FEATURE FUSION DEEP LEARNING METHOD FOR VIDEO AND AUDIO BASED EMOTION RECOGNITION

by

Yanan Song

A Thesis

Submitted to the Faculty of Purdue University

In Partial Fulfillment of the Requirements for the degree of

Master of Science in Electrical and Computer Engineering

Department of Electrical and Computer Engineering

Hammond, Indiana

December 2021

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF COMMITTEE APPROVAL

Dr. Lizhe Tan, Chair

Department of Electrical and Computer Engineering

Dr. Colin Elkin

Department of Electrical and Computer Engineering

Dr. Xiaoli Yang

Department of Electrical and Computer Engineering

Dr. Chenn Zhou

Department of Mechanical and Civil Engineering

Approved by:

Dr. Lizhe Tan

ACKNOWLEDGMENTS

Firstly, I would like to express my gratitude to my advisor and committee chair Dr. Lizhe Tan for his helpful instruction and guidance during the period of my Master's thesis research at Purdue University Northwest.

Secondly, I sincerely appreciate my family, especially my parents' financial support and encouragement while studying.

I am truly grateful to Dr. Chenn Zhou for funding me as a research assistant at the Center for Innovation through Visualization and Simulation (CIVS).

I would also thank my friend Zhankun Luo for giving me a lot of help and suggestions for my academic and personal difficulties.

Lastly, many thanks go to other thesis committee members, Dr. Colin Elkin, Dr. Xiaoli Yang and Dr. Chenn Zhou for their guidance and valuable comments.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

In this thesis, we proposed a deep learning based emotion recognition system in order to improve the successive classification rate. We first use transfer learning to extract visual features and use Mel frequency Cepstral Coefficients(MFCC) to extract audio features, and then apply the recurrent neural networks(RNN) with attention mechanism to process the sequential inputs. After that, the outputs of both channels are fused into a concatenate layer, which is processed using batch normalization, to reduce internal covariate shift. Finally, the classification result is obtained by the softmax layer. From our experiments, the video and audio subsystem achieve 78% and 77% respectively, and the feature fusion system with video and audio achieves 92% accuracy based on the RAVDESS dataset for eight emotion classes. Our proposed feature fusion system outperforms conventional methods in terms of classification prediction.

1. INTRODUCTION

1.1 Literature Review

Today, due to rapid development of artificial intelligence, emotion recognition systems are widely used in mobile applications, human-computer interaction, criminal investigation, mental health disease diagnosis, and interactive gaming. At present, the accuracy of emotion recognition has already exceeded that of thorugh human beings.

There are several approaches to recognize emotions. Emotion recognition using speech can be found in [1][2][3][4][5]. Emotion recognition using facial images is reported in [6][7] . Emotion recognition using physiological information like electroencephalogram (EEG) from brain signals can be found in [8]. For further improvement of prediction accuracy of the emotions, a multi-modal method, where emotion expressions are combined with voice and visual data, is proposed to achieve more efficient and accurate classification accuracy[9][10].

1.2 Motivation

Studies on facial expression recognition (FER) and speech emotion recognition have been done independently over the years. It is known that using, a single modality such as speech or facial expression may not correctly detect emotions in some scenarios. Therefore, a multi-modal emotion recognition system using a feature fusion method can be applied and further investigated. Attention mechanism has been successfully and widely used in neural language process[11] and computer vision[12]. Similarly, for the video and audio data, we can apply a recurrent neural network (RNN) to process the sequential information at each time step, and then use the attention model to assign different weights to obtain the weighted average as the input of RNN to improve classification rate of the emotions.

1.3 Thesis Scope

In this study, we investigate the recent trend of deep learning techniques, including transfer learning, batch normalization, optimization algorithms, and popular neural network structures. We design a video subsystem and an audio subsystem for single modality emotion

recognition. Next, we adopt a feature fusion using the video and audio data to develop a new multi-modal system to improve the emotion recognition accuracy. Finally, we compare the performance of our developed systems with the systems from the other researchers' methods using the same dataset.

1.4 Contribution of Thesis

We used transfer learning for visual feature extraction and GRU, and developed the video subsystem.

We used MFCC for audio feature extraction and GRU, and developed the audio subsystem.

We applied concatenative feature fusion with video and audio subsystems, and developed a multi-modal emotion recognition system.

We adopted attention mechanism with GRU in both subsystems to further improve the emotion recognition accuracy.

1.5 Organization of Thesis

This thesis is organized into following.

Chapter 1 Introduction: this chapter conducts literature review and briefly introduces the recent research and applications of emotion recognition.

Chapter 2 Background: this chapter reviews recurrent neural network, convolutional neural network, transfer learning, MFCC, and attention mechanism. Then, the methods of visual and audio feature extraction are developed.

Chapter 3 Methodology: In this chapter we describe details about thesis research, environment setup.

Chapter 4 Evaluation and Result: this chapter discuss our proposed system performance and accuracy, and compares with the other state-of-the-art methods.

Chapter 5 Conclusion and future work: we finally present the summary of this study and recommendations for our future work.

