PROPOSAL TUGAS AKHIR – IF184702

Parallelizing CNN and Transformer Encoders for Audio Based Emotion Recognition in English Language

Adam Satria Adidarma

NRP 05111942000001

Dosen Pembimbing

Shintami Chusnul Hidayati, S.Kom., M.Sc., Ph.D

NIP 1987202012004

Program Studi S1 Teknik Informatika

Departemen Teknik Informatika

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

Surabaya

2022

LEMBAR PENGESAHAN

Klasifikasi Emosi Manusia untuk Audio Berbahasa Inggris Menggunakan CNN dan Encoder Transformer dengan Teknik Pararel

PROPOSAL TUGAS AKHIR

Diajukan untuk memenuhi salah satu syarat

Memperoleh gelar Sarjana Komputer pada

Program Studi S-1 Teknik Informatika

Departemen Teknik Informatika

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

Oleh : **Adam Satria Adidarma**

NRP. 05111942000001

Disetujui oleh Tim Penguji Proposal Tugas Akhir:

1. Shintami Chusnul Hidayati, S.Kom., M.Sc., Ph.D Pembimbing

SURABAYA
December, 2022

APPROVAL SHEET

Parallelizing CNN and Transformer Encoders for Human Emotion Classification for Audio Based Emotion Recognition in English Language

FINAL PROJECT PROPOSAL

Submitted to fulfill one of the requirements

for obtaining a degree Bachelor of Computer Science at

Undergraduate Study Program of Informatics

Department of Informatics

Faculty of Intelligent Electrical and Informatics Technology

Institut Teknologi Sepuluh Nopember

By: **Adam Satria Adidarma**

NRP. 05111942000001

Approved by Final Project Proposal Examiner Team:

1. Shintami Chusnul Hidayati, S.Kom., M.Sc., Ph.D Advisor

SURABAYA

December, 2022

Klasifikasi Emosi Manusia untuk Audio Berbahasa Inggris Menggunakan CNN dan Encoder Transformer dengan Teknik Pararel

Nama Mahasiswa / NRP : Adam Satria Adidarma / 05111942000001
Departemen : Teknik Informatika FTEIC- ITS
Dosen Pembimbing : Shintami Chusnul Hidayati, S.Kom., M.Sc., Ph.D

ABSTRAK

Kecerdasan artifisial telah berdampak signifikan pada berbagai industri dan sektor masyarakat, dengan adopsi kecerdasan artifisial yang tumbuh 37% dari 2018 hingga 2019, menurut laporan Gartner. Pengenalan emosi bicara (PEB) adalah subbidang kecerdasan artifisial yang fokus pada mengenali aspek emosional manusia saat berbicara, terpisah dari konten semantik. Emosi berperan penting dalam komunikasi manusia dan telah menjadi objek penelitian yang semakin meningkat dalam beberapa tahun terakhir. Meskipun studi saat ini tentang deteksi emosi sering memfokuskan pada modalitas visual, seperti ekspresi wajah, emosi adalah konsep multimodal yang membutuhkan studi terhadap indikator visual, taktil, vokal, dan fisiologis. PEB dapat diterapkan dalam berbagai konteks, termasuk pusat panggilan, pendidikan, pemasaran, psikologi, dan kesehatan. Studi ini mengusulkan pendekatan untuk menerapkan sistem PEB menggunakan model *Convolution Neural Network* (CNN) yang bekerja pararel dengan jaringan encoder *Transformer* dan mengevaluasi performanya pada dataset audio RAVDESS berbahasa Inggris. Model yang diusulkan akan dibandingkan dengan berbagai arsitektur pembelajaran mesin dalam hal performa, termasuk CNN biasa dan *Support Vector Machine* (SVM), untuk menentukan pendekatan yang paling efektif untuk PEB.

Kata kunci: Kecerdasan Artifisial, CNN, Transformer, PEB, *Self-Attention*, Audio.

Parallelizing CNN and Transformer Encoders for Human Emotion Classification for Audio Based Emotion Recognition in English Language

Student Name / NRP : Adam Satria Adidarma / 05111942000001
Department : Teknik Informatika FTEIC- ITS
Advisor : Shintami Chusnul Hidayati, S.Kom., M.Sc., Ph.D

ABSTRACT

Artificial intelligence (AI) has had a significant impact on various industries and sectors of society, with the adoption of AI growing 37% from 2018 to 2019, according to a Gartner report. Speech emotion recognition (SER) is a subfield of AI that focuses on recognizing the emotional aspects of speech, separate from the semantic content. Emotions play a crucial role in human communication and have been the subject of increasing research in recent years. While current studies on emotion detection often focus on visual modalities, such as facial expressions, emotion is a multimodal concept that requires the study of visual, tactile, vocal, and physiological indicators. SER can be applied in various contexts, including call centers, education, marketing, psychology, and healthcare. This study proposes an approach to implement an SER system using a parallelized Convolutional Neural Network (CNN) model and Transformer encoder network and evaluate its performance on the RAVDESS English audio dataset. The proposed model will be compared to various machine learning architectures in terms of performance., including standard Convolutional Neural Networks (CNN) and the Support Vector Machine (SVM), to determine the most effective approach for SER.

Keywords: AI, CNN, Transformer, SER, Self-Attention, Audio.

Table of Contents

Table of Figures

Chapter I
Introduction

In this chapter, the research background and context will be examined, including the problem being addressed, the scope of the problem, and the purpose and potential benefits of the research being conducted.

1.1 Background

Artificial Intelligence has emerged in every industry and has a profound impact on every sector of human society. According to Gartner Report (Costello, 2019), artificial intelligence adoption has grown 37% during 2018-2019 because the capabilities of artificial intelligence have matured significantly over the years leading to the adoption of this technology by enterprises around the world. Speech Emotion Recognition (SER) is one of the emerging applications in the context of artificial intelligence. SER is the task of recognizing the emotional aspects of speech independently over the semantic content. Humans can efficiently perform this task as a natural part of our communication, but the ability to do it automatically using a programmable device is still a subject of research (Lech, Stolar, Best, & Bolia, 2020).

In the book of The Media Equation (Ivar, Byron, & Clifford, 1996), Studies in human-computer interaction made the discovery that people often interact with computers as if they were other people and react to similar feedback from humans. Most of these social aspects ranging from politeness to reciprocity have been observed in human-computer interactions. Computer scientists believed that emotions and machines should connect in order to have better and more effective communication. Both data-driven reasoning and emotional perception are crucial for a machine's intelligence (Cowie, 2001). Giving machines emotional intelligence, the general user experience, and machine performance will be improved.

Emotions play a big role in human communication. Over the past years, research to understand human emotions was increasing (Jarymowicz & Maria, 2012). There are already a variety of computer systems that uses emotional speech classification as security systems, psychology and computer vision applications, and interactive computer designs. Current studies on emotion detection mainly focus on visual modalities, including facial expressions, muscle movements, hand posture, body posture, *etc.* (Keltner, Dacher, & Cordaro, 2017). However, emotion is a multimodal concept, and the task to detect emotions requires interdisciplinary studies that include visual modality, tactile communication, vocalization, and physiological indications (Heredia, Cardinale, Dongo, & Díaz-Amado, 2021).

A speech recognition system's success depends on the selection of a speech multimodal database, the extraction of pertinent features, and the selection of an effective classification algorithm. In the aforementioned works, emotion detection using audio data was chosen because it can be applied to various computer application system that doesn't require visual modalities, such as emotion detection on call center services to analyze customer habits to help improve the quality of service for the provider through sounds. Emotion detection based on audio data can also help learning experience in the field of education to help improve students' mental health by monitoring their emotions through sound. This system can also be used across various applications, such as marketing, psychology, health care, *etc*.

Emotion classification and sound detection using multiple SVM methods, such as linear and nonlinear, have received significant interest recently (Sonawane, Inamdar, & Bhangale, 2017). Some studies also tried to improve the accuracy of this method by using transfer learning on pre-trained deep learning models (Latif, Rana, Younis, Qadir, & Epps, 2018). Their results showed that deep learning-based CNN methods outperformed the handcrafted feature-

based SVM method in image classification (Younghak Shin, 2017). This is because deep learning methods learn categories incrementally through its hidden layer architecture, defining low-level categories first, and then moving to the higher-level categories. The number of datasets can also be a factor in improving the quality of this CNN method (Mahapatra, 2018). In addition, some robust deep learning architectures such as GPT-3 & BERT (Brown, et al., 2020) (Devlin, Chang, Lee, & Toutanova, 2019) are emerging to solve sequential learning problems based on a self-attention mechanism in the Transformer Network (Vaswani, et al., 2017). These architectures are now considered a state-of-the-art technique in the field of NLP (Natural Language Processing).

