Sound Emotion Prediction: Machine Assisted Sound Effects Selection

Chung Kang Chen

Submitted in partial fulfillment of the requirements for the

Master of Music in Music Technology

in the Department of Music and Performing Arts Professions

Steinhardt School

New York University

Advisor: Tae Hong Park

Reader: Elena Georgieva

Spring 2022, June 17

ABSTRACT

Sound Editors and sound designers face the tremendous task of having to listen to hundreds of audio files per day. Finding the right sounds for a specific emotion can be an overwhelming task, especially when listening to thousands of files just to find a suitable one. This study presents an application that uses two neural networks to predict the emotion of audio files based on previous soundscape emotion recognition studies. Audio features are extracted from sound files to serve as the input for the neural networks to train and obtain inference. As descriptions of emotions are often subjective, Russel's circumplex of affect (valence-arousal model) is chosen as the method to represent the emotion value of the sound files. The resulting output of both neural networks is a valence arousal value that can be plotted on the user interface that allows users to interact and see the emotion distribution of their audio files. The objective evaluation of both deep learning models suggested that the neural networks performed similarly to previous studies. Whereas the subjective evaluation of the emotion recognition application shows positive feedback on the usage of the presented application within the current professional workflow.

ACKNOWLEDGEMENTS

I would like to thank the following people:

First and foremost, my family for their unconditional support they've provided me throughout the entire process of getting my master's degree. Without them I would have struggled a lot more.

Dr. Tae Hong Park, for his guidance and advisement on creating the program for this thesis. Your straightforward advice has made my path to finishing the thesis significantly easier.

All the friends I've made in the Music Technology program and across NYU. Every one of you have been an inspiration to push myself harder and improve myself every single day.

TABLE OF CONTENTS

LIST OF TABLES

1. INTRODUCTION

The way people enjoy their free time have gone through a major shift in recent years. In a similar fashion radio and television have changed how people received information and entertainment back in the first half of the 20th century, the internet and specifically mobile devices are dramatically changing how people spend their leisure time. An online study from DoubleVerify have shown that in the year of 2020, the amount of time spent on digital media have more than doubled from an average of 3 hours 17 minutes to approximately 7 hours (Four Fundamental Shifts in Media & Advertising During 2020, 2020).

Online video content especially has been on the rise in recent years with video centric platforms playing an increasingly dominant role in how people entertain themselves and socialize with each other. Instagram (2010), a photo sharing application have recently shifted its focus from being a photo sharing platform to a video sharing platform. TikTok (2016), a short form video sharing app has recently toped Google as the most visited website on the internet (Tomé & Cardita, 2021). Even in areas where traditional media used to have a monopoly, such as film and TV series, have slowly shifted its focus from theatrical releases to online distributions such as Netflix and Disney+.

As video is a collaborative medium that marries visuals with audio, demands of video and audio productions have also increased (Global Film and Video Services Market Report 2021 - Opportunities and Strategies to 2030 - ResearchAndMarkets.Com, 2021). Yet with the high demand of digital media, digital tools that supports the creation of such media have not improved as much to make up for the increasing demand, especially audio production tools. While video technology and technology for visual effects have improved immensely, audio technology have

been improving at a significantly slower rate. Tasks such as syncing audio, re-conforming, and searching sound effects often still requires a tedious amount of work to complete.

Sound editing and design has always been a crucial part of any audio production process. Without it, fictional worlds and emotional moments in films, games, and music would fall flat. Yet the process of finding the right sounds often takes up tremendous amount of time as sound libraries still rely solely on file names and metadata to describe the contents of the file. While realistic sounds don't typically pose a challenge in the current workflow, it is often abstract sounds that describe emotions (eerie, joyful, sad) that require the extra time investment to find something suitable. The use of sound effects is also limited to the file names, making the process of searching sound effects extremely unintuitive when artists are trying to create something original.

This paper aims to leverage the computation power of modern fast computers to speed up the process of searching abstract, emotional sound effects by creating a program that uses machine learning models to perform emotional recognition on sound files. In the first section, we will be looking at previous research on emotion (affective) science in music and soundscapes; comparing how different current technologies perform on predicting emotions. In the second section, details of creating the prototype software for predicting emotions of analyzed sound files as well as the training steps of the machine learning will be elaborated. In the third section, we will evaluate the program objectively by comparing it with previous research attempts as well as subjectively, by testing the user experience. In the last section, the outcomes of both the objective and subjective evaluations as well as possible future improvement on the program will also be discussed in detail.