2. BACKGROUND

2.1 Transfer Learning

2.1.1 Introduction of Transfer Learning

Transfer Learning is a popular method in deep learning. It is a process of transferring the learned model parameters to a new model to solve another problem. For the tasks in related fields, transfer learning allows sharing the learned model parameters to a new model to speed up the learning efficiency of the model without learning from scratch [13]. As shown in Figure 2.1, we train the model from scratch for task 1, obtain a pre-trained model and freeze the layers, then add trainable layers after the pre-trained model to train the new layers for task 2.

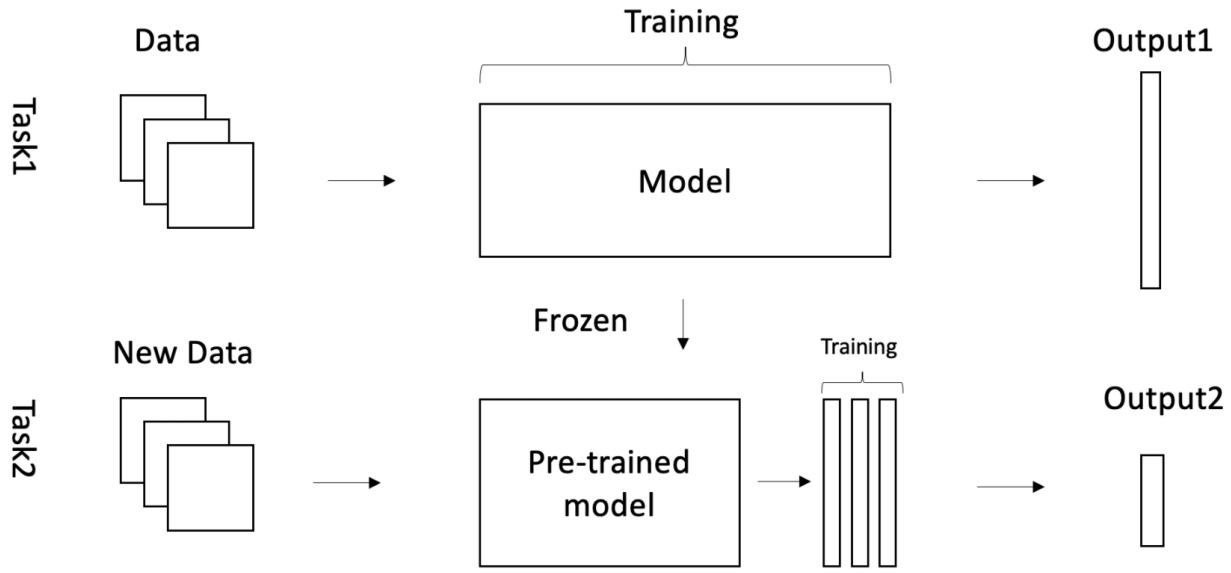


Figure 2.1. Transfer learning

2.1.2 Neural Network Models

VGGNet

VGGNet is a popular method for transfer learning. VGG is a model proposed by K. Simonyan and A. Zisserman from the University of Oxford[14]. VGG uses multiple convolutional layers with 3×3 convolution kernels instead of larger 7×7 kernels as used in

Table 2.1. VGGNet architecture

ConvNet ConFigureuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input(244 × 244 RGB image)					
conv3-64	conv3-64 LRN	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 conv1-256	conv3-256 conv3-256 conv3-256	conv3-256 conv3-256 conv3-256 conv3-256
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 conv1-512	conv3-512 conv3-512 conv3-512	conv3-512 conv3-512 conv3-512 conv3-512
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 conv1-512	conv3-512 conv3-512 conv3-512	conv3-512 conv3-512 conv3-512 conv3-512
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

AlexNet[15], which reduces the parameters and increased the expressing ability of the network. The architecture of VGG is shown in Table 2.1.

2.2 Architecture of Multilayer Perceptron (MLP)

A multilayer perceptron introduces one or more hidden layers on the top of the single-layer neural network. The hidden layer is located between the input layer and the output

layer[16]. Figure 2.2 shows a typical multi-layer perceptron (MLP) with an input layer, two hidden layers, and an output layer.

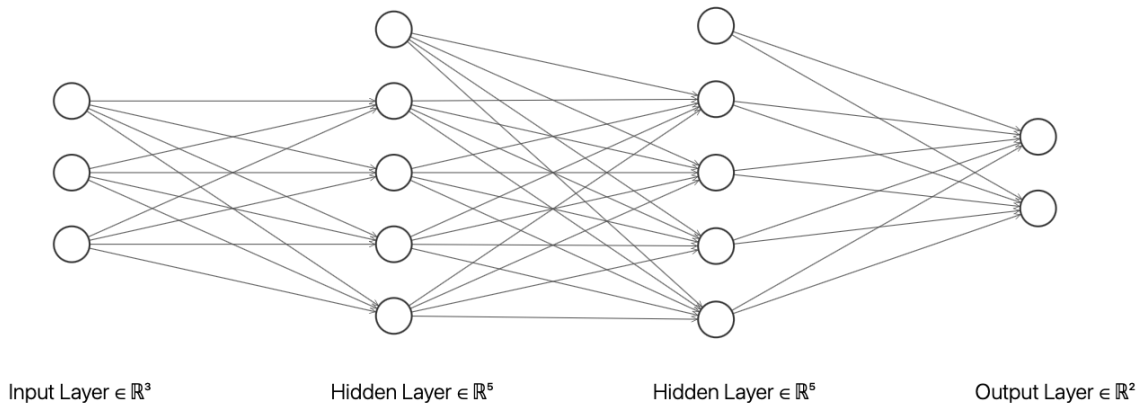


Figure 2.2. Multi-layer perception with 2 hidden layers

2.2.1 Activation Function

The activation function is required in the multilayer perceptron neural network. Consider a multi-layer perceptron neural network without activation. We denote input layer as X_{i} , hidden layer(s) as H_{j}, output layer as Y_k, weights as W, and bias as b, respectively. The hidden layer can be calculated as[16]:

