4% is quite impressive.

Overall, our approach is clearly successful in recognizing complex equalization operations, a result that can have significant impact in many music-related applications (see the future work discussion in Section 8).

In the case of music mastering, the more realistic conditions under which mastering engineers would have to perform these equalizations would be with frequency intervals of an octave or half of an octave. A quarter of an octave would be too surgical of an operation to be considered appropriate for mastering purposes, especially with wider bandwidths. Similarly, a width of half an octave would be much too small to be appropriate for mastering. Mastering usually involves frequency cuts and boosts across wider bands. Accuracy results for these mastering-relevant equalizations from Table 1 are shown in the second column of Table 2.

	Before Postprocessing	After Postprocessing
F1W1	93.9%	95.6%
F1W2	88.7%	90.5%
F1/2W1	93.6%	93.6%
F1/2W2	85.8%	85.8%

Table 2: Accuracy results relevant to mastering

One of the goals of this paper is to develop an automatic way to rate a master produced by a trainee against a professional master. As a result, we need to determine the number of equalization operations needed to transform one master to the other. For this purpose, we post-process our equalization recognition results so that the result is accurate if the number of equalization operations detected is correct regardless of whether the center frequencies of these operations were correct. After this postprocessing, our accuracy values for the mastering relevant datasets are shown in the third column of Table 2.

The increase in accuracy after postprocessing is minimal (approx. 3%) from which we can conclude that the error in the data comes from something other than the model misclassifying songs with the same number of equalization operations. This could suggest that the model performs well at identifying specific frequencies and when an error occurs it detects a non existent equalization or fails to detect an equalization.

The final set of experiments tested if our models that were trained using rock songs can be used to classify songs in other genres. The songs we used for the user study presented in the next section were of various genres, including electronic and acoustic music. We run these songs through our models using the process shown in Figure 8 and obtained accuracy values. Table 3 shows the results of these experiments. By comparing the numbers in Table 3 to those in Table 1, one can see that accuracy has dropped by a small margin. This indicates that a large amount of transfer learning happens between genres, but training a model with the same genre as the songs it will be tested with is necessary for the highest accuracy.

		Frequency Interval		
		1	1/2	1/4
Width	1/2	93.04%	95.45%	95.68%
	1	87.83%	88.18%	84.86%
	2	82.60%	76.36%	68.65%

Table 3: Accuracy results on user study songs

Overall, the results presented in this section indicate that our approach can not only provide a measure of how close to the gold standard is a candidate master, but also provide accurate suggestions as to which equalization moves should be applied to a piece of music in order to get it mastered properly. We validate this conclusion in the next section through the means of a user study.

6 USER STUDY

To assess how the performance of our models compares with human performance, we conducted a user study with 15 participants of various levels of experience with mastering. The goal of the user study was to determine whether the trained models we obtain with our methodology can identify equalization operations better or worse than humans at different levels of experience. The questions in the user study were of varying levels of difficulty in order to better assess the performance of the various participants and our models.

The participants were asked to listen to differently equalized versions of six different songs. Each song came with a set of questions testing the participants to see if they could recognize what equalization operations were performed on the songs. Some questions involved comparing a mastered song to an equalized version of the same song. Other tasks did not provide the participants with the knowledge of which clip was the original master, and asked them to rate a list of different versions of the song. The complete user study questionnaire is available online [3].

The equalization operations done to the user study songs were only in the treble or bass frequency bands which correspond to high and low frequencies respectively, as identification of precise frequencies requires significant expertise that even music professionals may lack. The width of the equalization operations used was two octaves, so as to ensure that the changes could be heard by the participants. Our trained model that corresponds the closest to these parameters is F1W2, so in the following we compare the results of our participants to those obtained by F1W2.

Table 4 presents the results we obtained from our model as well as the study participants. Correct answers are shown in green, while partially correct answers are shown in yellow (we considered a participant response that treble has been decreased as partially correct when bass had instead increased, as the volume at which the participant listened to each clip may have been a factor). Question 2 asked the participants to choose from among 5 clips the one that sounded more like a professional master. Question 4 asked the participants to rank three different versions of a clip that had progressively more equalizations applied to them. Questions 5 and