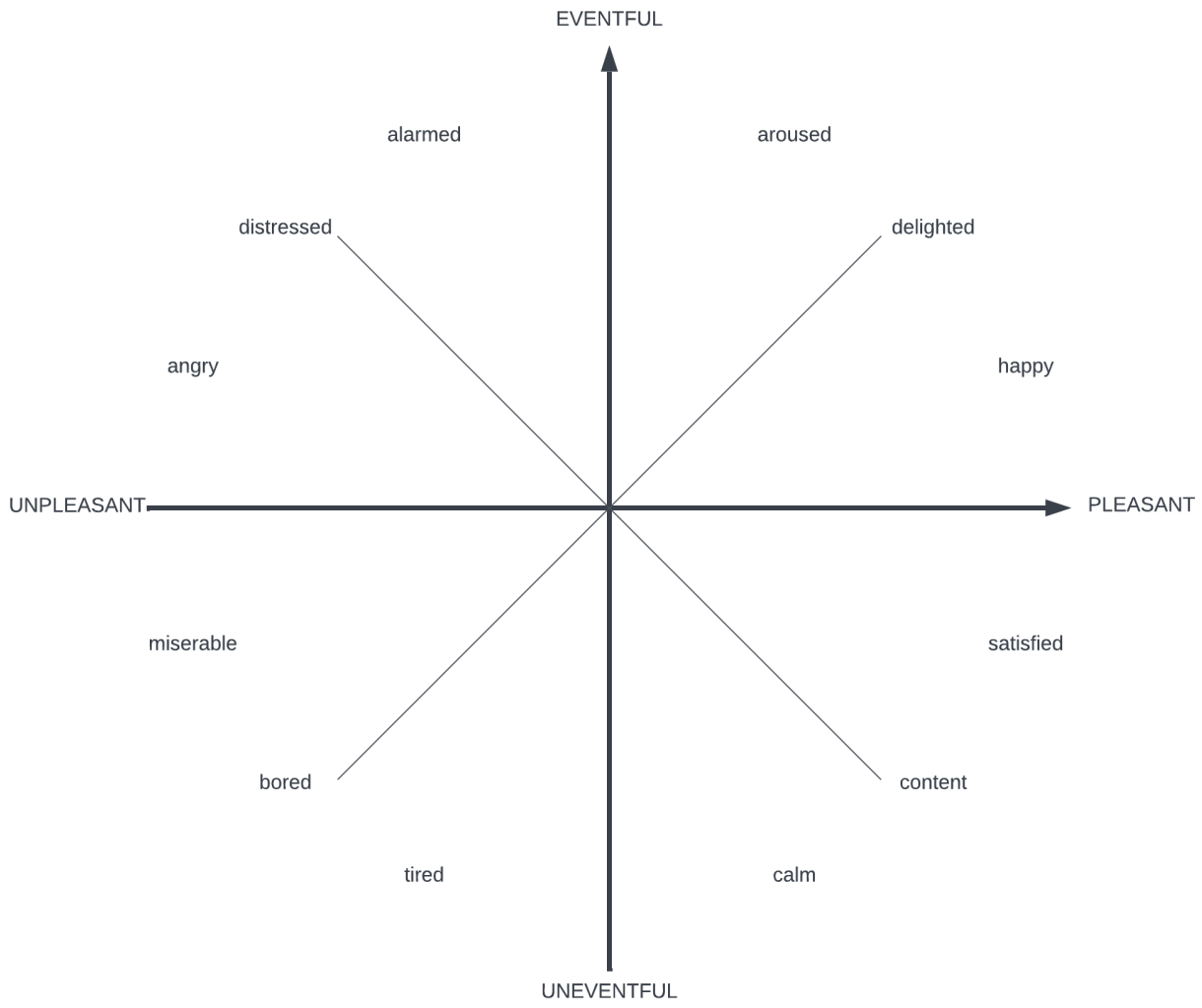

Figure 1: Valence-Arousal Space

2. PRIOR WORK

1. Emotion Recognition

Previous research in music emotions show that there are two major schools of representing emotions: a categorical representation and a dimensional representation (Kim et al., 2010). Categorical approach classifies music based on the relevance of said music to a set of emotional descriptors such as passionate, humorous, aggressive, etc., whereas a dimensional approach measures moods on a continuous scale of descriptors or a multidimensional model.

According to Russell, who propose a continuous dimensional model with each emotion systematically related to each other known as the circumplex of affect (Russell, 1980), the two dimensions of affect are valence and arousal. Valence is the degree of perceived pleasure/ pleasantness while arousal is used to describe the eventfulness or the amount of perceived stimulation. As shown in a Valence-Arousal (V-A) space (Figure 1), the x-axis is the valence, ranging from negative to positive emotions, and the y-axis is the arousal, ranging from low to high. In this research, the valence-arousal model will be used to represent the emotions as well as provide visual feedback of the application prediction.

2. Previous Soundscape Emotion Recognition Models and Systems

2.1 Models

While there is an abundance of models trained in previous research for music and speech emotion recognition tasks, a relatively small amount of research and models have been dedicated to the soundscape emotion recognition task. Here are some of the machine learning models from previous research.

Fan et al. (2016) compared two machine learning models and concluded that stepwise linear regression models perform better than support vector regression (SVR) models. Though the conclusion may be due to the cause of not having a large enough sample size, resulting in the overfitting of the SVR model, their research showed an interesting difference in preference for eastern and western culture when it comes the correlation of valence and arousal. While western culture has a preference in associating high arousal with positive affect, eastern culture has an inverse relation between the two (valence varies inversely with arousal). They have concluded

that there is a high level of agreement on valence and arousal in soundscapes from their participants.

Although previous research of soundscape emotion recognition used several databases in their studies, they are either private sound effects libraries, databases originally created for other purposes, or unreleased audio clips and annotations (Fan et al., 2016). However, to have a baseline to compare and improve upon in future research, Fan et al. (2017) created a new publicly available database called Emo-Soundscapes to accurately model and estimate the perceived emotion of a sound source. The dataset consists of 1213 audio clips, with 600 curated audio clips from Freesounds.org (Music Technology Group at Universitat Pompeu Fabra, n.d.-b) and 613 augmented audio clips that is a mixture of the 600 audio clips as well as all its valence and arousal labels.

The audio clips in the database were curated in six steps. First, the audio clips were automatically downloaded and manually selected to ensure the quality. The selected clips were then automatically segmented by a background/ foreground-classifier (Thorogood et al., 2016) that keeps audio regions with consistent characteristics. Audio clips that are not highly correlated to the semantic tags were then discarded while clips that have a high correlation between the sounds and the semantic tags were selected. The results were 600 six-second clips, 100 each of Schafer's six soundscape categories (Schafer, 1993).

The annotations were gained from crowdsourcing experiments where 1,182 annotators from 74 different countries compare audio clips base on the perceived valence and arousal. Annotators were told to rank the audio clips based on the valence and arousal in a pairwise comparison. Due to the task being highly subjective and ratings between different annotators and even the same annotator not being consistent, the paper chooses to use a ranking based system.

As the consistency between different annotators across different countries were important for the quality of the database, the researchers tested the inter-subject reliability to see whether different participants reach similar conclusions using Krippendorff's alpha (Krippendorff, 1970) and percentage of agreement as the measuring metric. Krippendorff's alpha value ranges from 0 to 1, with 0 being unreliable and 1 meaning perfectly reliable. The results of the inter-subjectivity test concluded with the agreement percentage reaching an 81.99% confidence level and 0.21 to 0.40 for the alpha, indicating that there was a reasonable amount of agreement and consistency between different subjects.

A 122x1213 dimensional feature set were included in the Emo-Soundscape dataset (Table 10). MIRToolbox (Lartillot et al., 2008) and YAAFE (Mathieu et al., 2010) were used for feature extraction. All audio clips have a sample rate of 44,100 Hz and a frame size of 23ms with 50% overlapping. A Hanning window was also applied on each frame before feature extraction. Using the "bag-of-frames" approach (Aucouturier et al., 2007), the mean and standard deviation of each feature were calculated to represent the long-term statistical distribution of the local spectral features.

A machine learning model was also trained for the purpose of serving as a baseline for future investigations. Support Vector Regression (SVR) is a widely used machine learning model in music and video emotion recognition as well as related affect computing tasks. The shuffle protocol and the leave-one-out protocol were used to validate the model. In the shuffle protocol, the dataset is shuffled 10 times, each time with 20% of the database randomly selected for testing while the rest is used for training. In the leave-one-out protocol, one clip is selected in each iteration for testing while the rest is used for training. Using these two validation protocols, they

have concluded that the SVR model is superior to previous models that uses a multiple linear regression model.

Further comparison on different models using the Emo-soundscape dataset was performed in 2018 (Fan et al., 2018), where five machine learning and deep learning architectures were compared and maximized in performance. The architectures were: fine-tuned convolutional neural network (CNN), CNN trained from scratch, long short-term memory recurrent neural networks (LSTM-RNN), LSTM-RNN trained from scratch, SVR, and transfer learning.

The fine-tuning strategy uses a pre-trained model with the last few layers replaced by newly trained layers that are dedicated to soundscape emotion recognition. As earlier layers are more generic, even though they are trained on tasks that are not related to emotion recognition, theoretically, they should be able to generalize to emotion recognition tasks. Later layers, however, require a more specific approach to the tasks at hand and will be less likely to generalize from models that are intended for other purposes. The authors took an existing VGG model that was fine-tuned from a model intended for video content classification to classify audio (Hershey et al. 2017) and further fine-tuned it to recognize emotion within soundscapes (VGGish). The last layer was replaced by a layer consisting of 64 fully connected neurons and one output layer to produce a score for the valence arousal prediction.

The CNN model consists of two convolutional layers and one output layer. The first convolutional layer consists of 256 neurons that filters the 54x30x1 input with eight kernels with a size of 5x5x1 and a stride of one and the second layer that uses eight size 3x3x8 kernels. Both convolutional layers use maxpooling (2x2) and a dropout rate of 0.15 for the output. The ReLu activation function was chosen for all the layers and the output layer uses a linear activation

function to predict the valence/ arousal score. The Xavier uniform was selected to initialize all the weights and the CNN was trained with the RMSProp optimizer with a batch size of 32 examples with a learning rate of 0.001 and a decay of 0.000001.