$$H = XW_{\mathrm{j}} + b_{\mathrm{j}} \tag{2.1}$$

The output layer can be calculated as:

$$Y = HW_k + b_k \tag{2.2}$$

$$Y = (XW_{\mathrm{j}} + b_{\mathrm{j}})W_k + b_k = XW_{\mathrm{j}}W_k + b_{\mathrm{j}}W_k + b_k \tag{2.3}$$

As shown in Equation 2.3, the neural network without an activation function is equal to linear regression, adding more hidden layers does not make any difference. The following common activation functions are used.

Sigmoid Function

Sigmoid Function is defined as[16]:

$$f(x) = sigmoid(x) = \frac{1}{1 + \mathrm{e}^{-x}} \tag{2.4}$$

Figure 2.3 shows the plot of the sigmoid function. The derivative of the sigmoid function is

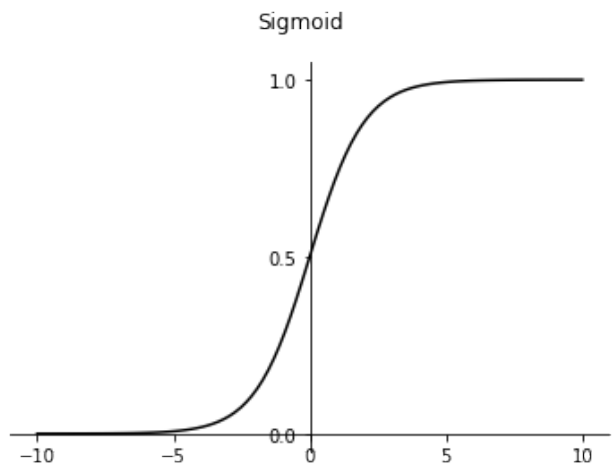


Figure 2.3. Sigmoid function

given below:

$$f'(x) = f(x)(1 - f(x) \tag{2.5}$$

Tanh Function

Tanh function is defined as[16]:

$$f(x) = tanh(x) = \frac{2}{1 + \mathrm{e}^{-2x}} - 1 \tag{2.6}$$

Tanh is plotted in Figure 2.4, the derivative of the Tanh function is expressed in Equation

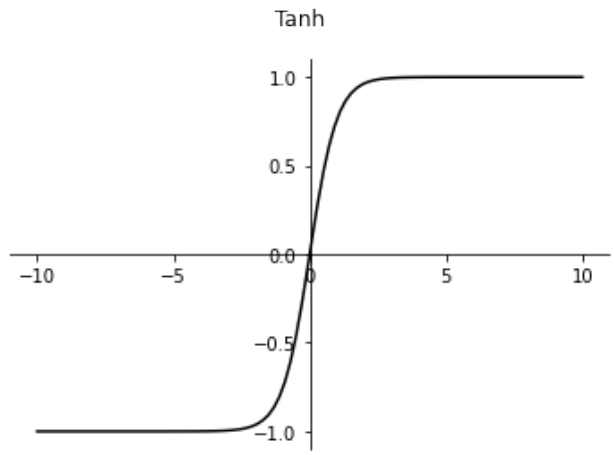


Figure 2.4. Tanh function

2.4:

$$f'(x) = 1 - f(x)^2 \tag{2.7}$$

Softmax Function

Softmax function is defined as[16]:

$$f(x)_{\mathrm{i}} = \frac{\mathrm{e}^{x_{\mathrm{j}}}}{\sum_{\mathrm{j}=1}^{K} \mathrm{e}^{x_{\mathrm{j}}}} \tag{2.8}$$

Rectified Linear Unit(ReLU)

The Rectified linear unit is shown in the Figure 2.5 and defined as[16]:

$$f(x) = \begin{cases} x & \text{if } x < 0 \\ 0 & \text{if } x \geq 0 \end{cases} \tag{2.9}$$

The derivative of the ReLU function is expressed as :

$$f'(x) = \begin{cases} 0 & \text{if } x < 0 \\ 1 & \text{if } x \geq 0 \end{cases} \tag{2.10}$$

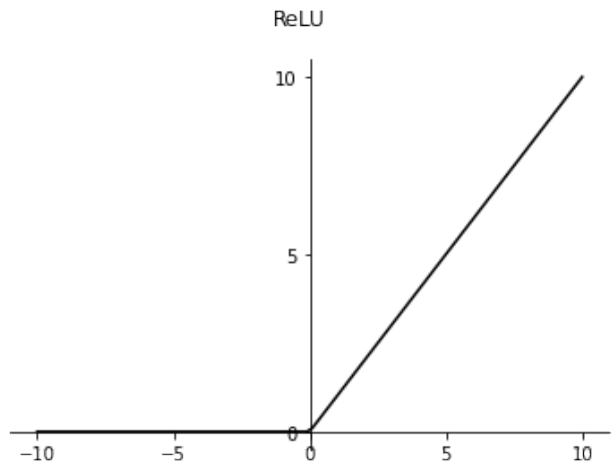


Figure 2.5. ReLU

ReLU function has better efficiency and is less computationally expensive when comparing with the sigmoid and tanh function.