Based on the above statement, this study aims to implement a deep learning-based CNN method in parallel with a self-attention mechanism Transformer encoder network in the process of detecting human emotions with an English audio dataset. With the application of this method, it is hoped that this model could provide an expressive feature representation with the lowest computational cost by extending the CNN filter channel size and reducing the feature maps, while the Transformer encoder is used for the network to learn how to predict the frequency distribution of different emotions according to the overall structure of the MFCC plot of each emotion.

1.2 Problem Statement

From the background stated previously, the problem statement can be expressed as follows:

- How to detect human emotions from audio data with proposed method?
- Which classification methods are more accurate to detect emotion through audio between SVM, CNN, and the proposed method?
- How to build the model architecture to give a good accuracy?

1.3 Problem Scope

In order to stay true to the issues raised above, this paper includes a number of constraints. The problem in this paper has the following limitations:

- Audio data is in English;
- In one voice of the dataset, there is only one emotion;
- The model can only distinguish between the eight emotions of happy, neutral, sad, calm, angry, fearful, disgust, and surprised;

1.4 Purpose

The purpose of this research is as follows:

- To find out the process to detect human emotions from audio data with proposed method;
- To determine which method has higher accuracy between SVM, CNN, and the proposed method for detecting emotions through audio;
- To determine what architecture is going to give a good accuracy for the proposed model;

1.5 Benefit

The benefit of this research is to implement a human emotion detection system through voice for emotional perceptions of a robot machine intelligence. This study can also be used

as a reference in further research on either speech emotion recognition or human emotion recognition using audio data based on deep learning methods. Some industries that can benefit from this study, such as call center services, can implement a human emotion detection system with CNN through the customer's voice in their services to help improve the quality of service for the provider.

Chapter II
Literature Review

This chapter will discuss about previous research on this topic and present the foundational theories that guide this study.

2.1 Related Works

In the context of detecting human emotions through audio data, the selection and extraction of audio features are important to understand. The Sequential Minimal Optimization (SMO) algorithm was used as the primary method of sound analysis during the training of SVM models in recent years. In this case, the sound is divided into a number of frames which will then be examined iteratively. There are two emotional characteristics of the voice that can be observed to understand human emotion (Citron, Gray, Critchley, Weekes, & Ferstl, 2014):

- Arousal, the level of autonomic activation that an event creates, which ranges from calm to excited.
- Valence, the level of pleasantness that an event generates and is defined along a continuum from negative to positive.

The INTERSPEECH 2013 (Steidl, et al., 2013) introduced us to various aspects of speech and audio that are connected to emotions which employ the SMO algorithm using a rather 'brute force' method to classify and define audio feature sets. Another research such as (Eyben, et al., 2016) introduced a new method of audio feature extraction using a minimal set of parameters, which implements prosodic, excitation, vocal tract, spectral descriptors, and an extension to the minimalistic set, which contains a small set of cepstral parameters (i.e., MFCC & Spectral Flux).

Emotion recognition from pure speech is one of the most sophisticated and sophisticated and widespread techniques and progress in this field relies heavily on the composition of emotional speech datasets. The structure of the emotional speech corpus can be divided into two parts in general. The first part is lab recording, which is a collection of speech datasets that are often recorded in a recording studio using high-quality microphones and accompanied by linguistics experts. Some of the corpora that use this type of structure are EmoDB (Burkhardt, Paeschke, Rolfes, Sendlmeier, & Weiss, 2005), a database of german emotional speech comprising 800 sentences with 10 utterances by 10 different actors that could be used in normal conversation and could be interpreted according to all the emotions employed. IEMOCAP (Busso, et al., 2008), a database consisting of 12 hours worth of audiovisual with multimodal and multispeaker data, including 10 actors both scripted and improvised sessions recorded by the University of Southern California's SAIL Lab. AESDD (Vryzas, Kotsakis, Liatsou, Dimoulas, & Kalliris, 2018), includes Greek language expressions of acted emotional speech and the other controlling spontaneous emotional speech. The second corpus type is non-lab recording. This corpus contains utterances that reflect emotions involuntarily in natural scenarios, such as living spaces, theatrical performances, etc. Some examples that employ this type of corpus are DAPS (Mysore, 2015), this dataset is a collection of aligned recordings of the same speech made on typical consumer devices in real-world settings that consist of approximately 4 and a half hours of data. Freefield1010 (Stowell & Plumbey, 2013), a collection of 7690 excerpts from field recordings throughout the world, was later standardized for research. CHEAVD (Li, Tao, Chao, Bao, & Liu, 2017), containing 140 minutes of emotional segments from movies, TV shows, and talk shows with 238 speakers, ranging from children to the elderly, covers a wide range of speaker diversity.

Studies on different methods of speech representation have been done in recent years with

various types of deep-learning architecture. In 2019 (Schneider, Baevski, Collobert, & Auli, 2019), the wav2vec model introduced us to unsupervised learning for speech recognition by learning representations of unprocessed audio data. Then in 2020 (Baevski, Zhou, Mohamed, & Auli, 2020), the second version of this model was introduced which improves the model even further by employing a self-supervised training method based on contrastive learning for automatic speech recognition. However, in 2021, HuBERT (Hsu, et al., 2021) highlighted many issues with the self-supervised learning approach. These problems include (1) many pronunciation units in the speech, (2) no vocabulary of sound units during the pre-training phase, and (3) the length of sound units being changeable without any segmentation. With these problems, the idea of the HuBERT model is to apply the prediction loss only to masked regions and force the model to learn good high-level representations of unmasked inputs to infer the targets of masked ones correctly. Other studies such as the UniSpeech (Wang, et al., 2021) pointed out a problem in the speech recognition community that some of the successful techniques require thousands of hours of human-annotated speech recordings for training which is not available for a lot of languages spoken worldwide. The UniSpeech model can learn consistent contextual representations using both supervised and unsupervised data. This model consists of convolutional feature extraction, a transformer encoder, and a feature quantizer. UniSpeech is able to perform better than both supervised and unsupervised pre-training on multilingual speech recognition tasks. Furthermore, WavLM (Chen, et al., 2022) was introduced as an extension of HUBERT (Hsu, et al., 2021) to masked speech prediction and denoising modeling, so the pre-trained model performs well on both automatic and non-automatic speech recognition to solve full stack speech processing tasks. This model achieved the best performance on multiple speech datasets.

In a typical speech emotion recognition system, audio data, feature extraction, classification models, and emotional output recognition are all included. Some of the popular classification methods right now for an emotion recognition system include SVM (Sonawane, Inamdar, & Bhangale, 2017), Hidden Markov Model (HMM) (Starner & Pentland, 1995), and Recurrent Neural Network (RNN) (Chamishka, et al., 2022). Speech emotion recognition tasks require an emotion speech database for training the model. In this study, the RAVDESS (The Ryerson Audio-Visual Database of Emotional Speech and Song) (Livingstone & Russo, 2018) datasets are used for human emotion classification which has a recording of 24 actors each with 60 trials for 8 emotion classes including happy, neutral, sad, calm, angry, fearful, disgust and surprised with a total of 1440 North American English utterances in total.

2.2 Basic Theory

This chapter will explain the basic theory used as a reference in this study. Among other things, this chapter will explain the literature review, human emotion, voice understanding, speech recognition, feature extraction, neural network convolution, and transformers, as well as a brief explanation of the framework library, used to implement emotion detection in the human voice in this study, namely PyTorch.

2.2.1 Emotion

Emotion is an aspect of consciousness which are generally understood to represent the synthesis of subjective experience, expressive behavior, and neurochemical activity. Most researchers consider them to be part of the evolutionary legacy of the human species and serve adaptive purposes by supplementing common perception and facilitating social communication. (Solomon, 2009) Emotions come in a variety of forms, and they all have an impact on how humans live and relate to each other. There are times when we may feel as

though these emotions are controlling us. Our actions, behaviors, and perceptions are all influenced by the emotions we are experiencing at any given time. According to (Cherry, 2021), psychologist Paul Eckman identifies six fundamental emotions that were shared across all human societies in the 1970s. These emotions include *happiness, sadness, disgust, fear, surprise*, and *fury*. Later, he expanded this list for *pride, humiliation, embarrassment, and enthusiasm*. Figure 2.1 depicts various human emotions nowadays.

Figure 2.1: Human Emotions (Charlie, 2014).