The LSTM-RNN model consists of two stacked LSTM units and a single output layer, with 128 neurons in each unit for the arousal model and 64 in each unit for the valence model. Tanh was selected as the activation function in the LSTM units and the linear activation for the output layer. The Xavier uniform was used to initialize all the weights and the model was trained with the RMSProp optimizer with a batch size of 32 examples with a learning rate of 0.001 and a decay of 0.000001.

The transfer learning model combines a VGGish model with a SVR model. Since VGGish embeddings are more compact than raw audio features, it is used as a feature extractor that converts raw audio into a 128-D embedding that is used as the input of the SVR model that was proposed in the Emo-soundscape dataset. The RBF kernel was selected for the SVR model, and the grid search method was used to find the C and gamma parameters.

R^2 and mean squared error (MSE) were used to evaluate the performance of all the models. The research concluded that while the CNN (trained from scratch) model performs the best at both R^2 (0.892) and MSE (0.035) in arousal, the fine-tuned model performs best at both R^2 (0.759) and MSE (0.078) with the CNN (trained from scratch) model being a close second (R^2: 0.712, MSE: 0.096) when it comes to valence.

2.2 Automatic Emotion Recognition System

While there was a plethora of machine learning models that have been trained in research papers, there is currently only one automatic emotion recognition system that has been made.

The Impress automatic soundscape affect classification system (Thorogood & Pasquier, 2013) is a system that uses supervised machine learning to take real time audio inputs and output soundscape affect feedbacks to the users. The system was designed to assist soundscape composers with their work, as there is an increasing amount of need for such tools that provide feedback on the affect of audio environments for soundscape composers.

The output of this soundscape emotion recognition system is a grid-like dimensional output that follows Russell's circumplex model of affect for criterion (Russell, J. A. 1980), with valence on the x-axis and arousal on the y-axis. As this is a model more suitable for fluctuating emotions compared to other research methods such as doing a survey, this has been adopted as the output of the model for the system this paper presents.

There are two stages for the Impress affect classification system. The first being the collection stage where audio and the user emotional response of the audio environment is collected. Audio analysis and feature extraction is done by extracting features on a four second audio buffer. Then the mean and standard deviation of the audio features are calculated to represent the signal. The extracted features are total loudness, perceptual spread, perceptual sharpness, and MFCC. The audio is recorded in AIF file format with a sample rate of 22,500 Hz, and a 23ms Hanning window with a 50% overlap was applied before the feature extraction stage. A visual representation of Russell's circumplex of affect which allows users to select points on the grid that match their emotions in the moment is then collected and used as the ground truth for each recording. In the second stage, the data collected from the previous stage is modeled using a multiple linear regression (MLR) model and the affect of any new recording is estimated.

Using a k-fold validation method on the MSE, the accuracy of the MLR model was evaluated. The valence model scored a 0.0392 while the arousal model scored a 0.0348 for their

respective MSE. The authors of the research noted that the models used for the prediction were not perfect prediction models and there might be other unknown variables or cultural factors such as automobiles and manmade noise that may have contributed to the result.

3. METHODOLOGY

Previous sections demonstrated several machine learning models that have been trained to perform emotional recognition tasks. As there have yet to be something readily available that allows users to test the plausibility of such products, especially in the context of estimating emotions of sound effects, here the paper presents an application that predicts soundscape emotions as a proof of concept that would allow users to experiment and possibly provides useful insight to the field.

As the purpose of the application is to assist with the productivity of searching sound effects with a certain emotion, the application would be following a simple design philosophy that allows the ease of use as well as the clarity of the presented information. The emotion recognition model should also be as accurate as the amount of data would allow it to be. In this section, the design and the approach of the application will be thoroughly explained and discussed.

1. Application Overview

The application is a desktop application designed to assist the search of sound effects that could be used alongside existing audio manipulating applications such as a digital audio workstation. By ingesting folders of audio files, the application would be producing predictions of emotions for each file and visualize the emotion distribution of the files on a 2-dimensional emotional space. Depending on the kind of sounds inside the folder, the application would produce a different visualization of the distribution on the emotional plane. Users would then be able to select the files based on the emotion predictions and listen to or copy the file onto their preferred audio working environments.

Structurally, the application is a one-page application with four major components under the hood. The file handling component, a feature extraction portion, the neural networks for predicting emotions, and the visualization/ user interaction part. The file handling component handles the interaction with the computer system as well as reading and storing audio data for future processing purposes. The feature extraction portion extract features from audio clips and store the feature statistics to a container of features. The models predict valence and arousal or the coordinates of the emotional plane, and the visualization/ interaction part allows users to see the distribution of their directory of audio clips as well as listen and drag the clips to other audio manipulation tools.

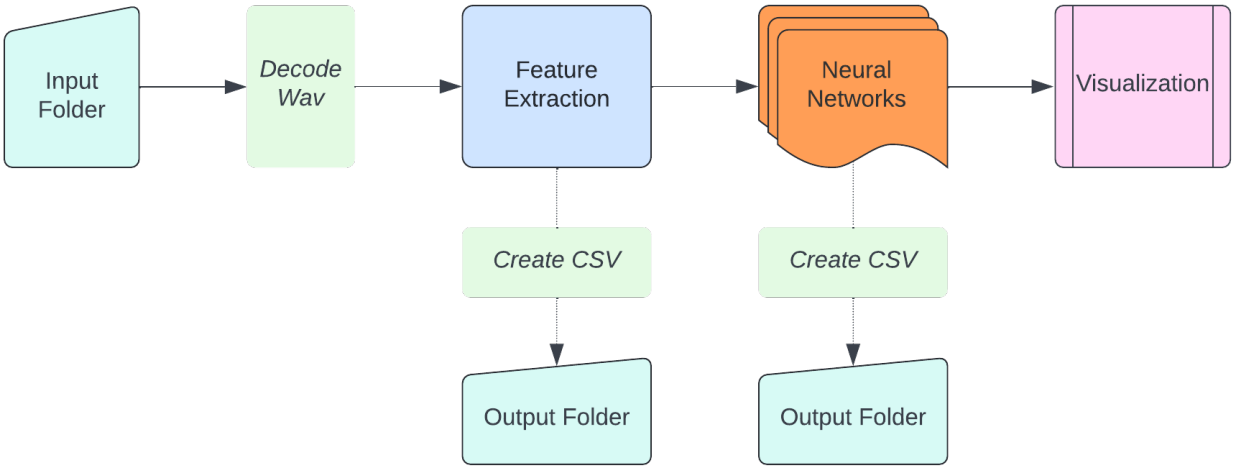

Figure 2: Application flow diagram

2. Materials and Methods

In this section, we will elaborate on the design as well as the process of creating the application. The entire application was created using Node.js while the neural network was written using Python. Node.js is a JavaScript runtime environment that allows JavaScript code to

be executed outside of a browser. As Node.js has a plethora of existing libraries that can be leveraged, it was chosen as the language to write in to allow faster prototyping and test the concept as soon as possible. Python on the other hand is a high-level interpreted programming language that is heavily used in the data science field. With several existing machine learning and data manipulation libraries in the language, Python was chosen so that the model can be built with ease.

2.1 Application Design

The entire application was written in Electron.js for its simplicity. Electron.js is a cross-platform JavaScript library that allows users to build desktop applications using the exact same tools as one would use when building a website, namely HTML, CSS, and JavaScript. It also allows the application to be converted into a web application in the future with the help of module bundlers such as Webpack or Browserfy to expand the user base and further the production of the application with relative ease.