2.3 Loss Function

To train the neural networks, loss function is used to estimate the inconsistency between the prediction value and the ground truth value. The smaller the loss function, the better performance of the model[17].

2.3.1 Cross Entropy

To calculate the error of the classification problem, we use the cross-entropy as our loss function. For two discrete probability distributions p and q, the cross-entropy is defined as follows[17]:

$$Cross\ Entropy\ Loss = -\sum_{\mathrm{i}=1}^{C} y_{\mathrm{i}} \cdot log(\hat{y}_{\mathrm{i}}) \tag{2.11}$$

The derivative of the cross-entropy loss function for the softmax function is shown below:

$$\frac{\partial \xi}{\partial z_{\mathrm{i}}} = y_{\mathrm{i}} - \hat{y}_{\mathrm{i}} \tag{2.12}$$

where ξ denotes cross-entropy loss function, z_{i} denotes softmax input, y_{i} denotes ground truth, $\hat{y}_{\mathrm{i}}$ denotes output probability, C denotes output size. Note that the softmax function is described in Equation 2.8.

2.4 Architecture of Convolutional Neural Networks(CNN)

2.4.1 Convolution Layer

Convolution Operation

For a 2 dimension signal I, convolution kernel K, convolution operation at (u, v) defined in Equation 2.13.

$$(I * K)(u, v) = \sum_{\mathrm{i}} \sum_{\mathrm{j}} I(\mathrm{i}, \mathrm{j}) \cdot K(u - \mathrm{i}, v - \mathrm{j}) \tag{2.13}$$

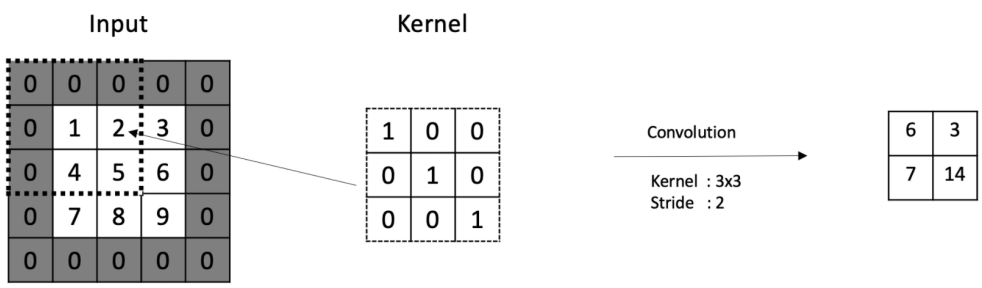


Figure 2.6. Convolution operation

Padding

For convolution operation, padding is usually added due to the fact that pixels on the edge will never be located in the center of the convolution kernel after the input is processed by the convolution operation, the part of the information will be lost. To extend the area of a convolution neural network process, add padding pixels to the border to allow convolution kernel scanning out of the original edge. The value of padding pixels is usually set to zero. The padding operation example is shown in Figure 2.7.

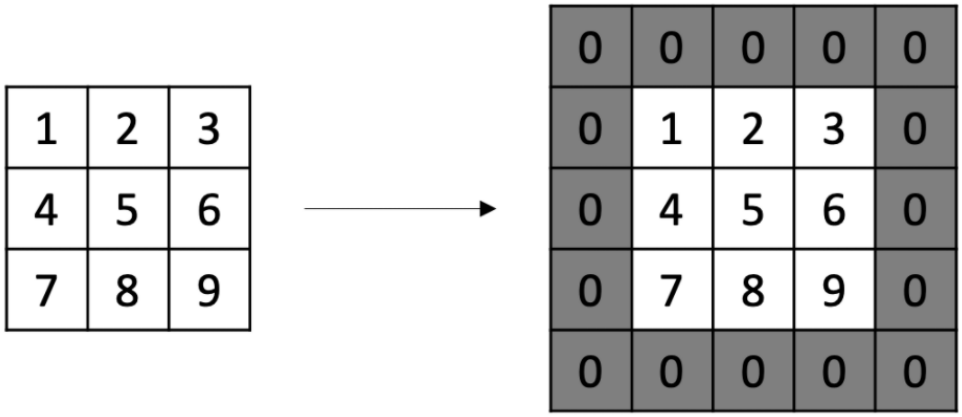


Figure 2.7. Padding

2.4.2 Activation Layer

Similar to the multi-layer perceptron, convolutional neural networks also need activation functions to introduce non-linear transformation. The common choices of activation functions are sigmoid, tanh, ReLU, etc.

2.4.3 Pooling Layer

The pooling operation samples the features contained within a sub-region of the feature map. The main purpose of the pooling layer is to reduce the dimensions of the feature map, the number of parameters, and computation in the neural network. The common polling methods include the maximum polling, average polling, and global polling, which will be described below.

Max Pooling

Max pooling operation selects the maximum element in the specified filter. An example of Max pooling operation with 2×2 pool size and 2 strides is shown in Figure 2.8.