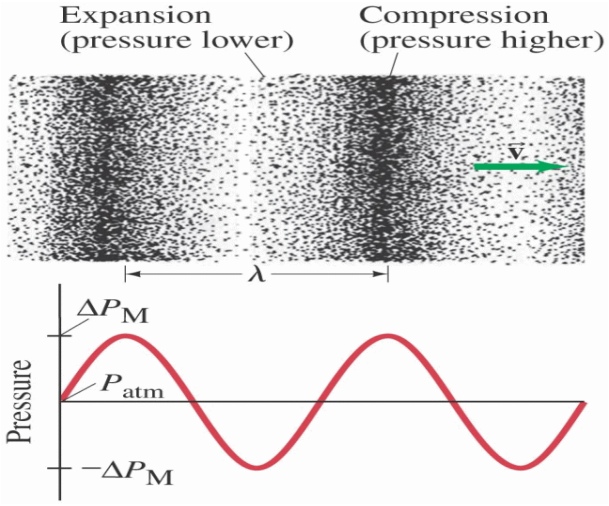

Figure 2.2: Longitudinal Nature of Sound Wave (StudyCorgi, 2022).

2.2.2 Sound

Sounds are produced by sound waves. Humans could hear it by passing a medium through the ears. All sound is produced by the vibration of molecules. For example, when a person makes a sound, there are vibrations move through the air molecules. Sound waves travel away from where they originate. When these vibrating air molecules reach the ear, the eardrum also vibrates. The bones in the ear vibrate as if the object that generated the sound waves vibrates. There are three types of continuous mediums which are solids, liquids, and gases. Sound travels faster through a solid medium since the particle here is closer together than in gases or liquid

medium. These vibrations let humans hear different things such as music. There are also irregular vibrations called noises. Human beings could make very complex sounds used for talking. A sound wave is a longitudinal wave that has two parts (Compression and Rarefaction). Compression is where air molecules are pushed together. Rarefaction is where the molecules are far apart. Sound is produced by a series of mechanical compressions and rarefactions of mechanical waves that sequentially propagate through a medium (StudyCorgi, 2022). Figure 2.2 shows a representation of the longitudinal nature of sound waves.

2.2.3 Speech Recognition

Speech Recognition is an interdisciplinary subject of computer science and computational linguistics that develops approaches and technology to enable the translation of spoken language into text by computer machines with the main benefit of searchability. It is often referred to as computer voice recognition or automatic speech recognition (ASR). Speech recognition draws on expertise and research from the domains of computer science, linguistics, and computer engineering.

Speech recognition systems use computer algorithms to process**,** interpret**,** and convert spoken words into text. A software program converts the sounds picked up by the microphone into characters that computers and humans can understand**.** This program must be able to adapt to the highly variable and context-specific nature of human speech. The software algorithms that process human speech are trained on a variety of speech patterns, speaking styles, language, accents, and idioms. The software also separates speech from the background noises that often accompany the signals (Yu & Deng, 2015).

2.2.4 Feature Extraction

In machine learning, feature extraction is the process of turning raw data into numerical features that can be processed while keeping the information in the original dataset. The amount of redundant data in the dataset is decreased within this process. In the end, the data reduction speeds up the learning and generalization phases of the machine learning process while also enabling the model to be built with less computation power. This study employs one of the most popular feature extraction methods in the context of Speech Emotion Recognition (SER) called the Mel-Frequency Cepstral Coefficient (MFCC) (Kishore & Satish, 2013). The procedure to find MFCCs is mainly with the following steps shown in Figure 2.3:

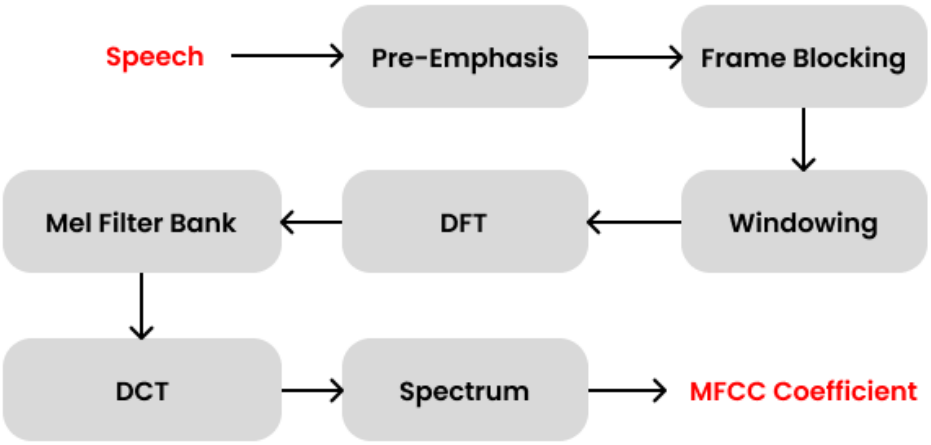

Figure 2.3: MFCC Block Diagram.

a. Pre-Emphasis

The structure of a voice production system's design causes dampening in high-frequency regions. Pre-Emphasis amplifies high-frequency sections and conducts filtering which is used to offset the spectrums of voiced regions. Widely used pre-emphasis filter is given in Equation 2.1,

$$y[n] = x[n] - a * x[n-1], a \approx 0.95 - 0.97. \quad (2.1)$$

Where:

- $y[n]$ is the output signal at time n.
- $x[n]$ is the input signal at time n.
- a is the pre-emphasis coefficient.
- $x[n-1]$ is the input signal at the previous time step (n-1).

b. Frame Blocking

Due to voice signal as a slow time-varying signal, speech analysis over a short enough time span is required for stable acoustic features. Frame blocking entails processing the voice signal at short time intervals to extract the characteristic features in a more stable condition.

c. Windowing

Windowing is the process of splitting an audio signal into segments of specific lengths. This reduces the effect of aliasing or signal discontinuity at the beginning and end of each frame that could occur due to the frame-blocking process.

d. Discrete Fourier Transform (DFT)

Discrete Fourier Transform is one of the most powerful tools in digital signal processing which enables us to find the spectrum of a finite-duration signal. In MFCC, DFTs are used to convert each windowed frame into a magnitude spectrum with Equation 2.2,

$$X(k) = \sum_{n=1}^{\infty} x(n) e^{-j2\pi kn/N} \quad (2.2)$$

Where:

- $X(k)$ is the k^{th} frequency domain sample, with k ranging from $0\ to\ N-1$.
- $x(n)$ it the n^{th} time domain sample, with n ranging from $0\ to\ N-1$.
- N it the number of samples in the sequence.
- j is the imaginary unit $(sqrt(-1))$.
- π is the mathematical constant $(3.1415\ldots)$.

e. Mel-Frequency Warping

In this process block, the triangle waves that make up the Mel filter bank's f frequency in Hz units are used to create the signal. As a result, using this method, the signal's value in $M(f)$ frequency units is determined. The MFCC coefficient value is determined by the number of filters in Mel's filter bank. The Mel scale is a nonlinear scale that compresses the higher

frequencies, which are more difficult for humans to perceive. The algebraic equation for the process of converting Mel spectrum and FFT frequency values in Hz to Mel frequency units is defined in Equation 2.3 as:

$$M(f) = 2595.\log_{10}(1 + \frac{f}{700}) \quad (2.3)$$

Where:

- $M(f)$ is the frequency of Mel.
- f is the frequency in Hz.
- log_{10} is the logarithm base 10.

f. Discrete Cosine Transform (DCT)

A DCT is applied to the transformed Mel frequency coefficients to produce a set of cepstral coefficients. The Mel spectrum was represented on a log scale which results in a signal in the cepstral domain with frequency peaks corresponding to the pitch on the signal. Since most of the signal information is represented by the first few MFCC coefficients, the system can be made robust by extracting only those coefficients ignoring higher-order DCT components.

g. Mel Cepstrum

The final result of the MFCC block process shown in Figure 3 is the coefficient of the Mel frequency cepstrum. A cepstrum representation of the speech spectrum adequately represents the local spectral characteristics of the signal for a given frame analysis.

2.2.5 Convolutional Neural Networks (CNN)

Convolutional neural networks are a subset of deep learning techniques that have gained prominence in several computer vision applications and are generating attention in many different fields, including speech recognition. CNN was intended to learn spatial hierarchies of characteristics automatically and adaptively, from low to high-level patterns. CNN is a mathematical construct that is usually composed of three types of layers including convolution, pooling, and fully connected layers. Compared to the traditional hand-crafted feature extraction techniques, CNN is far more data-hungry because of its millions of learnable parameters to estimate and is more computationally expensive, resulting in requiring graphical processing units (GPUs) for model training (Yamashita, Nishio, Do, & Togashi, 2018). Figure 2.4 shows a general view of how layers are connected inside a CNN architecture.