Question		1	2	3	4	5	6
Correct Answer		Bass has been increased	B	Increase Treble	1: A 2: C 3: B	Increase Bass	Decrease Bass
Model Answer		Bass/Mids increased	B	Increase Treble	1: A 2: B/C	Increase Bass Increase Treble	Increase Bass Increase Treble
User #	Experience (Years)						
User 1	0-1	Treble has been decreased	B	Increase Treble	1: A 2: C 3: B	Nothing	Nothing
User 2	0-1	Bass has been increased	A	Increase Treble	1: C 2: A 3: B	Nothing	Decrease Bass
User 3	0-1	Bass has been increased	D	Increase Treble	1: B 2: A 3: C	Nothing	Increase Treble
User 4	0-1	Nothing has been changed	A	Increase Treble	1: A 2: C 3: B	Nothing	Decrease Treble
User 5	0-1	Treble has been decreased	A	Increase Bass	1: B 2: C 3: A	Nothing	Decrease Treble
User 6	0-1	Treble has been decreased	C	Increase Treble	1: A 2: C 3: B	Nothing	Decrease Treble
User 7	0-1	Treble has been increased	D	Decrease Treble Increase Bass	1: B 2: A 3: C	Increase Treble	Decrease Treble Decrease Bass
User 8	0-1	Nothing has been changed	A	Decrease Treble Decrease Bass	1: A 2: B 3: C	Nothing	Decrease Treble
User 9	1-4	Nothing has been changed	B	Increase Treble	1: C 2: A 3: B	Nothing	Decrease Treble
User 10	1-4	Bass has been increased	A	Nothing	1: A 2: C 3: B	Decrease Bass Nothing	Increase Bass Nothing
User 11	1-4	Bass has been increased	B	Decrease Treble	1: A 2: C 3: B	Increase Treble	Nothing
User 12	1-4	Treble has been decreased	A	Increase Treble Increase Bass	1: C 2: A 3: B	Decrease Treble	Decrease Bass
User 13	4-8	Bass has been increased	B	Increase Treble	1: A 2: C 3: B	Nothing	Decrease Bass
User 14	4-8	Bass has been increased	C	Increase Treble	1: A 2: C 3: B	Nothing	Increase Bass
User 15	8+	Bass has been increased	B	Increase Treble	1: B 2: C 3: A	Nothing	Nothing

Table 4: User study results (explained in the text). Best viewed in colour

6 used equalization operations with a gain value of 3dB to test if humans can detect such operations (our models performed quite poorly on them).

Out of all 15 participants, 8 had 0-1 years of experience, 4 had 1-4 years, 2 had 4-8 years and one had more than 8 years. This provided us with enough variability in experience to see how skill affected performance on each of the tasks. One of the main conclusions we arrived at from having many users with minimal previous experience in mastering or mixing was that even very simple questions proved to be very difficult for these users. Given this, it is safe to say that for future user studies and experiments the participants should have a decent amount of experience in mastering to gain any valuable insight into human performance in these tasks. At the same time, this indicates that approaches such as the one presented in this paper can be immediately helpful to anyone interested in mastering, or ear training in general for that matter.

Based on the answers of the first four questions, it is fair to say that our model performs usually better than participants with 0-1 or 1-4 years of experience. This would indicate that it can be used for training purposes with novice mastering engineers. There were only 3 participants with more than 4 years of experience, but they all performed better than our model. Further study will need to be conducted to determine how we could improve our model's performance to the level of a music professional.

For the questions related to equalization operations with a gain of 3dB the participants seemed to not be able to recognize the differences regardless of skill level. Even the one participant with 8+ years of experience did not get the correct answer for either question. From this we can conclude that such low gain equalization operations are hard to detect even by music professionals, which would mean that they are not as critical for our model to detect as well.

In conclusion, the user study indicated that our model performs better than beginner mastering engineers but not as well as experienced ones, which is appropriate for our purposes, as our goal is not to replace music professionals, but to provide automatic help to mastering engineers in training.

7 RELATED WORK

Machine learning classification techniques are popular in music, where many researchers have attempted to classify different features in music datasets like genre, instruments and mood. The most popular methods for the classification of such features involve the use of deep neural networks. Source separation techniques have also been used with machine learning tools attempting to isolate vocals, lyrics, and instruments. Although not as popular, mixing and mastering research has recently started to get some traction with machine learning techniques.