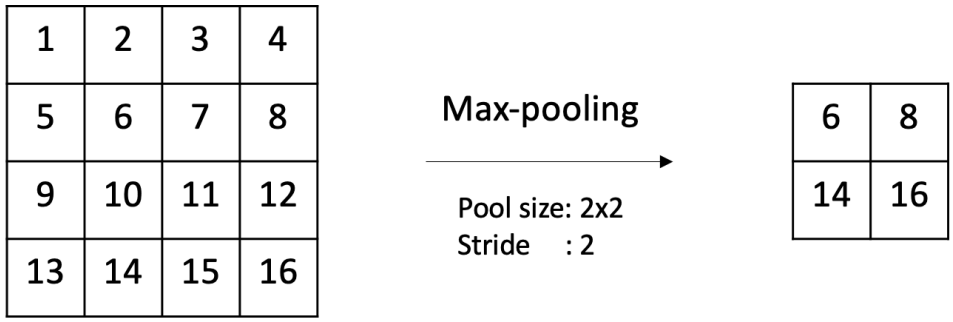


Figure 2.8. Max-pooling

Average Pooling

Average pooling operation computes the average of all elements in the filter, An example of Average pooling operation with 2×2 pool size and 2 strides is shown in Figure 2.9.

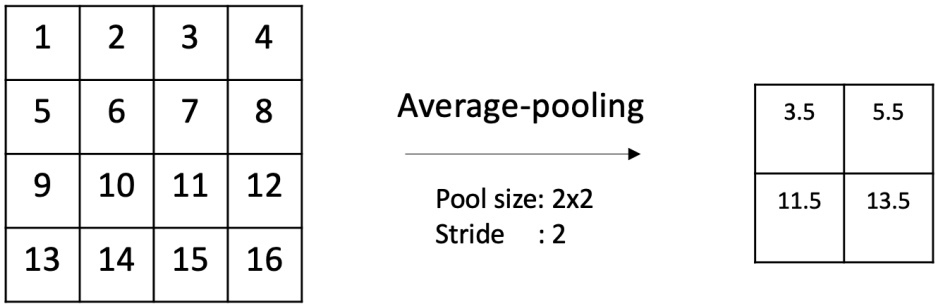


Figure 2.9. Average-pooling

Global Pooling

Global pooling operation does the pooling operation in the whole channels. It can be either global max pooling or global average pooling, as shown in Figure 2.10.

2.4.4 Fully-connected Layer

Similar to the multi-layer perception, fully-connected layers in the convolutional neural networks have full connections to the previous activation layer for nonlinear mapping.

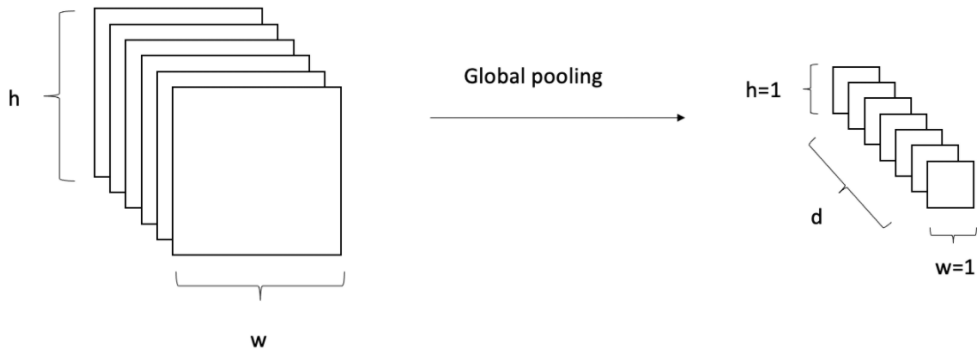


Figure 2.10. Global Pooling

2.4.5 Architecture of Recurrent Neural Network(RNN)

Long Short-term Memory(LSTM)

Recurrent neural network(RNN) suffers from vanishing and exploding gradients. Long short term memory network solves this problem by utilizing input gate i_t, forget gate f_t, output gate o_t and memory cells $\tilde{c}_t$. as defined in the Figure 2.11 and Equations 2.14 to 2.19 [18]:

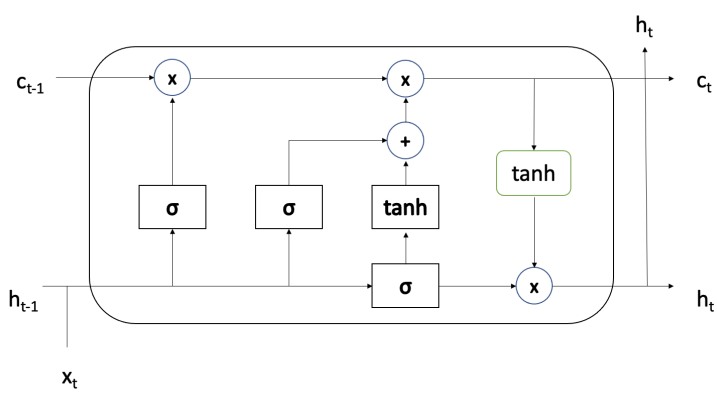


Figure 2.11. LSTM Architecture

$$f_t = \sigma_g(W_f x_t + U_f h_{t-1} + b_f) \tag{2.14}$$

$$\mathrm{i}_t = \sigma_g(W_\mathrm{i} x_t + U_\mathrm{i} h_{t-1} + b_\mathrm{i}) \tag{2.15}$$

$$o_t = \sigma_g(W_o x_t + U_o h_{t-1} + b_o) \tag{2.16}$$

$$\hat{c}_t = \sigma_c(W_c x_t + U_c h_{t-1} + b_c) \tag{2.17}$$

$$c_t = f_t \circ c_{t-1} + \mathrm{i}_t \circ \hat{c}_t \tag{2.18}$$

$$h_t = o_t \circ \sigma_h(c_t) \tag{2.19}$$

Gate Recurrent Unit(GRU)