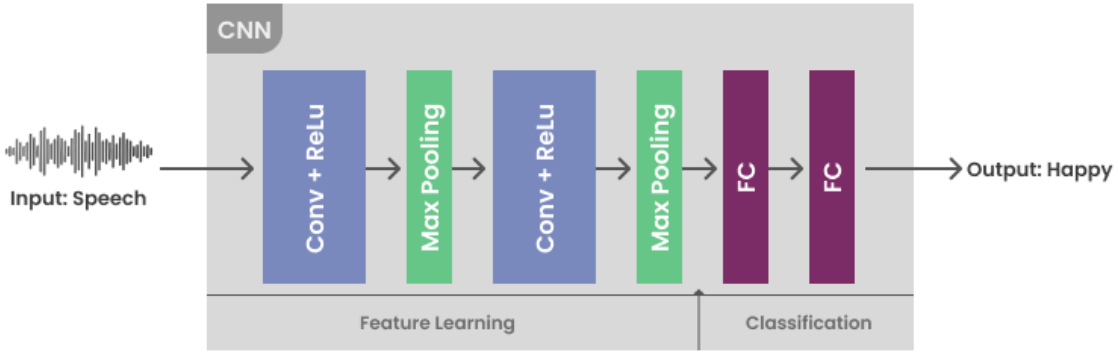

Figure 2.4: CNN Architecture.

a. Convolution

Convolution is a special type of linear operation used in feature extraction, where small numerical arrays (kernels) are applied to the input. This is an array of numbers called a tensor. The element-wise product between each element of the kernel and the input tensor is computed at each position of the tensor and summed to get the output value at the corresponding position of the output tensor, called a feature map, depicted in Figure 2.5. This process is repeated by applying multiple kernels to form any number of feature maps representing different properties of the input tensor. Therefore, different kernels can be viewed as different feature extractors. Two important hyperparameters that define the convolution operation are the size and number of kernels.

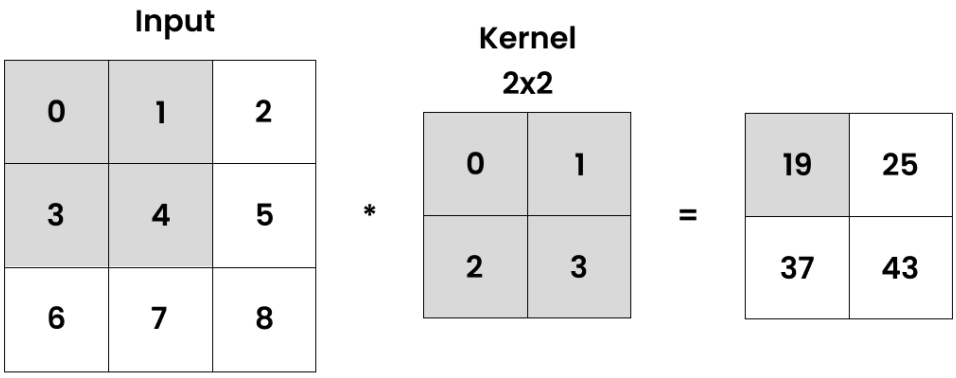

Figure 2.5: 2×2 Convolution Filter.

b. Activation Function

The activation function is the node that is added at the end of each output of the neural network. In the CNN architecture, the activation function is the final calculation of the feature map output, or the generation of feature patterns after the convolution or merging calculation process. Although smooth nonlinear functions like the *sigmoid* or *hyperbolic tangent* (tanh) function have been employed in the past because they are mathematical representations of the behavior of biological neurons, the *rectified linear unit* (ReLU) is currently the most widely utilized nonlinear activation function, which simply computes the function in Equation 2.4 as follows:

$$f(x) = \max(0, x) \tag{2.4}$$

Where:

- $f(x)$ is the output of the function.
- x is the input to the function.

c. Max Pooling

A pooling layer offers a standard down-sampling method that lowers the feature map's in-plane dimensions to introduce translation invariance to slight shifts and distortions and limit the number of ensuing learnable parameters. One of the most popular types of pooling operations is max pooling. The idea behind max pooling is that it preserves the most important information from the input while discarding less important information. This can be particularly useful for classification tasks, where the max pooling layer can help the model focus on the most important features in an audio, such as spectral peaks, spectral roll-off points, and spectral flux. Figure 2.6 shows an example of a max pooling with 2×2 filter on a 4×4 feature map.

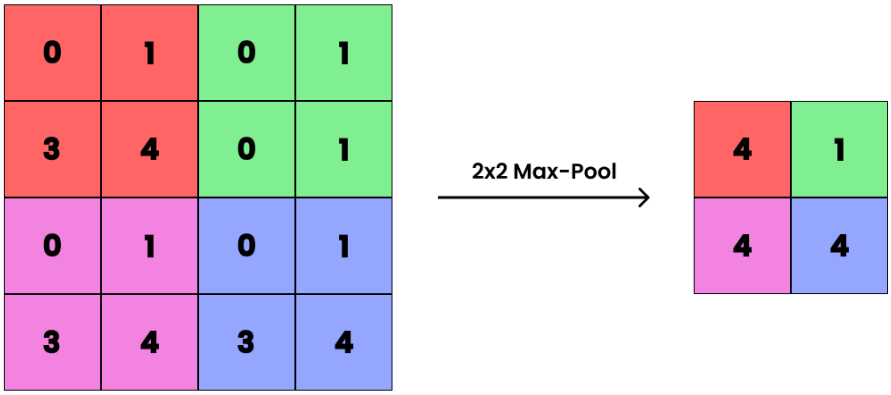

Figure 2.6: 2x2 Max Pooling Layer

d. Fully Connected Layer

Feature maps generated from the feature extraction layers are still in the form of a multidimensional array. Therefore, these feature maps are typically flattened, or converted into a one-dimensional array of vectors, and connected to one or more fully connected layers, also known as dense layers, in which each input is connected to their outputs by learnable weight resulting in probabilities for each class in the classification tasks. After passing through the fully connected layers, the final layer uses the SoftMax activation function that normalizes real values output from the last fully connected layer to get probabilities of the input being in a particular class (classification) where each value ranges between 0 and 1. The final fully connected layer usually has as many output nodes as there are classes.

2.2.6 Transformer

The transformer is a deep learning model architecture that is built entirely on the self-attention mechanism to weigh the importance of each part of the input data differently. It is mainly used in the fields of natural language processing (NLP). This architecture is designed to process sequential input data to solve NLP-related tasks such as text translation or summarization. However, unlike Recurrent Networks (GRU, LSTM), transformers could process the entire input at once. Attention mechanisms provide context for each position in the input sequence which allows for more parallelization than recurrent neural networks and therefore reduces training time. The model of the transformer architecture follows the overall architecture of Figure 2.7 using stacked self-attention and pointwise fully connected layers for both the encoder and decoder shown in the left and right halves of the figure respectively (Vaswani, et al., 2017).

a. Self-Attention

In artificial neural networks, attention is a technique designed to mimic cognitive attention. This effect improves some parts of the input data and reduces others. The motivation for this is that networks need to pay more attention to small but important pieces of data. Learning which parts of the data are more important than others is context-dependent, which is trained by gradient descent. Attention functions can be described as associating a query and a set of key-value pairs with an output. Where query, key, value, and output are all vectors. The output is computed as a weighted sum of the values. The weight assigned to each value is calculated by the query compatibility function using the appropriate key.

Self-attention, also called intra-attention, is an attention mechanism that associates

different positions of a single sequence to compute representations of the same sequence. In a self-attention mechanism, each element in the input is represented as a vector, and the model learns a set of attention weights that determine how much importance each element should be given when producing the output. The attention weights are learned through training and allow the model to selectively focus on certain parts of the input while ignoring others. Self-attention has proven especially useful for machine reading, summarizing summaries, or generating image descriptions.