Convolutional neural networks have been used to classify non-musical environmental sounds by Piczak [13]. In her paper, she uses a convolutional neural network whose architecture consists of a mixture of fully connected layers, max-pooling layers, and convolutional layers to classify three small datasets of environmental sounds. The first, ESC-50, consists of 2000 short sounds of 50 classes; the second, ESC-10, consists of 400 sounds of 10 classes from the ESC-50 dataset; and the third, UrbanSound8K, with 8732 sounds of 10 classes. Her model performed best on the UrbanSound8K dataset with an accuracy of 73.7 percent, showing the validity of these types of techniques.

Another example of classification techniques being applied to environmental auditory events comes from the Tampere University of Technology [7]. In their paper, deep neural networks are used to recognize events like motorcycles, rain, or babies crying. The dataset they used had 61 distinct classes with 1325 audio files. The architecture used for the classification involved a network with 5 hidden fully connected layers with 70 neurons each, and was trained using backpropagation. A pre training step of unsupervised training was applied to the network before the supervised portion. Their results achieved an accuracy of 64.6 percent which they compared to 2 layer classifiers which performed at an accuracy of 60.2 proving the efficacy of deep neural networks on these types of audio classification problems.

In a study related to music applications, researchers from the university of Surrey, UK used deep neural networks to separate singing voice from an audio file. They first attempt to estimate the pitch of the voice to extract the component of the signal responsible for said pitch. By estimating the fundamental frequency of the voice using a deep neural network, the use of the popular YINFFT algorithm becomes much more effective. The researchers applied a short term fourier transform to the audio file to obtain a spectrogram of the data. The data consisted of a mixture of a voice and an additional signal, where the target for the neural network to predict was the mask which when subtracted from the mixed signals would produce the vocal signal. The data used was 200 audio files from the iKala dataset, and the network was trained using SGD. The results yielded much better performance for the YINFFT algorithm [15].

Similarly, Park and Lee train a convolutional neural network to discriminate between music and noise audio files [12]. The data used was converted into spectrogram format as in the previous example. The architecture used for this neural network consisted of 3x3 kernels for each convolution layers with max pooling. The researchers used 450 different popular music audio tracks for the music portion of the data and for the noise they used environmental sounds, like restaurant ambiance or street traffic. The network was trained using backpropagation and stochastic gradient descent. The classification was close to perfect with an accuracy of 99.7 percent, showing that the network learned how to distinguish music from noise.

When working with audio, one may need to transform the data into more useful representations to use with different existing models. Spectrograms are usually helpful with working with audio and convolutional neural networks. Lonce Wyse provides a good summary of differing representations, especially spectrograms for audio when working with convolutional neural nets. His research looks at sonograms, spectrograms, and other representations and compares their performance on different machine learning tasks as well as the different kinds of information lost by each type of representation [21].

McCormack et al. have also done work in the field of mastering and mixing in the use of dynamic range compression using FFT representations of audio. They attempt to improve the use of compressors in mixing by creating a compressor that works as a high resolution multiband dynamic range compression. Their work could be used in automatic mixing applications. Their design uses an STFT based implementation to maintain the time domain in the audio representation [10].

8 CONCLUSION

This paper presented a machine learning-based approach that can detect equalization operations performed on musical audio. The main contributions of the paper are:

- A standard audio processing technique, such as STFT, and a standard machine learning algorithm, such as gradient boosting, is sufficient to recognize equalization operations with high accuracy.
- A transfer learning approach using difference spectrograms and a powerful image recognition model is not well suited to this task.
- Our approach can be used to help with the training of mastering engineers by automatically providing a rating of their work, as well as suggestions on the equalization operations to improve it.

There are several avenues for further research for our work:

- We plan to conduct further experiments with our methodology using many more combinations of parameters in order to study how our accuracy changes as we move in the space of parameter combinations. This will hopefully help us improve our approach even further.
- Our plan is to build a web-based version of the mastering training tool in order to facilitate its use by the whole community of music professionals online. This will allow us to further understand the effectiveness of our approach, hopefully spurring improvements that will make the tool a standard part of the arsenal of music educators.
- We plan to investigate the possible use of difference spectrograms further by training a model from scratch using only difference spectrograms as the input. It will be interesting to see if such a model will perform closer to our gradient boosting ones. Alternatively, we would like to experiment with different image classifiers, such as the IBM Watson Visual Recognition service [4]. We were only able to use this service for this work in a limited fashion, due to the limits of a free account.
- We plan to expand this work to include other aspects of music mastering, such as dynamic range compression, which is often used to make a master consistently loud by increasing the volume of the softer parts of a song.
- The work presented in this paper is a springboard for further musical applications of machine learning. We'd like to utilize the knowledge gathered from our mastering training tool towards building models that can suggest equalization operations for mastering even in the absence of a gold standard. Our models' ability to detect equalization operations can also be used for other musical applications, such as the automatic mixing of songs.