The design of the interface is quite simple. As the purpose of this paper is to create a prototype, the focus of the design will be on the core functionality. The application consists of an application title, an audio player, a visualization area, a status update, and buttons for control (Figure 3). The simplistic design showcases the functionality of the application without bogging down the production process with too much additional functionality and design. The entire structure of the application is written in HTML with minimal CSS for styling.

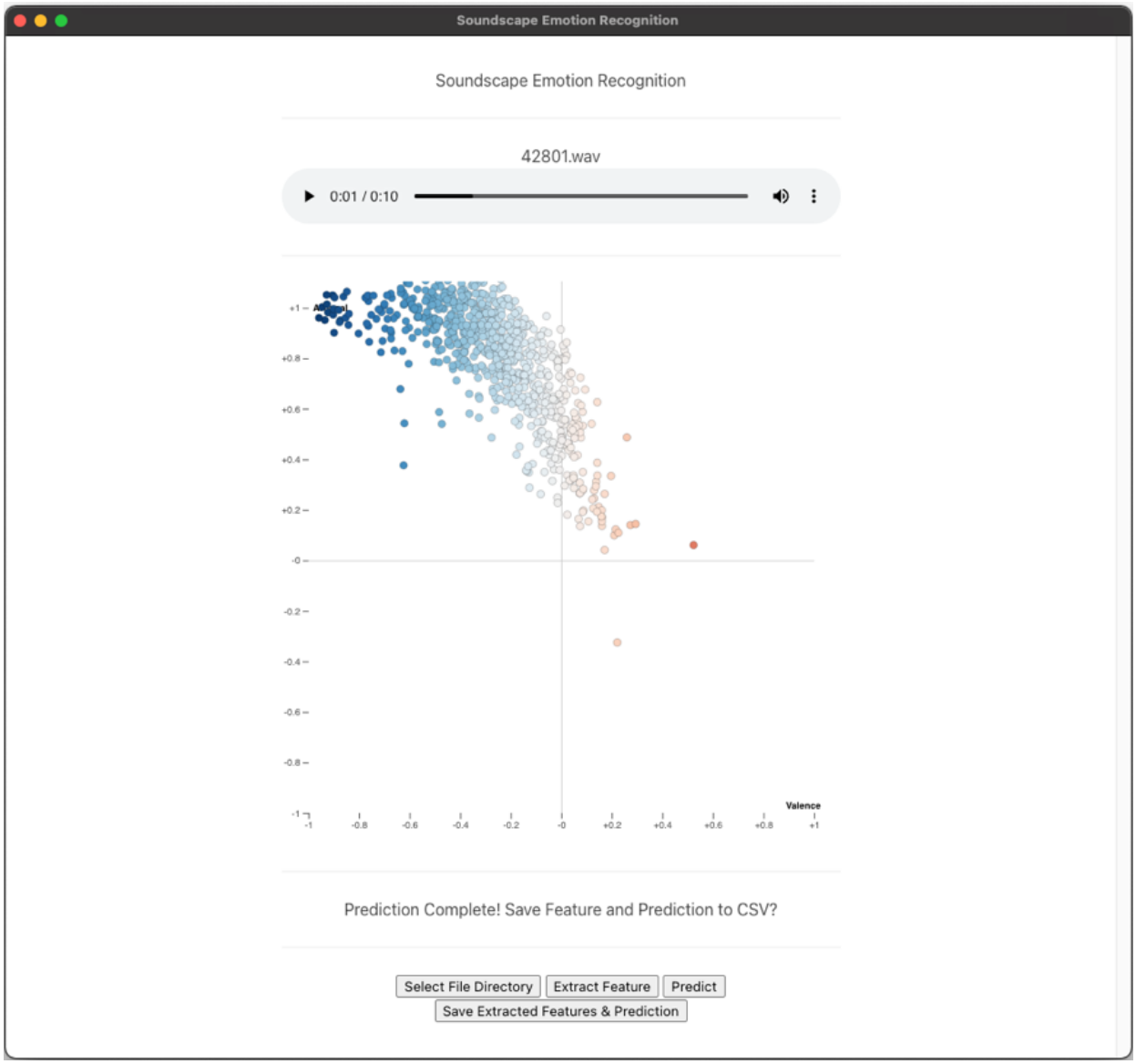

Figure 3: Application interface and emotional plot

2.2 File I/O

The entire logic and interaction of the application was written in JavaScript. The file in/ out functions work under the user interface when users select directories for predictions and when they save the extracted features and predictions into csv files. Electron.js and the native

Node.js file system module were used to handle the file read and write while the node-wav module was used to decode wav files into float 32 arrays.

When “Select File Directory” is selected, the *showOpenDialog* function is called and the applications opens a browser window for users to select a folder of wav files of their choice. All the file names inside the directory are saved into a file list object after the folder is selected and will later be called when doing feature extraction and visualization. The wav files are not decoded until the extract feature function is called, just before extracting the feature.

When “Save Extracted Features & Predictions” is selected, the *showSaveDialog* is called and the application opens browser windows for users to select folders to save the extracted features and predictions. A CSV file is created for the features and another for the predictions by calling the *createCSVContent* function before the application writes the file to the user selected paths.

2.3 Dataset

The dataset that was used to train the neural network is the Emo-soundscapes dataset (Fan et al. 2017). The dataset includes 1,213 six-second audio clips. 600 of which was curated from freesounds.org while the remaining 613 was created by artificially mixing the curated 600 sounds. The dataset includes 100 soundscapes from each of Schafer’s soundscape categories (Schafer, 1993): human, mechanical, nature, quiet, social, and indicator sounds. With the dataset also comes with a total of 29 precalculated audio features (Table 10). The features were calculated using the bag-of-frame approach to represent the long-term statistical distribution of the audio signal (Aucouturier et al., 2007), resulting in a 122-dimension audio feature set for the entire dataset.

Each of the audio clips in the dataset was labeled with corresponding valence and arousal scores. The scores were acquired from 1,182 annotators by performing pairwise comparison on audio clips using a ranking system. The rankings were then converted to ratings by mapping values from 1 to 1,213 to a corresponding +1 to -1 range.