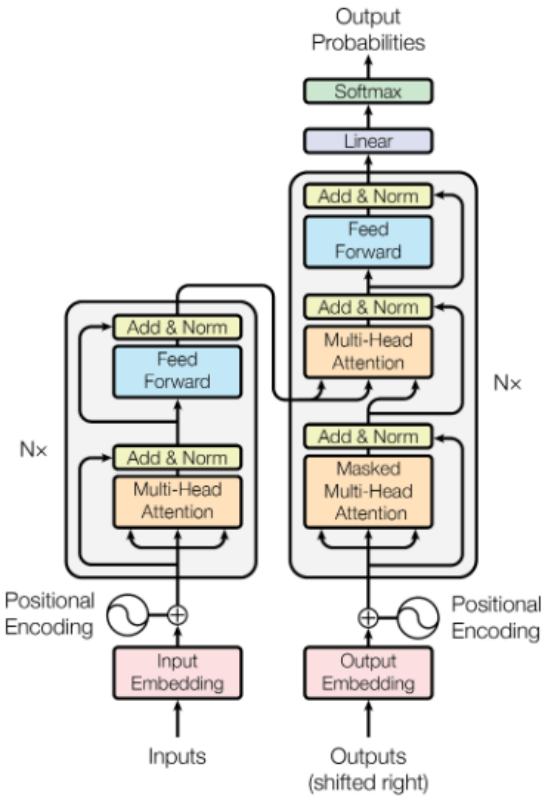

Figure 2.7: Transformer Architecture (Vaswani, et al., 2017).

b. Multi-Head Self Attention

In Transformer, the Attention module iterates its computation several times in parallel. Each of them is called an attention head. The Attention module splits its query, key, and value parameters N times, passing each split individually through a separate head. All these similar attention calculations are combined to produce a final attention score. This is called multi-headed attention and gives the Transformer greater power to encode multiple relationships and nuances for each word. Multi-head attention allows the model to jointly pay attention to information from different representational subspaces at different positions. In most general form, the multi-head attention mechanism can be represented as shown in Equation 2.5. Figure 2.8shows that a multi-head attention consists of several attention layers running in parallel.

$$MultiHead(Q, K, V) = concat(head_1, head_2, \ldots, head_h) * W^O \quad (2.5)$$

Where:

- $Q, K, and\ V$ are matrices of queries, keys, and values respectively.

- $head_1, head_2, \ldots, head_h$ are the attention maps computed by the h different attention heads.
- W^O is a learned projection matrix.
- $concat$ is a function that concatenates the attention maps along the second dimension.

Each attention head computes an attention map using Equation 2.6 below:

$$head_h = attention(QW_h^Q, KW_h^K, VW_i^V) \tag{2.6}$$

Where:

- $W_i^Q,\ W_i^K,\ and\ W_i^V$ are learned projection matrices for the h^{th} attention head.

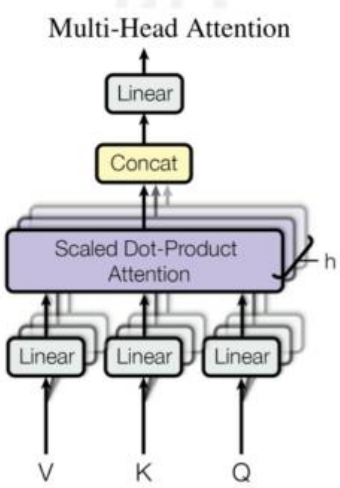

Figure 2.8: Multi-Head Attention (Vaswani, et al., 2017).

c. Scaled Dot-Product Attention

Transformers implement scaled dot product attention depicted in Figure 2.9, that follows the steps of the general attention mechanism. Scaled dot product attention first computes the dot product of each query and every key. Then divide each result by $\sqrt{d_k}$ and apply the softmax function. In doing so, it obtains the weights that are used to scale the values. The formula for scaled dot product attention was defined below in Equation 2.7 as:

$$Attention(Q, K, V) = softmax\ (\frac{QK^T}{\sqrt{d_k}})\mathrm{V} \tag{2.7}$$

Where:

- $Q, K, and\ V$ are matrices of queries, keys, and values respectively.
- QK^T is the dot product of the queries and keys.
- d_K is the dimensionality of the keys.
- $softmax$ is the SoftMax function, which normalizes the attention weights.

In practice, the computations performed by scaled dot product attention can be efficiently

applied to the entire set of queries at once. For this purpose, the matrices $Q, K, and\ V$ are supplied as inputs to the attention function. The scaling factor $1/\sqrt{d_k}$ is included to help stabilize the attention weights and improve the numerical stability of the model.

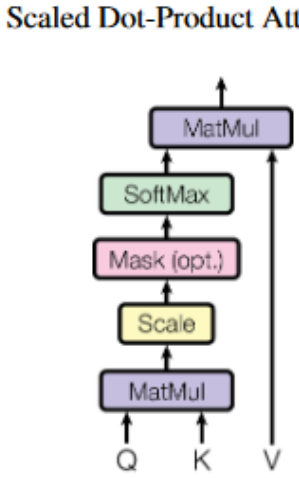

Figure 2.9: Scaled Dot-Product Attention (Vaswani, et al., 2017).

d. Encoder

Figure 2.10 shows an encoder block's two main components: the self-attention mechanism and a feed-forward neural network. The self-attention mechanism accepts an input encoding from previous encoders and weighs their relevance against each other to produce an output encoding. Then, a feed-forward neural network processes each output code independently. These output encodings are passed as inputs to the following encoders as well as the decoders block. Each sub-layer employs a residual connection and normalization layer.

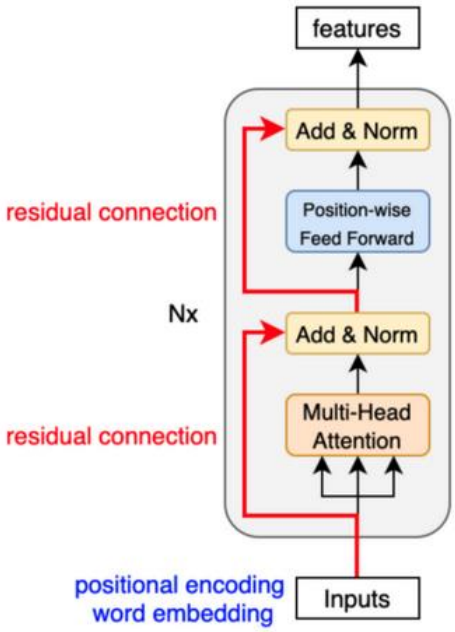

Figure 2.10: Encoder Block (KiKaBeN, 2021).

e. Decoder

The decoder block takes the encoder's two main components of a self-attention mechanism and a feed-forward neural network and inserts a third sub-layer that performs multi-head attention over the output of the encoder stack, shown in Figure 2.11. This new sub-layer obtains relevant information from the encoding produced by the encoder block. Like the encoder block, each sub-layer employs a residual connection and a normalization layer.

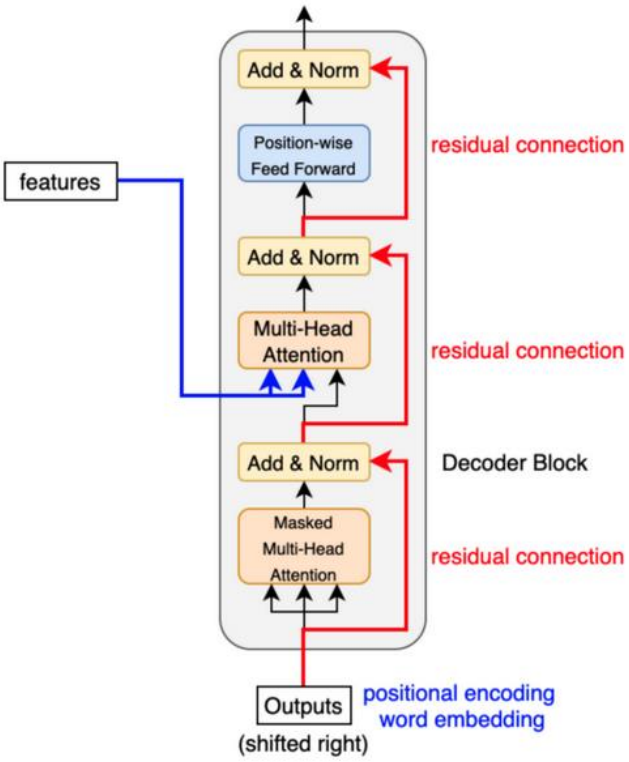

Figure 2.11: Decoder Block (KiKaBeN, 2021).

2.2.7 PyTorch

PyTorch is an open-source machine learning framework based on the Python programming language and the torch library. It is developed primarily by the Meta AI research team and can be used in both Python and C++ programming languages. However, this framework works best with Python. Over 200 and more different mathematical operations are supported by the PyTorch framework and its popularity is still growing because it makes building models for artificial neural networks simpler. Researchers primarily utilize PyTorch for research and applications using artificial intelligence (AI).

Because of the pythonic nature of this framework, PyTorch is able to utilize core python concepts such as classes, structures, and conditional loops making it easy and intuitive to understand. PyTorch is also popular for its dynamic computation graphs, which allow greater flexibility in building complex architectures. This allows neural network developers and scientists to run and test pieces of code in real-time, rather than waiting for the entire program to be written (Paszke, et al., 2019).

Chapter III
Methodology

This chapter will provide an overview of the proposed method for our study, including the tools and techniques that will be used, as well as plans for implementation and testing.

3.1 Designed Method

This section provides a summary of the proposed architectural model's functionality and includes a diagram (Figure 3.1) that gives an overview of the model's architecture.