Gated recurrent unit (GRU) is a type of recurrent neural network (RNN) introduced in [19]. A GRU operates using an update gate and reset gate. The update gate controls how much past information pass to the next states while the reset gate controls how much past information to forget. The difference between LSTM and GRU is that GRU uses hidden state to the transfer function and does not have a cell state. The GRU only has two gates, reset gate r_t and update gate z_t. The structure of GRU and operation equations are depicted in the Figure 2.12 and Equations 2.20 to 2.23 [18]:

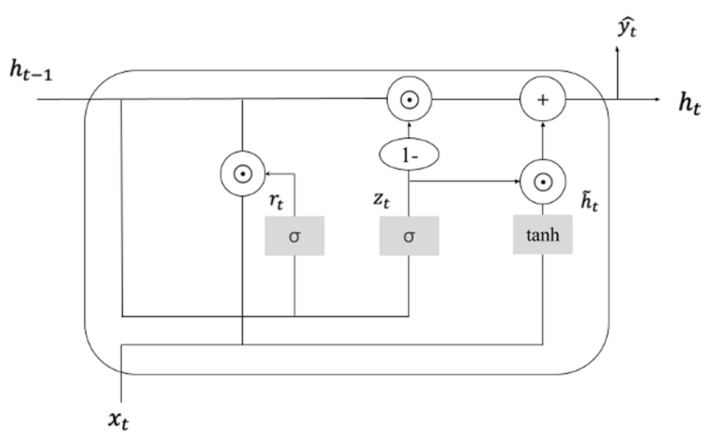


Figure 2.12. GRU architecture

$$h_t^{\mathrm{j}} = (1 - z_t^{\mathrm{j}}) \odot h_{t-1}^{\mathrm{j}} + z_t^{\mathrm{j}} \odot \hat{h_t}^{\mathrm{j}} \tag{2.20}$$

$$z_t^{\mathrm{j}} = \sigma(W_z x_t + U_z h_{t-1})^{\mathrm{j}} \tag{2.21}$$

$$\hat{h_t^{\mathrm{j}}} = tanh(W x_t + U(r_t \odot h_{t-1}))^{\mathrm{j}} \tag{2.22}$$

$$r_t^{\mathrm{j}} = \sigma(W_r x_t + U_r h_{t-1})^{\mathrm{j}} \tag{2.23}$$

Note that σ denotes the sigmoid function, x_t denotes input vector, h_t denotes output vector, $\hat{h}_t$ denotes candidate activation vector, W, U and b denotes parameter matrices and vector, and $\odot$ denotes the element-wise multiplication.

2.5 Audio/Speech Features of MFCC

In the speech processing area, the most commonly used voice feature is the Mel-scale frequency cepstral coefficients(MFCC)[20]. In the research of human hearing mechanisms, the human has different hearing sensitivity to different sound frequencies ranging from 200 Hz to 5000 Hz. The MFCC computation consists of the following steps:

2.5.1 Pre-emphasis

The first step of getting the MFCC feature is passing the voice signal via a high-pass filter. Pre-emphasis is used to boost the high-frequency section of a signal and flatten the frequency spectrum of the signal, it also increases the Signal-to-Noise Ratio (SNR). The high-pass filter is defined as:

$$H(z) = 1 - a \times z^{-1} \tag{2.24}$$

Note that the constant α denotes pre-emphasis coefficient.

2.5.2 Frame Blocking

This step divides the audio signal into short frames with overlapping between adjacent audio.

2.5.3 Windowing

To increase the continuity between each frame, adjacent audio frames are multiplied by a Hamming window [21]. The Hamming window shown in Figure 2.13 is defined as:

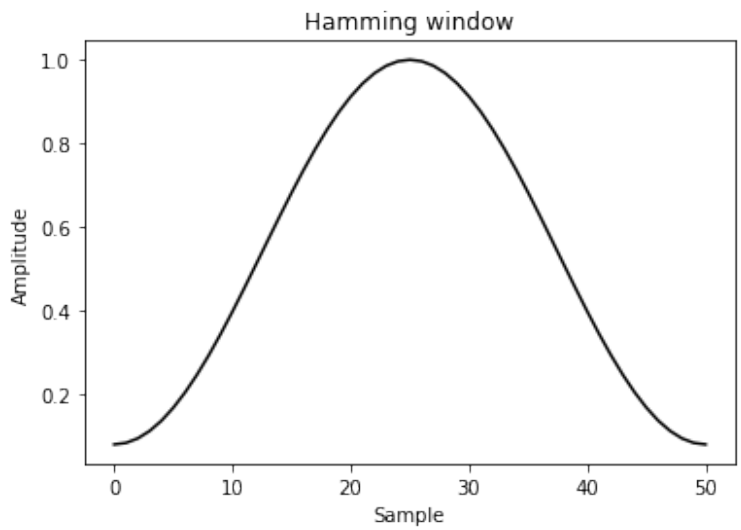


Figure 2.13. Hamming Window

$$w(n) = 0.54 - 0.46\cos(2\pi\frac{n}{N}), \quad 0 \leq n \leq N. \tag{2.25}$$

2.5.4 Fast Fourier Transform

Since the transformation of the signal in time domain is usually difficult to see the characteristics of the signal, it is necessary to retrieve the energy distribution in the frequency domain. The energy distributions can effectively represent the characteristics of voice features.