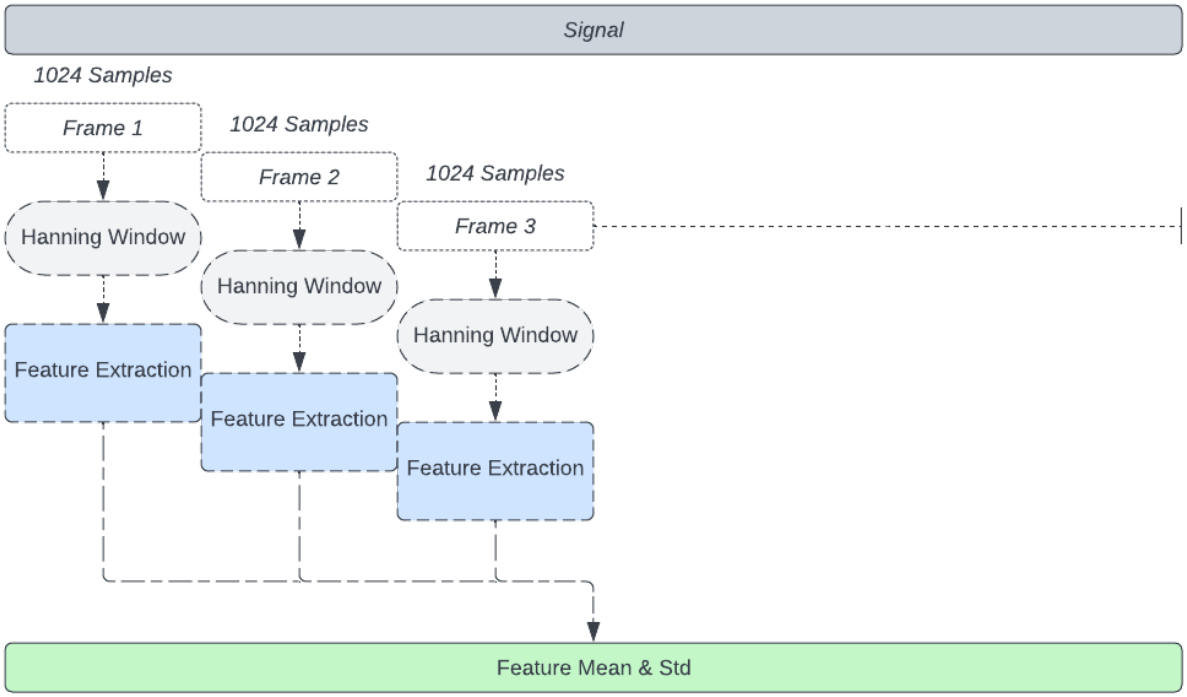

Figure 4: Feature extraction flow chart

2.4 Feature Extraction

Feature extraction will be performed on all the audio clips given to the application. Feature extraction is the task of extracting measured data from audio signals that represents information about the signal. While analog audio signals are sampled at high rates to capture the realism, the amount of computational power required to process the huge amount of data is very expensive. Features allow the signal to be represented in several mathematic values instead of its entirety while also reducing the amount of computational power required to process them.

Meyda.js (Rawlinson et al., 2015) is a JavaScript feature extraction library that works with Web Audio API in real time inside the browser or offline with Node.js, which is what this program is using. It allows audio features to be extracted synchronously by simply importing the library with a Meyda object and calling the extract function on a signal. The Meyda object has been configured to have a buffer size of 1,024 which will then be passed through a Hanning window before extracting features.

The following features are the feature set Meyda.js is able to extract. RMS, zero crossing rate, spectral roll off, spectral centroid, spectral spread, spectral skewness, spectral kurtosis, spectral flatness, MFCC, chromagram, loudness, energy, perceptual sharpness, and spectral slope. All of which will be extracted from audio clips to store into a feature array that contains all the extracted feature for one clip.

As the dataset used to train the model uses a bag-of-frame approach to represent the audio data in each sound file. We took the same approach to ensure the input dimensions of the model is identical to the dimension of the feature vector used for training. The bag-of-frame approach takes the feature extracted from the short-term audio frames and calculates the long-term statistical distribution of the entire signal. In this application, the entire audio signal of one clip will be divided into a number of 1,024 samples to perform feature extraction and then store into an array called *featureContainer*. We will then call the *getStats* function on the feature container to get the mean and standard deviation of the set of audio features. The resulting feature vector extracted from one audio clip is a JavaScript Float32Array with a dimension of 1x74. After calculating the statistics of the feature set, the resulting feature statistics will be pushed onto a 2-dimensional array that contains the feature set for all files called *allfilesFeatureStats* to prepare for the valence and arousal prediction model. As the training

dataset is normalized, each extracted features were also normalized to the largest value of their respective features of all the files in the directory.

One caveat or perhaps a bug of the Meyda.js library is the limitation of the length of the signal. It is typical to have each frame of the audio data to be a size of powers of two for faster transformation algorithms such as the Cooley-Tukey Fast Fourier transform algorithm. Yet the library would also require the entire signal length to be powers of two, any signal length that isn't a power of two will result in the failure of extracting some spectral domain feature. To resolve this issue, signals will be truncated before extracting features, resulting in a loss of valuable audio information.

2.5 Neural Network

Two neural networks have been trained to perform predictions on valence and arousal. The Google TensorFlow library is used to train and deploy both models. It is a deep learning library that allows programmers to build, save, and deploy models in Python and JavaScript. Both models were trained in Python, saved into a JSON file, and then deployed in JavaScript into the application.

As the purpose of this paper is to present a tool for assisting sound effects selection, the focus will be on building the tool itself and testing the possibilities of such tools using existing methods. Here we will be referring to previous research and models presented in Fan's soundscape emotion research (Fan et al., 2018) and build models that perform similar tasks. The previous presented models perform multiple linear regression on the input feature set by taking several features and predicting a valence or arousal rating ranging from +1 to -1. Here we built two simple neural networks using TensorFlow to perform the emotion predictions.

Both neural networks are identical in terms of structure. They are composed of two densely connected layers. The first dense layer is connected to the 74x1 input with 50 neurons. The second dense layer, connected to the first one, consists of 20 neurons. Both layers are connected to a dropout layer before connecting to the next layer, with a dropout rate of 0.2. The last layer, the output layer, is composed of just one neuron. The ReLU activation function were used in the first 2 dense layers and the linear activation function were used in the output layer to obtain the prediction score. Finally, the model was trained using the RMSprop optimizer and mean squared error loss function over 100 epochs. The models were saved into a Keras HDF5 file after the training, then converted to JSON format along with the binary files of the weights using the Tensorflow.js converter.

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                 Output Shape              Param #
=================================================================
 dense (Dense)                (None, 50)                3750