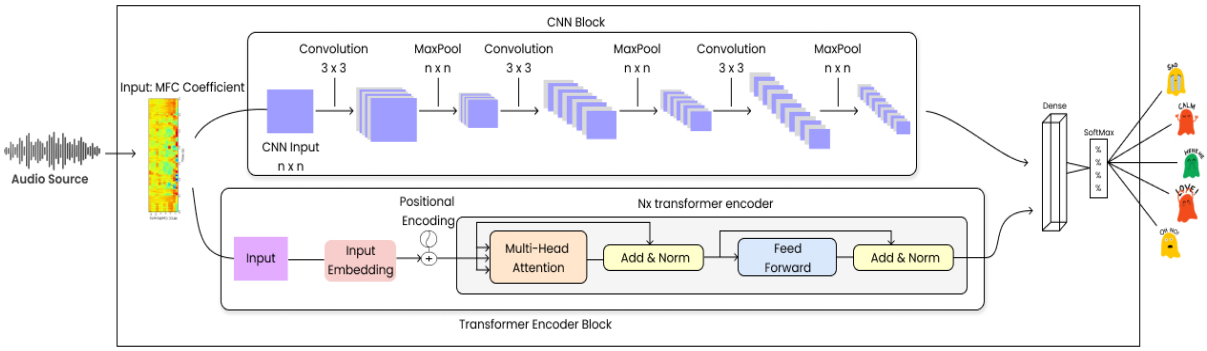

Figure 3.1: Model Architecture.

The CNN architecture in this study is based on recent advancements in image and sequence processing. It includes a series of convolutional and pooling layers, similar to the classic LeNet architecture (LeCun, Bottou, Bengio, & Haffner, 1998), which extract features from the input data and reduce the size of the feature maps through downsampling. The fully-connected layers then process the extracted features to produce the final output of the network, which is transformed into a probability distribution over the possible classes using the SoftMax function. While LeNet is a relatively simple architecture compared to modern CNNs, it has been successful in many classification tasks and has been applied in various domains such as handwritten digit recognition.

The Transformer architecture is precisely as (Vaswani, et al., 2017). However, in this study, only the encoder blocks are employed which is a component of the Transformer architecture that was introduced in the paper. It is used to process the input sequence and extract relevant features that will be passed to the fully-connected layers, which process these features to produce the final output of the network.

The encoder block consists of a self-attention layer followed by a feedforward layer. The self-attention layer uses a dot-product attention mechanism to calculate the attention weights between each pair of input elements. These weights are then used to compute a weighted sum of the input elements, which is used as the output of the self-attention layer.

The feedforward layer consists of two linear transformations with a ReLU activation function in between. It takes the output of the self-attention layer as input and produces the final output of the encoder block.

The success in the use of the parallel deep learning technique of GoogleNet (Szegedy, et al., 2015), also known as Inception-v1, was the inspiration for the parallel architecture of this study, which allows the network to process multiple features concurrently. This could be achieved by using a series of inception modules, which will be concatenated and fed into the

fully-connected (dense) layer. This parallel architecture enables GoogleNet to achieve good performance while being relatively efficient in terms of the number of parameters and computation time. It has been widely used in many image classification and object detection tasks.

3.2 Supporting Tools

In order to carry out this study, certain tools and equipment will be needed, including both hardware and software. The specific devices that will be used in this research are listed below:

3.2.1 Hardware

The hardware necessary for this study includes:

1. Lenovo Legion 5 2021 Laptop with the following specifications:
    a. AMD Ryzen 7 5800H (8 cores / 3.20GHz)
    b. NVIDIA RTX 3070 Laptop GPU
    c. 16GB of Random Access Memory (3200MHz)
    d. 1TB Solid State Drive (SSD)

3.2.2 Software

To ensure that the proposed model in this study performs correctly, certain software tools will be utilized to support this research. The software that will be used in this study includes:

1. Operating System: Windows 11
2. Programming Language: Python 3.10.9
3. Editor: Jupyter Notebook
4. Framework: PyTorch 1.13.1

3.3 Implementation and Trial Plans

This section will explain the dataset used in the study, as well as the stages of implementation for the proposed method, including pseudocode and explanations. The evaluation metrics used to assess the performance of the proposed method will also be described.

3.3.1 Dataset

The dataset used for this study is the RAVDESS (Livingstone & Russo, 2018) dataset to classify emotions from one of eight classes. The RAVDESS (Ryerson Audio-Visual Database of Emotional Speech and Song) is an audio dataset that is widely used in research on emotion recognition and speech processing. The dataset consists of recordings of actors speaking and singing in different emotional states.

The RAVDESS dataset includes a total of 7356 files, comprising 24 actors (12 male and 12 female). The actors were recorded speaking and singing in eight different emotional states: neutral, calm, happy, sad, angry, fearful, surprised, and disgusted. Each actor recorded a total of 144 items (72 speech and 72 song items), with each item being approximately 5 seconds long. The actors were recorded in a studio setting, and the audio and video recordings were captured simultaneously.

The RAVDESS dataset is valuable for researchers studying emotion recognition and speech processing because it includes both audio and video recordings, which allows for the study of both vocal and facial expressions of emotion. It is also useful because it includes a diverse range of emotions, and because the actors are native speakers of North American English, which is a common language used in research.

3.3.2 Implementation Stage

This section will outline the steps involved in conducting the research and explain the progression of the study from beginning to end. The various stages of the research are depicted in Figure 3.1, which provides a visual representation of the proposed method's model architecture flow.

a. Feature Extraction

The initial stage in this research is by extracting the audio features from the input data using the Mel-Frequency Cepstral Coefficients (MFCC). MFCC is a popular feature extraction technique used in speech and audio processing because it is able to capture the spectral characteristics of an audio signal in a compact and efficient manner. MFCCs are derived from the power spectrum of an audio signal and are based on the Mel-scale, which is a non-linear scale that is based on the perceived frequency of a sound by the human ear. This makes MFCCs well-suited for tasks such as speech recognition and speaker identification, where the human ear is the primary means of perception. Algorithm 3.1 shows a pseudocode for extracting MFCC features from an audio signal using the librosa library.

Algorithm 3.1: MFCC Feature Extraction.

```
import librosa

def extract_mfcc(audio_path):
  #Load the audio signal from the audio file
  signal, sr = librosa.load(audio_path)

  #Pre-processing: apply a Hanning window to the signal and compute the
STFT
  windowed_signal = signal * librosa.filters.window('hann', len(signal))
  stft = librosa.stft(windowed_signal)

  #Mel-scale transformation: convert the STFT to the Mel-scale
  mel_basis = librosa.filters.mel(sr, n_fft=2048, n_mels=40)
  mel_spectrum = np.dot(mel_basis, np.abs(stft) ** 2)

  #Discrete Cosine Transform (DCT): convert the Mel-scaled spectrum to
the frequency domain using DCT
  mfcc = librosa.feature.mfcc(S=librosa.power_to_db(mel_spectrum),
n_mfcc=40)

  #Cepstral Mean Normalization (CMN): apply CMN to the MFCC coefficients
  mfcc_cmn = librosa.feature.cmn(mfcc, center=True)

  return mfcc_cmn
```

The implementation loads the audio signal from a file using the librosa library, applies a Hanning window to the signal, and computes the short-time Fourier transform (STFT). Then, it converts the STFT to the Mel-scale using a Mel-scale filterbank and applies the Discrete Cosine Transform (DCT) to the resulting Mel-scaled spectrum to obtain the MFCC coefficients. Finally, the Cepstral Mean Normalization (CMN) is applied to the MFCC coefficients to normalize the overall spectral envelope.

b. Model Architecture Design

After the audio features have been extracted, the next step is to create the model

architecture and use the extracted features as input to the model which will allow the model to classify human emotions based on the extracted features. There are two blocks of the deep learning model for the purposed method, the CNN block and the Transformer block which will be working in parallel with each other. The idea is for the CNN to give spatial feature representation of the input data, and the Transformer block in sequence encoding to try and model as accurately as possible the temporal relationships between pitch transitions in human emotions. The expansion of CNN filter channels and reduction of feature maps will provide the most expressive feature representation with the lowest computational cost, while the Transformer encoder will learn to predict frequency distributions of different emotions according to the global structure of the MFCC plot of each emotion. The implementation for CNN and Transformer block will be shown in Algorithm 3.2 and Algorithm 3.3, respectively.