2.5.5 Triangular Bandpass Filters

The audio is passed through a set of triangular bandpass filters, which is a filter bank with M filters, to get the log energy. Converting from frequency to Mel scale is defined as:

$$mel(f) = 2595 \times log_{10}(1 + \frac{f}{700}) \tag{2.26}$$

Figure 2.14 displays the frequency responses of the Mel filter bank.

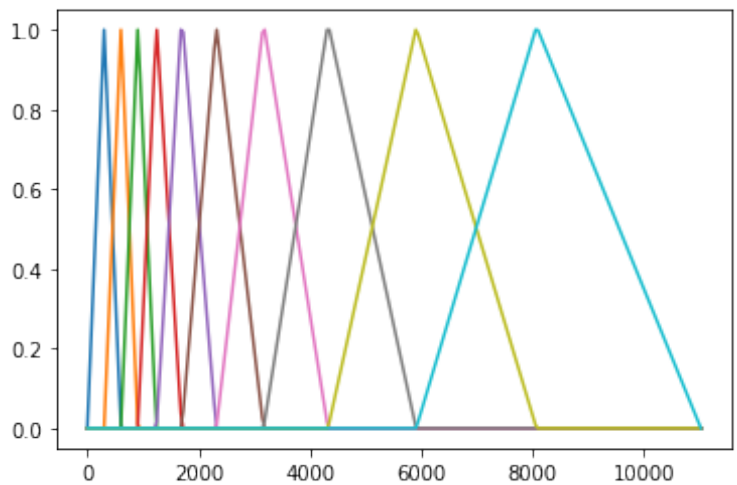


Figure 2.14. Mel filter bank

2.5.6 Discrete Cosine Transform

MFCC takes the log energy into the discrete cosine transform to get the mel-scale cepstrum of order L. The discrete cosine transform is defined as:

$$C(n) = \sum_{N-1}^{m=0} s(x) \cos(\frac{\pi n(m - 0.5)}{M}), \quad n = 1, 2, ..., L \tag{2.27}$$

2.6 Attention Mechanism

The attention mechanism in neural networks is similar to attention in humans. At present, the attention mechanism has become one of the most widely used methods in the

field of natural language processing, and computer vision, as examples in BERT[22], Transformer[23] etc. There are two types of attention: soft attention[24] and hard attention[25].

2.6.1 Soft Attention

Soft attention is deterministic, given an input x, soft attention discredits irrelevant areas by multiplying a smaller attention score and high attention area multiplying a greater attention score. To compute the attention score s_{i} for the input x_{i} , we have:

$$s_{\mathrm{i}} = tanh(W_c h_{t-1} + W_x x_{\mathrm{i}}) \tag{2.28}$$

We use softmax normalization to compute attention weight a_{i}.

$$a_{\mathrm{i}} = softmax(s_{\mathrm{i}}) = \frac{\mathrm{e}^{s_{\mathrm{i}}}}{\sum \mathrm{e}^{s_{\mathrm{i}}}} \tag{2.29}$$

Then, the weighted average of x_{i} is computed as.

$$Z = \sum a_{\mathrm{i}} x_{\mathrm{i}} \tag{2.30}$$

2.6.2 Hard Attention

Hard attention only focuses on one region stochastically. For the given input x_{i}, the hard attention uses attention weight a_{i} to stochastic sampling as follows:

$$Z \sim x_{\mathrm{i}}, a_{\mathrm{i}}$$

2.7 Batch Normalization

During the deep neural network training stage, the process of changing the distribution of internal nodes due to changes in the parameters of the network is called the internal covariate shift. Internal covariate shift can cause the following problems:

1. Upper layers network needs to constantly adjust to adapt to change of input data disruption, resulting in decreased learning speed.

2. When using the saturated activation functions in neural networks, such as sigmoid and tanh, it is easy for the model training to fall into the saturated regime. When this happens, the gradient will become very small, which will slow down the convergence speed. To solve those problems, we use batch normalization to fix the means and variances of each layer's input. The batch normalization is conducted using Equations 2.31 to 2.34. First, we calculate the mean and variance for the mini-batch B[26].

$$\mu_B = \frac{1}{m}\sum_{\mathrm{i}=1}^{m} x_{\mathrm{i}} \tag{2.31}$$

$$\sigma_B^2 = \frac{1}{m}\sum_{\mathrm{i}=1}^{m} (x_{\mathrm{i}} - \mu_B)^2 \tag{2.32}$$

Then, we normalize x_{i}:

$$\hat{x}_{\mathrm{i}} = \frac{x_{\mathrm{i}} - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}} \tag{2.33}$$