 dropout (Dropout)            (None, 50)                0

 dense_1 (Dense)              (None, 20)                1020

 dropout_1 (Dropout)          (None, 20)                0

 dense_2 (Dense)              (None, 1)                 21

=================================================================
Total params: 4,791
Trainable params: 4,791
Non-trainable params: 0
_________________________________________________________________
```

Figure 5: Arousal and valence neural network structure

To use the models for prediction, first the model is loaded as soon as the software is opened using the *loadLayerModel* function of the TensorFlow.js library. The array of extracted feature stats acquired from the previous feature extraction section are converted into a tensor then reshaped into the right input shape *[1, 74]* before passing it into the valence and arousal models for predictions. The scores from the arousal and valence models are then saved into a size 2 array for each of the files for visualization later.

2.6 Visualization

D3.js is a JavaScript library that allows the manipulation of the document object model (DOM) based on a given set of data. The Node.js version of D3.js were used in this application to create plot and handle user interactions with the plot. The emotion predictions of the audio files are visualized and presented to the user with a valence arousal plot. The valence/ arousal plot is a scatter plot with valence on the x-axis and arousal on the y-axis. Both valence and arousal ranges from -1 to +1.

The model predictions are converted to a JavaScript object before plotting onto a SVG element based on the valence and arousal scores. Each audio clip is represented with a dot on the plot. Higher arousal scores (y-axis) mean the higher the amount of activity is in a clip and vice versa. Valence score relates to how the sounds place on the negative to positive emotion spectrum, with negative emotion being closer to -1 and positive emotion closer to +1. An additional touch for the valence is the mapping of color from blue to red corresponding to the range of -1 to +1 (Figure 3).

Users are limited to two types of interaction with the visualization: mouse over the dots and clicking the dots. While the mouse hovers over the dots, the dots increase in size and shows

the name of the corresponding audio file name. Upon moving the mouse out of the dot, the dot returns to its normal size. Users are also able of clicking on the dot to load the corresponding audio file into the HTML audio player to listen to the file and decide whether the file fits into their needs.

4. EVALUATION

1. Objective Evaluation: Model Evaluation

An objective evaluation was conducted to analyze the performance of the neural network. As the Emo-soundscapes dataset consists of 1,213 clips of 122-dimension feature vector, including features the Meyda.js library was not able to extract. The performance of the presented models had to be evaluated to assure its validity. K-fold cross-validation were used as the evaluation method to get the summary statistics of all models. As the models are all linear regression models, mean and standard deviation of the mean squared error (*MSE*) and the coefficient of determination (R^2) are used as the evaluation metrics.

1.1 Dataset

The Emo-soundscapes dataset were used to validate as well as train the proposed arousal and valence neural network. The dataset consists of 1,213 6-second audio clips for the entire dataset and a 122-dimension precalculated feature statistics as well as an arousal and valence ground truth rating for each audio clip which will be used during validation and training. Modifications of the original dataset were required to fit into the available tools of the application. As the features Meyda.js can extract are limited compared to that of the Emo-soundscapes dataset (74-dimension as opposed to 122-dimension), the unavailable features in the dataset were entirely stripped, resulting in a 74-dimension feature vector for 1,213 audio clips.

1.2 Machine Learning Models

Support Vector Regressions

Two baseline models (one for arousal, another for valence) was created by recreating the Support Vector Regression (SVR) model presented in the Emo-soundscapes dataset (Fan et al., 2017) to evaluate the effects of reducing the feature vector of the model input. The SVRs were implemented in Python using the scikit-learn package. All SVR models had the kernel set to the RBF kernel and the gamma set to auto. As the features the Meyda.js library is only able to extract a 74-dimension feature vector, a baseline SVR model of 122-dimension input were created to compare the difference and effects of reducing the input feature vector.

Neural Networks

The neural networks presented in this paper were also evaluated against the SVR models. Two models of 122-dimension input (one for arousal, another for valence) and two models of 74-dimension input were evaluated against its SVR counterpart to assess the neural networks.

1.3 Evaluation Metrics

Mean Squared Error (MSE)

Mean Squared Error or mean squared deviation is the average squared distance from the estimated points to the ground truth. It is a commonly used risk function in a linear regression model to evaluate the quality of a machine learning model. Due to randomness of the distribution of the data point, the mean squared error will most likely be larger than zero, hence one of the most used metrics of estimating a linear regression model is to minimize the *MSE*. As the model presented in this paper is a multiple linear regression model, mean squared error will be used as one of the metrics to validate the performance of the model. *MSE* is calculated in the following equation.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(Y_i - \widehat{Y}_i\right)^2$$

Coefficient of Determination (R^2)

Coefficient of Determination is the ratio of the variance of the dependent variable (y) that is predicted from the independent variable (x). It is a statistical measure to evaluate the ability of a linear regression model. R^2 scales from 0% to 100% and, although not always the case as we will discuss in the following section, a higher R^2 value indicates a better predicting ability of a model. R^2 is calculated in the following equation.

$$\boldsymbol{R^2} = \frac{\boldsymbol{model\ sum\ of\ squares\ (MSS)}}{\boldsymbol{total\ sum\ of\ squares\ (TSS)}}$$

$$= \frac{\boldsymbol{TSS - residual\ sum\ of\ squares\ (RSS)}}{\boldsymbol{TSS}}$$

$$= \mathbf{1} - \frac{\frac{1}{n}\sum_{i=1}^{n}\left(Y_i - \widehat{Y}_i\right)^2}{\frac{1}{n}\sum_{i=1}^{n}(Y_i - \overline{Y}_i)^2}$$

$$= \mathbf{1} - \frac{\boldsymbol{MSE}}{\boldsymbol{Var(y)}}$$

1.4 K-Fold Cross-Validation

K-fold cross-validation is a statistical procedure used to evaluate the performance of machine learning models on a set of data samples. Cross-validation is mainly used to evaluate the performance of a machine learning model when presented with unseen datasets. The k-fold cross-validation procedure has a single parameter of k that splits the dataset into k number of training and testing groups.

To evaluate the baseline models and the neural networks, the dataset was split into 10 groups of data samples, where 90% of the samples were used for training and 10% of the samples were used to test the performance of the model. The statistics (mean and standard deviation) of the *MSE* and R^2 in each fold were then used to validate the performance of each model.

1.5 Performance Analysis

Validation Results

In general, the neural networks and the recreated SVR models performed similarly in R^2 for both arousal and valence, though both models performed worse than previously presented models during the validation process. We have also discovered that reducing the input feature dimension didn't worsen the performance of the models by any significant amount. When training with the entire dataset, however, the proposed neural networks performed comparatively or slightly better than previously presented models (Fan et al., 2018).

Table 1 and 2 presents the 10-fold validation averaged result of the arousal and valence SVR model with the 122-dimension input from the dataset. The model performed worse than the model presented in the Emo-soundscapes dataset (Fan et al., 2017), demonstrating around 79.4% and 56.3% in R^2 for arousal and valence respectively while the previous model using multiple linear regression demonstrated 85.3% and 62.3% in R^2 for arousal and valence respectively.

Arousal (avg)	Mean	Std
R^2	0.7937	0.0314
MSE	0.0673	0.0093

Table 1: Average prediction results for SVR arousal (122-dimension)

Valence (avg)	Mean	Std
R^2	0.5625	0.0483
MSE	0.1433	0.0142

Table 2: Average prediction results for SVR valence (122-dimension)