Algorithm 3.2: CNN Block.

```
#Input layer
input_data = read_data(data_path)

#Convolution layer
feature_maps = apply_filters(input_data, filters)

#Activation layer
activated_feature_maps =
apply_activation(feature_maps,activation_function)

#Pooling layer
pooled_feature_maps =
apply_pooling(activated_feature_maps,pooling_function)

#Repeat
for i in range(num_iterations):
  feature_maps = apply_filters(pooled_feature_maps[i-1], filters)
  activated_feature_maps = apply_activation(feature_maps,
activation_function)
  pooled_feature_maps = apply_pooling(activated_feature_maps,
pooling_function)
```

Algorithm 3.2 presents an implementation of the CNN Block for the proposed deep learning model. The first layer is the input layer takes in audio features with a certain size and number of channels. Second layer is the convolution layer then applies a set of filters to this input data, generating a set of feature maps. In this study, the filters are 3x3 matrices and they are applied to the input data through a process called convolution. Third layer is the activation layer which applies an activation function (i.e., ReLU) to the feature maps generated by the previous layer. This layer introduces non-linearity to the model, allowing it to learn more complex relationships in the data. Finally, the pooling layer down-samples the feature maps by applying a pooling operation (i.e., MaxPooling). This helps reduce the size of the feature maps and, as a result, lowers the computational complexity of the model. This sequence of applying convolutional layers, activation layers, and pooling layers is repeated three times in the proposed model architecture.

The Transformer encoder implementation was shown in Algorithm 3.3 which is designed to process sequential data of the audio source. It consists of a series of self-attention layers and feedforward layers, which are used to predict frequency distributions of different emotions

according to the global structure of the MFCCs of each emotion. In the implementation, the output sequence is initialized first as an empty list. Then the input sequence is embedded by applying an embedding matrix to each element of the input, resulting in a sequence of embedded vectors. The embedded sequence will then be passed through a series of self-attention layers to compute a sequence of context-aware representations. Each self-attention layer applies the attention mechanism to the input sequence to compute a weighted sum of the input vectors, where the weights are computed based on the relationships between the input elements. These context-aware representations are then passed through a series of feedforward layers to compute a sequence of transformed representations. Each feedforward layer applies a linear transformation to the input, followed by a nonlinear activation function. Finally, the transformed representations are used to compute the output sequence by applying a linear transformation and an activation function to each element of the transformed representations.

Algorithm 3.3: Transformer Encoder Block.

```
def transformer_encoder(x, params):
  #Initialize output sequence y
  y = []

  #Embed input sequence x
  x_emb = embed(x, params["embedding_matrix"])

  #Pass embedded sequence x_emb through self-attention layers
  x_att = x_emb

  for i in range(num_self_attention_layers):
    #Compute context-aware representation x_att using self-attention
    x_att = self_attention_layer(x_att,
params["self_attention_layer_{}".format(i)])

  #Pass context-aware representation x_att through feedforward layers
  x_ff = x_att
  for i in range(num_feedforward_layers):
    #Compute transformed representation x_ff using feedforward layer
    x_ff = feedforward_layer(x_ff,
params["feedforward_layer_{}".format(i)])

  #Use transformed representation x_ff to compute output sequence y
  for i in range(len(x)):
    #Compute output y_i using linear transformation and softmax
    y_i = softmax(x_ff[i] @ params["output_matrix"])
    y.append(y_i)

  return y
```

The final stage of the purposed model is to concatenate both outputs from the CNN model and the Transformer encoder model and pass the resulting tensor through a dense layer with a softmax activation function for prediction. The softmax function is a common choice for the activation function in the final layer of a classification model. It takes a vector of arbitrary real-valued scores and converts it into a probability distribution, where the probability of each class is given by the corresponding element in the output vector. Algorithm 3.4 outlines the process of combining the outputs of the CNN and Transformer models, passing them through a dense layer, and applying the softmax function to the output of the dense layer to make predictions.

Algorithm 3.4: Dense Layer Concatination.

```
import torch

#Concatenate the outputs along the feature dimension
combined_output = torch.cat((cnn_output, transformer_output), dim=1)

#Pass the combined output through a dense layer with a softmax activation
predictions = torch.nn.Sequential(
    torch.nn.Linear(cnn_features + transformer_features, num_classes),
    torch.nn.Softmax(dim=1)
)(combined_output)

return predictions
```

The dense layer implementation of Algorithm 3.4 starts by combining the output tensors of CNN and Transformer which has the shape (batch_size, feature_size) for each of the models, respectively. The combined output is then passed through a linear layer with the number of class units, which is the eight different emotional states in the RAVDESS dataset. The dense layer has weights and biases that will be learned during training to transform the combined output into the final prediction. Finally, the SoftMax activation function is applied to the output of the linear layer that will convert the prediction scores into a probability distribution over the classes. The class with the highest probability is taken as the model's prediction.

3.3.3 Model Evaluation

Model evaluation is an important step in the development of a deep learning model, as it could assess the performance of the model on unseen data and determine its suitability for a given task. This section will outline some general considerations for evaluating deep learning models for a speech emotion recognition task.

There are several ways to evaluate the performance of the purposed deep learning model. This study aims to compare the performance of three different machine learning models on a speech emotion recognition task, the standard Convolution Neural Network (LeNet) model, the Support Vector Machine (SVM) model, and the purposed method for this study. A combination of different evaluation metrics will be used to evaluate the performance of these models.

First, the model's training and validation accuracy will be tracked to ensure that the model is not overfitting the training data. Tracking the training process of a model could help identify and address the overfitting and underfitting in the data. Overfitting occurs when the model performs well on the training data but poorly on the validation or test data, indicating that it has learned patterns that are specific to the training data and are not generalizable. While underfitting occurs when the model performs poorly on both the training and validation data, indicating that it is not able to learn the underlying patterns in the data. In addition to addressing overfitting and underfitting, tracking the model’s accuracy on the validation set can also help choose the best hyperparameter values leading to the best model performance by observing and changing the effect on the validation accuracy. In summary, this method will show how well the models can learn the classification task and identify any issues with the optimization

process. After training the model, the test set will be used to evaluate their performance using several metrics.

One of the best metrics to evaluate is the test set accuracy for each model to get an idea of how the model performs on unseen data. This metric will give a summary of the model's performance on the test set. The test set is a set of data that the model has not seen during training, and therefore provides a more realistic evaluation of the model's performance. Evaluating the model on the test set accuracy can give a more accurate assessment of the model's generalization ability, which is its ability to perform well on unseen data. This is essential for understanding the model's suitability for deployment and its potential real-world performance.

In addition to accuracy, a confusion matrix can also be used for each model to understand the types of errors that these models are making and to identify any imbalances in the data. The confusion matrix is a table that shows the number of true positive, true negative, false positive, and false negative predictions made by the model. True positive predictions are those where the model correctly predicts the positive class, while true negative predictions are those where the model correctly predicts the negative class. False positive predictions are those where the model incorrectly predicts the positive class, while false negative predictions are those where the model incorrectly predicts the negative class. This evaluation metric will be able to see how well the models are performing in each emotion class and a more detailed understanding of the model's strengths and weaknesses.

Other metrics such as precision, recall, and the F1 score can be used on the test set for each model. These evaluation metrics are commonly used to assess the performance of deep learning models, particularly for classification tasks. Precision measures the proportion of true positive predictions made by the model among all positive predictions, while recall measures the proportion of true positive predictions made by the model among all actual positive examples. The F1 score is a combination of precision and recall and is calculated as the harmonic mean of the two. The F1 score is useful because it takes into account both the precision and recall of the model and provides a single metric that reflects the model's overall performance. These metrics of evaluation give a detailed understanding of the model's performance and compare them to each of the eight different emotion classes in the RAVDESS dataset.

Lastly, the weighted average of the F1 scores could be considered as one of the metrics of evaluation for this study. The weighted average of the F1 score is a variation of the F1 score that is used to evaluate the performance of a classification model when dealing with imbalanced classes. In an imbalanced dataset, the classes are not equally represented, which can make it difficult to accurately evaluate the model's performance. The weighted average helps address the issue by adding weights in different classes in the calculation of the F1 score. These weights reflect the relative importance of the different classes and can be used to give more emphasis to the performance of the model on a particular class. This metric is useful for evaluating the model's performance with imbalance classes and can be used to compare the overall performance of the three models for the speech emotion recognition task.

Schedule of Activities

NO	Nama Kegiatan	Minggu ke-													
		1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	Studi Pustaka														
2	Perancangan peralatan														
3	Survey Lapangan														
4	Eksperimen														
5	Analisa														
6	Pengolahan data														
7	Pelaporan kemajuan														
8	Pembuatan absrak seminar														
9	Mengikuti seminar														
10	Penyusunan laporan Tugas Akhir														

References

Baevski, A., Zhou, H., Mohamed, A., & Auli, M. (2020). wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations. *arXiv*(3).

Brown, Mann, T. a., Ryder, B. a., Subbiah, N. a., Kaplan, M. a., Dhariwal, J. D., . . . Gretch. (2020). Language Models are Few-Shot Learners. In *Advances in Neural Information Processing Systems* (pp. 1877-1901). Curran Associates, Inc.