REFERENCES

[1] Asperes - sound to image (spectrogram) and image to sound converter. 2004. (2004). http://www.findbestopensource.com/product/asperes
[2] Looking at Sound. 2005. (2005). http://betarhythm.blogspot.com/2005/11/looking-at-sound.html
[3] Pedro Casas. 2018. APTLY Lab Survey. (2018). https://form.jotform.com/80285070476256
[4] IBM Watson Developer Cloud. 2017. Visual Recognition | IBM. (2017). https://www.ibm.com/watson/developercloud/visual-recognition.html
[5] Hui Fang and Meng Zhang. 2017. Creatism: A deep-learning photographer capable of creating professional work. *arXiv preprint arXiv:1707.03491* (2017). https://arxiv.org/abs/1707.03491 bibtex:creatism.
[6] Jerome H. Friedman. 2001. Greedy function approximation: A gradient boosting machine. *The Annals of Statistics* 29, 5 (Oct. 2001), 1189–1232. https://doi.org/10.1214/aos/1013203451
[7] Oguzhan Gencoglu, Tuomas Virtanen, and Heikki Huttunen. 2014. Recognition of Acoustic Events using Deep Neural Networks. *2014 22nd European Signal Processing Conference (EUSIPCO)* (Nov. 2014), 5.
[8] Kaggle Inc. 2018. Your Home for Data Science. (2018). https://www.kaggle.com/
[9] Inception-v3. 2016. (2016). https://www.tensorflow.org/tutorials/image_recognition
[10] Leo McCormack and Vesa Valimaki. 2017. FFT-based Dynamic Range Compression. *Proceedings of the 14th Sound and Music Computing Conference, July 5-8, Espoo, Finland, At Espoo, Finland* (July 2017).
[11] Bobby Owsinski. 2010. *The Audio Mastering Handbook: The Mastering Engineer's Handbook.* Cengage Learning.
[12] Taejin Park and Taejin Lee. 2015. Music-Noise Segmentation in Spectrotemporal Domain Using Convolutional Neural Networks. (2015), 2.
[13] Karol J. Piczak. 2015. Environmental sound classification with convolutional neural networks. IEEE, 1–6. https://doi.org/10.1109/MLSP.2015.7324337
[14] Liudmila Prokhorenkova, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush, and Andrey Gulin. 2017. CatBoost: unbiased boosting with categorical features. *arXiv:1706.09516 [cs]* (June 2017). http://arxiv.org/abs/1706.09516 arXiv:1706.09516.
[15] Gerard Roma, Emad M Grais, Andrew J R Simpson, and Mark D Plumbley. 2016. Singing Voice Separation Using Deep Neural Networks and F0 Estimation. 2.
[16] Mark Shtern, Rabia Ejaz, and Vassilios Tzerpos. 2017. Transfer Learning in Neural Networks: An Experience Report. In *Proceedings of the 27th Annual International Conference on Computer Science and Software Engineering (CASCON '17).* IBM Corp., Riverton, NJ, USA, 201–210. http://dl.acm.org/citation.cfm?id=3172795.3172818
[17] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. Rethinking the Inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* 2818–2826. http://www.cv-foundation.org/openaccess/content_cvpr_2016/html/Szegedy_Rethinking_the_Inception_CVPR_2016_paper.html
[18] Librosa Development Team. 2018. Librosa library. (2018). https://librosa.github.io/librosa/
[19] Tensorflow. 2016. How to Retrain Inception's Final Layer for New Categories. (2016). https://www.tensorflow.org/tutorials/image_retraining
[20] Wikipedia. 2018. Gradient Boosting entry on Wikipedia. (2018). https://en.wikipedia.org/wiki/Gradient_boosting
[21] Lonce Wyse. 2017. Audio spectrogram representations for processing with Convolutional Neural Networks. (2017), 5.
[22] Yandex. 2018. Catboost. (2018). https://tech.yandex.com/catboost/
[23] Yandex. 2018. MultiClass loss formula. (2018). https://tech.yandex.com/catboost/doc/dg/concepts/loss-functions-docpage/#loss-functions__multiclassification