Table 3 and 4 presents the average validation results of the arousal and valence model with a reduced amount of feature (74-dimension feature vector). Interestingly, reducing the input feature vector of the to a 74-dimension feature vector, which is around 60% of the original input dimension, only slightly decreased the R^2 score by a marginal amount when compared to the 122-dimension input model. The 74-dimension input SVR model had roughly 78.6% and 56.3% R^2 on arousal and valence respectively.

Arousal (avg)	Mean	Std
R^2	0.7856	0.0261
MSE	0.0711	0.0097

Table 3: Prediction results for SVR arousal (74-dimension)

Valence (avg)	Mean	Std
R^2	0.5473	0.0515
MSE	0.1489	0.0117

Table 4: Prediction results for SVR valence (74-dimension)

Table 5 and 6 presents the averaged 10-fold validation results of the 122-dimension input neural network. The model demonstrated around 72.3% and 49.5% R^2 score in arousal and valence respectively. Compared to the recreated SVR models, the neural networks only performed slightly worse, which is just slightly lower than the corresponding SVR models.

Arousal (avg)	Mean	Std
R^2	0.7228	0.2351
MSE	0.0689	0.0332

Table 5: Prediction results for Neural Network arousal (122-dimension)

Valence (avg)	Mean	Std
R^2	0.4946	0.1732
MSE	0.1372	0.0304

Table 6: Prediction results for Neural Network valence (122-dimension)

Table 7 and 8 shows the results of the 74-dimension input model. The respective average R^2 score for arousal and valence is around 73.1% and 50.4%. An interesting note on the result is that the neural networks of the reduced feature dimension input performed slightly better than that of the full 122-dimension input model which we will be discussing in a later section.

Arousal (avg)	Mean	Std
R^2	0.7309	0.1618
MSE	0.0707	0.0285

Table 7: Prediction results for Neural Network arousal (74-dimension)

Valence (avg)	Mean	Std
R^2	0.5042	0.1214
MSE	0.1384	0.0383

Table 8: Prediction results for Neural Network valence (74-dimension)

Compared to the SVR models, the neural networks performed only slightly lower during the k-fold validation process, although both approaches are still lower than that of the one presented in the Emo-soundscapes paper, the results validated the decision of reducing the input

feature dimension to fit the available features in Meyda.js. This also validates the performance of the proposed neural networks.

Training Results

Table 9 presents the final training metrics of the neural networks after 100 epochs. All 1,213 sound clips were used to train the model. Compared to the validation results, there was a significant increase in both arousal and valence. When compared to the SVR models present in Emo-soundscapes (Fan et al., 2017) and that of previous deep learning methods (Fan et al., 2018), the neural networks this paper presents performed better than the SVR models and performed comparatively to their VGGish models, convolutional neural networks (CNN), and long short-term memory recurrent neural networks (LSTM-RNN) in both arousal and valence.

	Arousal	Valence
R^2	0.8684	0.7281
MSE	0.0438	0.0906

Table 9: Training results from Neural Network

2. Subjective Evaluation

2.1 Participants

To evaluate the accuracy and the usefulness of the application, verbal observations were made by asking for expert opinions to two individual audio professionals. The author of this paper also participated in the evaluation of the application. All participants of the subjective evaluation have at least three or more years of experience within the field of film audio post-production.

2.2 Method

The participants were given a demonstration of the application and a couple follow up questions about the application. The demonstration consists of predicting the soundscape emotions in a folder of audio files followed by playing the audio clips on the scatter plot of the audio clips inside the application (as shown in figure 3) to demonstrate the relation between the audio content and the predicted emotion. Participants are free to listen to the audio clips multiple times to compare different clips and evaluate the consistency of the prediction. In addition, participants were also given an explanation of the V-A plot in case they were not familiar with the terminology.

Participants were asked questions on their opinion of the application, including the accuracy of the predictions for arousal and valence as well as their feedback on the usability of the application. Questions for the accuracy of predictions are as follows:

a. Please describe how accurate is the prediction for arousal?
b. Please describe how accurate is the prediction for valence?
c. Please describe the usefulness of the application in terms of integrating this with your existing workflow and toolset.

2.3 Results

Early evaluation results shows that the neural network performed accurately on the predictions for arousal. In general, participants agree that the higher the value of the arousal, the

more activity is present within the content of the audio clip. However, participants reported a high correlation between loudness and arousal as well as a high correlation between noise content and arousal.

Valence on the other hand, didn't performed as expected for most participant. While there were some predictions that were consistent with the audio content, most participants reported that they do not agree with the predictions of the model on the valence scale.

In terms of the usability of the application. All participants agree that integrating the program into a typical production workflow would be immensely useful as it would reduce the amount of time used on searching for sound effects. They also agree that utilizing artificial intelligence to assist with their current production process would be incredibly helpful.

5. DISCUSSION

The performance of the neural networks was presented in the evaluation section and the results were compared to previous models. Training results showed comparable results to previous models while validation results were lower than previous models. A subjective expert opinion was also collected to evaluate the usefulness of the presented application.

While the arousal model predicts activity inside audio clips accurately, the valence model, however, is far from perfect. Interestingly, in certain datasets, the valence and arousal plot were extremely skewed towards the first and second quadrants. When listening to the audio from the visualization interface, arousal have been consistent in predicting the amount of activity in the file, while valence seems to be quite arbitrary. This may be due to the lack of consensus in valence as negative or positive emotion is usually quite subjective. There also might be other independent variables affecting valency that was not taken into consideration.

The small number of users who have tested the application have given positive feedback on the usability. While existing tools such as Basehead, Soundminer, and Soundly are great for integration with local audio libraries, the process of searching for the right sound effects still requires the sound professional to possibly listen through every sound effect, with zero to no assistance from current technology. One of the complaints on current toolsets is the crudeness of current tools for searching sound effects. There is also a lack of feedback from the sound professional for the sound effects that have been used, such as automatically tagging the audio clips with the region of audio being used and context of said use. The application this paper presents, thus, serve as a good starting place to develop such kinds of tools.

6. CONCLUSION & FUTURE WORK

In this study, a soundscape emotion predicting system is presented using simple neural networks to predict the arousal and valence scores for the circumplex of affect. This work promises to aid in speeding up the audio selection process in audio productions. Sound professionals and artists adopting this program will benefit by having a machine provide emotional evaluation to sound files in addition to the original file names and descriptions. There is also the added benefit of being able to look for sounds that are similar in mood, leading to the increased possibility of finding the right sound for the work they are currently doing.