Burkhardt, F., Paeschke, A., Rolfes, M., Sendlmeier, W. F., & Weiss, B. (2005). A database of German emotional speech. *INTERSPEECH 2005 - Eurospeech, 9th European Conference on Speech Communication and Technology*, (pp. 1517-1520). Lisbon.

Busso, C., Bulut, M., Lee, C., Kazemzadeh, A., Mower, E., Kim, S., . . . Narayanan, S. (2008). IEMOCAP: Interactive emotional dyadic motion capture database. *Language Resources and Evaluation, 42*, 335-359.

Chamishka, S., Madhavi, I., Nawaratne, R., Alahakoon, D., Silva, D. D., Chilamkurti, N., & Nanayakkara, V. (2022). A voice-based real-time emotion detection technique using recurrent neural network empowered feature modelling. *SpringerLink, 81*, 35173–35194.

Charlie. (2014, July 14). *It's No Disgrace To Use Your Face!* (The SAVI Singing Actor) Retrieved December 2022, 17 from https://www.savisingingactor.com/its-no-disgrace-to-use-your-face/

Chen, S., Wang, C., Chen, Z., Wu, Y., Liu, S., Chen, Z., . . . Wei, F. (2022). WavLM: Large-Scale Self-Supervised Pre-Training for Full Stack Speech Processing. *IEEE Journal of Selected Topics in Signal Processing, 16*, 1505-1518.

Cherry, K. (2021, April 5). *The 6 Types of Basic Emotions and Their Effect on Human Behavior* . (verywellmind) Retrieved November 20, 2022 from https://www.verywellmind.com/an-overview-of-the-types-of-emotions-4163976

Citron, F., Gray, M. A., Critchley, H., Weekes, B., & Ferstl, E. C. (2014). Emotional valence and arousal affect reading in an interactive way: Neuroimaging evidence for an approach-withdrawal framework. *Neuropsychologia, 56,100*, 79–89.

Costello, K. (2019, January 21). *Gartner Survey Shows 37 Percent of Organizations have Implemented AI in Some Form.* (Gartner) Retrieved October 11, 2022 from https://www.gartner.com/en/newsroom/press-releases/2019-01-21-gartner-survey-shows-37-percent-of-organizations-have

Cowie, R. (2001). Emotion Recognition in Human-Computer Interaction. *IEEE, 18*(1), 32-80.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *ArXiv*.

Eyben, Scherer, F. a., Schuller, K. R., Sundberg, B. W., André, J. a., Busso, E. a., . . . P., K. (2016). The Geneva Minimalistic Acoustic Parameter Set (GeMAPS) for Voice Research and Affective Computing. *IEEE Transactions on Affective Computing, 7*(2), 190-202.

Heredia, J., Cardinale, Y., Dongo, I., & Díaz-Amado, J. (2021). A Multi-modal Visual Emotion Recognition Method to Instantiate an Ontology. *16th International Conference on Software Technologies*, 453-464.

Hsu, W.-N., Bolte, B., Tsai, Y.-H. H., Lakhotia, K., Salakhutdinov, R., & Mohamed, A.

(2021). HuBERT: Self-Supervised Speech Representation Learning by Masked Prediction of Hidden Units. *IEEE/ACM Transactions on Audio, Speech, and Language Processing, 29*, 3451-3460.

Ivar, Byron, R., & Clifford, N. (1996). *The media equation: How people treat computers, television, and new media like real people and places.* Cambridgeshire, England: Center for the Study of Language and Inf.

Jarymowicz, & Maria. (2012). Understanding Human Emotions. *Journal of Russian & East European Psychology*, *50*(3), 9-25.

Keltner, Dacher, & Cordaro, D. T. (2017). Understanding Multimodal Emotional Expressions: Recent Advances in Basic Emotion Theory. In *The Science of Facial Expression* (pp. 57-76). New York: Social Cognition and Social Neuroscience.

KiKaBeN. (2021, December 13). *Transformer's Encoder-Decoder: Let's Understand The Model Architecture*. Retrieved December 18, 2022 from https://kikaben.com/transformers-encoder-decoder/

Kishore, K., & Satish, K. (2013). Emotion recognition in speech using MFCC and wavelet features. *2013 3rd IEEE International Advance Computing Conference (IACC)*, 842-847.

Latif, S., Rana, R., Younis, S., Qadir, J., & Epps, J. (2018). Transfer Learning for Improving Speech Emotion Classification Accuracy. *arXiv*(4).

Lech, M., Stolar, M., Best, C., & Bolia, R. (2020). Real-Time Speech Emotion Recognition Using a Pre-trained Image Classification Network: Effects of Bandwidth Reduction and Companding. *Frontiers in Computer Science, 2*, 14.

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-Based Learning Applied to Document Recognition. *Proceedings of the IEEE*(11), 2278-2324.

Li, Y., Tao, J., Chao, L., Bao, W., & Liu, Y. (2017). CHEAVD: a Chinese natural emotional audio–visual database. *Journal of Ambient Intelligence and Humanized Computing, 8*, 913-924.

Livingstone, S. R., & Russo, F. A. (2018). The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS). *Zenodo, 13*(5), e0196391.

Mahapatra, S. (2018, March 22). *Why Deep Learning over Traditional Machine Learning?* (Towards Data Science) Retrieved October 14, 2022 from https://towardsdatascience.com/why-deep-learning-is-needed-over-traditional-machine-learning-1b6a99177063

Mysore, G. J. (2015). Can We Automatically Transform Speech Recorded on Common Consumer Devices in Real-World Environments into Professional Production Quality Speech? - A Dataset, Insights, and Challenges. *IEEE Signal Processing Letters.*

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., . . . Chilamkurthy, S. (2019). PyTorch: An Imperative Style, High-Performance Deep Learning Library. *Advances in neural information processing systems, 32*.

Schneider, S., Baevski, A., Collobert, R., & Auli, M. (2019). wav2vec: Unsupervised Pre-training for Speech Recognition. *arXiv*(4).

Solomon, R. C. (2009, July 29). *emotion*. (Encyclopedia Britannica) Retrieved November 20, 2022 from https://www.britannica.com/science/emotion

Sonawane, A., Inamdar, M. U., & Bhangale, K. B. (2017). Sound based human emotion

recognition using MFCC & multiple SVM. *IEEE*, 1-4.

Starner, T., & Pentland, A. (1995). Real-time American Sign Language recognition from video using hidden Markov models. *Proceedings of International Symposium on Computer Vision - ISCV*, 265-270.

Steidl, B. S., Batliner, A., Vinciarelli, A., Scherer, K., Ringeval, a., Chetouani, M., . . . Kim, S. (2013). The INTERSPEECH 2013 computational paralinguistics challenge: Social signals, conflict, emotion, autism. *Proceedings INTERSPEECH 2013, 14th Annual Conference of the International Speech Communication Association.* Lyon, France.

Stowell, D., & Plumbey, M. D. (2013). An open dataset for research on audio field recording archives: freefield1010. *arXiv*.

StudyCorgi. (2022, June 25). (StudyCorgi) Retrieved November 21, 2022 from https://studycorgi.com/the-characteristics-of-sound/

StudyCorgi. (2022, 25 June). Retrieved December 17, 2022 from https://studycorgi.com/the-characteristics-of-sound/

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., . . . Rabinovich, A. (2015). Going deeper with convolutions. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1-9.

Vaswani, Shazeer, A. a., Parmar, N. a., Uszkoreit, N. a., Jones, J. a., Gomez, L. a., . . . Illia. (2017). Attention is All you Need. In *Advances in Neural Information Processing Systems.* Curran Associates, Inc.

Vryzas, N., Kotsakis, R., Liatsou, A., Dimoulas, C., & Kalliris, G. (2018). Speech Emotion Recognition for Performance Interaction. *Journal of the Audio Engineering Society. Audio Engineering Society, 66*(6), 457-467.

Wang, Wu, C. a., Qian, Y. a., Kumatani, Y. a., Liu, K. a., Wei, S. a., . . . Xuedong. (2021). UniSpeech: Unified Speech Representation Learning with Labeled and Unlabeled Data. *Proceedings of the 38th International Conference on Machine Learning, 139*, 10937-10947.

Yamashita, R., Nishio, M., Do, R. K., & Togashi, K. (2018). Convolutional neural networks: an overview and application in radiology. *Springer Open*(9), pages 611–629.

Younghak Shin, I. B. (2017). Comparison of hand-craft feature based SVM and CNN based deep learning framework for automatic polyp classification. *IEEE*, 3277-3280.

Yu, D., & Deng, L. (2015). *Automatic Speech Recognition A Deep Learning Approach.* London: Springer London.