While this study uses regression models as arousal and valence predictors and the resulting application serves as a promising starting point for implementing such tools in the workflows of sound professionals, increasing the robustness of the model would be the logical next step of improving the accuracy of the application. Incorporating different methods of predicting emotions, specifically valence, would improve the accuracy and functionality of the application significantly. A feasible approach would be to classify sounds in the audio clip. As sounds tend to have existing association with certain emotions, using classification as an additional source (weights in the neural networks) for predicting valence would be beneficial to the overall accuracy of the system. For example, bird chirps tend to make people calm and peaceful, hence it ranks higher on valence. By further increasing the ways of measuring valence, the model would be even better at predicting affect, thus increasing the possibility of being able to allow sound professionals to incorporate this tool into their production workflow.

REFERENCES

Aucouturier, J. J., Defreville, B., & Pachet, F. (2007). The bag-of-frames approach to audio pattern recognition: A sufficient model for urban soundscapes but not for polyphonic music. The Journal of the Acoustical Society of America, 122(2), 881-891.

Fan, J., Thorogood, M., & Pasquier, P. (2016). Automatic soundscape affect recognition using a dimensional approach. Journal of the Audio Engineering Society, 64(9), 646-653.

Fan, J., Thorogood, M., & Pasquier, P. (2017, October). Emo-soundscapes: A dataset for soundscape emotion recognition. In 2017 Seventh international conference on affective computing and intelligent interaction (ACII) (pp. 196-201). IEEE.

Fan, J., Tung, F., Li, W., & Pasquier, P. (2018). Soundscape emotion recognition via deep learning. Proceedings of the Sound and Music Computing.

Four Fundamental Shifts in Media & Advertising During 2020. (2020, September 23). DoubleVerify. https://doubleverify.com/four-fundamental-shifts-in-media-and-advertising-during-2020/

Global Film and Video Services Market Report 2021 - Opportunities and Strategies to 2030 - ResearchAndMarkets.com. (2021, September 10). Business Wire. https://www.businesswire.com/news/home/20210910005333/en/Global-Film-and-Video-Services-Market-Report-2021---Opportunities-and-Strategies-to-2030---ResearchAndMarkets.com

Hershey, S., Chaudhuri, S., Ellis, D. P., Gemmeke, J. F., Jansen, A., Moore, R. C., ... & Wilson, K. (2017, March). CNN architectures for large-scale audio classification. In 2017 ieee international conference on acoustics, speech and signal processing (icassp) (pp. 131-135). IEEE.

Instagram. (2010). [Software]. Meta. https://www.instagram.com/

Kim, Y. E., Schmidt, E. M., Migneco, R., Morton, B. G., Richardson, P., Scott, J., ... & Turnbull, D. (2010, August). Music emotion recognition: A state of the art review. In Proc. ismir (Vol. 86, pp. 937-952).

Krippendorff, K. (1970). Estimating the reliability, systematic error and random error of interval data. Educational and Psychological Measurement, 30(1), 61-70.

Lartillot, O., Toiviainen, P., & Eerola, T. (2008). A matlab toolbox for music information retrieval. In Data analysis, machine learning and applications (pp. 261-268). Springer, Berlin, Heidelberg.

Mathieu, B., Essid, S., Fillon, T., Prado, J., & Richard, G. (2010, August). YAAFE, an Easy to Use and Efficient Audio Feature Extraction Software. In ISMIR (pp. 441-446).

Music Technology Group at Universitat Pompeu Fabra. (n.d.-b). Freesound. Freesound. Retrieved April 25, 2022, from https://freesound.org/

Rawlinson, H., Segal, N., & Fiala, J. (2015, January). Meyda: an audio feature extraction library for the web audio api. In The 1st web audio conference (WAC). Paris, Fr.

Russell, J. A. (1980). A circumplex model of affect. Journal of personality and social

psychology, 39(6), 1161.

Schafer, R. M. (1993). The soundscape: Our sonic environment and the tuning of the world. Simon and Schuster.

Thorogood, M., & Pasquier, P. (2013, May). Impress: A Machine Learning Approach to Soundscape Affect Classification for a Music Performance Environment. In NIME (pp. 256-260).

Thorogood, M., Fan, J., & Pasquier, P. (2016). Soundscape audio signal classification and segmentation using listeners perception of background and foreground sound. Journal of the Audio Engineering Society, 64(7/8), 484-492.

TikTok. (2016). [Software]. ByteDance. https://www.tiktok.com/

Tomé, J., & Cardita, S. (2021, December 20). In 2021, the Internet went for TikTok, space and beyond. The Cloudflare Blog. https://blog.cloudflare.com/popular-domains-year-in-review-2021/

LIST OF TABLES

Rms	Spectral Brightness	Spectral Entropy	Chromagram (centered) 12	Loudness
Decrease Slope	Spectral Centroid	MFCC 13	Key (tonal center)	Energy
Spectral Fluctuation	Spectral Spread	Spectral Flux	Modality	Perceptual Sharpness
Event density	Spectral Skewness	Low Energy Rate	Harmonic change detection function (HCDF)	Spectral slope
Zero Crossing Rate	Spectral Kurtosis	Novelty	Inharmonicity	Spectral Variation
Spectral Roll Off	Spectral Flatness	Pitch	Chromagram 12	

Table 10: Audio features in Emo-soundscape

Rms	Spectral Centroid	Spectral Kurtosis	Chromagram 12	Perceptual Sharpness
Zero Crossing Rate	Spectral Spread	Spectral Flatness	Loudness	Spectral slope
Spectral Roll Off	Spectral Skewness	MFCC 13	Energy	

Table 11: Audio features available in Meyda.js

SVR 10-fold Cross Validation (122-dimension)

SVR Arousal R^2 10-fold Cross Validation									
1	2	3	4	5	6	7	8	9	10
0.8157	0.7544	0.8049	0.8339	0.8432	0.7500	0.7694	0.8153	0.7752	0.7752

Table 12: 10-fold cross validation R^2 score for SVR arousal (122-d)

SVR Arousal *MSE* 10-fold Cross Validation									
1	2	3	4	5	6	7	8	9	10
0.0571	0.0752	0.0584	0.0600	0.0549	0.0839	0.0717	0.0672	0.0665	0.0779

Table 13: 10-fold cross validation MSE score for SVR arousal (122-d)

SVR Valence R^2 10-fold Cross Validation									
1	2	3	4	5	6	7	8	9	10
0.6109	0.6002	0.5616	0.4616	0.5710	0.6170	0.5427	0.5185	0.6149	0.5270

Table 14: 10-fold cross validation R^2 score for SVR valence (122-d)

SVR Valence *MSE* 10-fold Cross Validation									
1	2	3	4	5	6	7	8	9	10
0.1438	0.1283	0.1400	0.1789	0.1512	0.1229	0.1381	0.1464	0.1410	0.1426

Table 15: 10-fold cross validation MSE score for SVR valence (122-d)

Neural Network 10-fold Cross Validation (122-dimension)

Neural Network Arousal R^2 10-fold Cross Validation									
1	2	3	4	5	6	7	8	9	10
0.8048	0.8466	0.8163	0.8680	0.8495	0.7629	0.8431	0.7566	0.0446	0.6356

Table 16: 10-fold cross validation R^2 score for Neural Network arousal (122-d)

Neural Network Arousal *MSE* 10-fold Cross Validation									
1	2	3	4	5	6	7	8	9	10
0.0481	0.0427	0.0393	0.0338	0.0379	0.0939	0.0638	0.0837	0.1287	0.1176

Table 17: 10-fold cross validation MSE score for Neural Network arousal (122-d)

Neural Network Valence R^2 10-fold Cross Validation									
1	2	3	4	5	6	7	8	9	10
0.6618	0.6191	0.6106	0.5009	0.6870	0.3480	0.3253	0.5995	0.1142	0.4796

Table 18: 10-fold cross validation R^2 score for Neural Network valence (122-d)

Neural Network Valence *MSE* 10-fold Cross Validation									
1	2	3	4	5	6	7	8	9	10
0.1143	0.1062	0.1268	0.1320	0.0895	0.1396	0.1756	0.1616	0.1941	0.1324

Table 19: 10-fold cross validation MSE score for Neural Network valence (122-d)

SVR 10-fold Cross Validation (74-dimension)

SVR Arousal R^2 10-fold Cross Validation									
1	2	3	4	5	6	7	8	9	10
0.8149	0.7969	0.7978	0.7780	0.7805	0.8260	0.8055	0.7627	0.7455	0.7478

Table 20: 10-fold cross validation R^2 score for SVR arousal (74-d)

SVR Arousal *MSE* 10-fold Cross Validation									
1	2	3	4	5	6	7	8	9	10
0.0554	0.0672	0.0675	0.0839	0.0803	0.0545	0.0703	0.0790	0.0791	0.0735

Table 21: 10-fold cross validation MSE score for SVR arousal (74-d)

SVR Valence R^2 10-fold Cross Validation									
1	2	3	4	5	6	7	8	9	10
0.4863	0.5777	0.5878	0.5600	0.4668	0.5738	0.5645	0.4607	0.6008	0.5943

Table 22: 10-fold cross validation R^2 score for SVR valence (74-d)

SVR Valence *MSE* 10-fold Cross Validation									
1	2	3	4	5	6	7	8	9	10
0.1526	0.1425	0.1440	0.1517	0.1726	0.1386	0.1312	0.1646	0.1491	0.1422

Table 23: 10-fold cross validation MSE score for SVR valence (74-d)

Neural Network 10-fold Cross Validation (74-dimension)

Neural Network Arousal R^2 10-fold Cross Validation									
1	2	3	4	5	6	7	8	9	10
0.7486	0.8482	0.8240	0.8699	0.7778	0.6792	0.8182	0.7485	0.2762	0.7187

Table 24: 10-fold cross validation R^2 score for Neural Network arousal (74-d)

Neural Network Arousal *MSE* 10-fold Cross Validation									
1	2	3	4	5	6	7	8	9	10
0.0619	0.0422	0.0377	0.0332	0.0560	0.1270	0.0739	0.0865	0.0975	0.0908

Table 25: 10-fold cross validation MSE score for Neural Network arousal (74-d)

Neural Network Valence R^2 10-fold Cross Validation									
1	2	3	4	5	6	7	8	9	10
0.6194	0.5983	0.7083	0.5130	0.6302	0.3913	0.3917	0.4056	0.3310	0.4528

Table 26: 10-fold cross validation R^2 score for Neural Network valence (74-d)

Neural Network Valence *MSE* 10-fold Cross Validation									
1	2	3	4	5	6	7	8	9	10
0.1286	0.1120	0.0950	0.1288	0.1057	0.1303	0.1583	0.2397	0.1466	0.1392

Table 27: 10-fold cross validation MSE score for Neural Network valence (74-d)