Finally, the output from the batch normalization is given by:

$$y_{\mathrm{i}} = \gamma \hat{x}_{\mathrm{i}} + \beta \equiv BN_{\gamma,\beta}(x_{\mathrm{i}}) \tag{2.34}$$

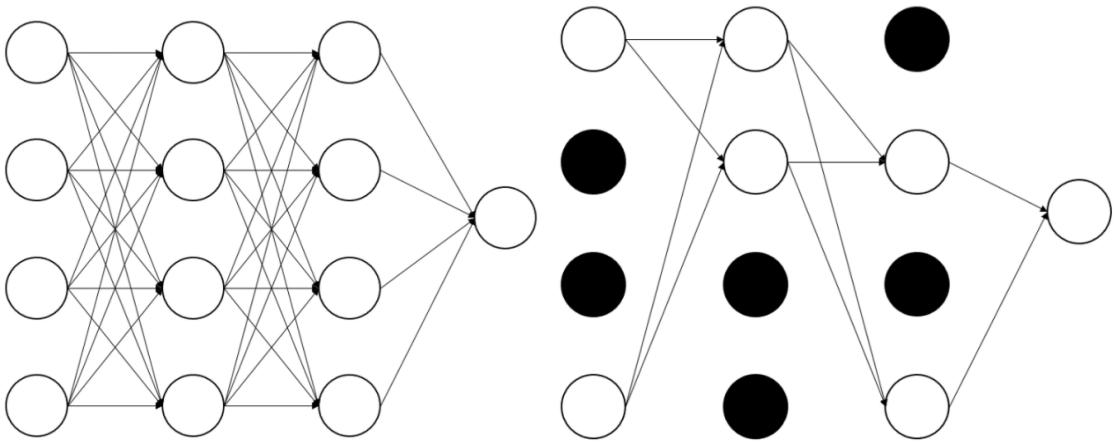
where m denotes the batch size, ϵ denotes a positive constant for numerical stability.

2.8 Dropout

In machine learning, if a large model is trained on relatively a small number of training samples, the trained model is prone to be over-fitting. Over-fitting is specifically manifest in the model which has a small error on the training data but the error on the test data is relatively large[27].

Dropout is randomly ignoring a certain percentage of neurons during the training stage, which makes the model more generalized and does not rely too much on some local fea-

tures. Consider a neural network with L hidden layers $l \in 1, ..., L$, forward propagation with dropout operation defined as Equation 2.35 to 2.38. z^l denote the input to layer l, y^l denote the output from the layer l, f denote the activation function. Figure 2.15 shows an example of dropout operation in a 1 input layer, 2 hidden layers, 1 output layer neural network with 0.5 dropout probability.



(a) Neural network without dropout (b) Neural network with dropout

Figure 2.15. Dropout operation

$$r_{\mathrm{j}}^{(l)} \sim Bernouilli(p) \tag{2.35}$$

$$\tilde{y}^{(l)} = r^{(l)} * y^{(l)} \tag{2.36}$$

$$z_{\mathrm{i}}^{(l+1)} = w_{\mathrm{i}}^{(l+1)} \tilde{y}^{l} + b_{\mathrm{i}}^{(l+1)} \tag{2.37}$$

$$y_{\mathrm{i}}^{(l+1)} = f(z_{\mathrm{i}}^{(l+1)}) \tag{2.38}$$

2.9 Optimization Algorithm

2.9.1 Gradient Descent

Gradient descent is a first-order optimization for finding local minimum. Let f be a continuously differentiable function, f' is the derivative of f, . Then, define a small constant $\alpha > 0$ as the learning rate [28].

$$f(x + \epsilon) \simeq f(x) + \epsilon f'(x) \tag{2.39}$$

replacing ϵ by the gradient of function f at x: $f'(x)$ leads to.

$$f(x - \alpha f'(x)) \simeq f(x) - \alpha f'(x)^2 \tag{2.40}$$

if $f'(x) \neq 0$, $\eta f'(x)^2 > 0$.

$$f(x) - \alpha f'(x) \leq f(x) \tag{2.41}$$

Then, use Equation 2.42

$$x \leftarrow f(x) - \eta f'(x) \tag{2.42}$$

to iterate for finding the local minimum until the stop condition is reached. For a training dataset with n samples, the gradient of the loss function denoted by f, that is the objective function, can be calculated as:

$$\nabla f(x) = \frac{1}{n} \sum_{\mathrm{i}=1}^{n} \nabla f_{\mathrm{i}}(x) \tag{2.43}$$

2.9.2 Stochastic Gradient Descent(SGD)

Different from gradient descent, stochastic gradient descent does not calculate gradient for all training samples, stochastic gradient descent computes the gradient at a random

point for each iteration to reduce the computational cost. The objective function J of the stochastic gradient is calculated using Equation 2.44 [28]:

$$\theta = \theta - \eta \cdot \nabla_\theta J(\theta; x^{(\mathrm{i})}; y^{(\mathrm{i})}) \tag{2.44}$$

where x denotes training example, y denotes label, θ denotes model's parameter.

2.9.3 Adam

Adam is an first-order gradient-based optimization algorithm. It does an exponentially weighted moving average of small batch stochastic gradients based on the RMSProp algorithm[29]. Adam defined in Equations 2.46 to 2.50, α denotes step size, $\beta_1, \beta_2 \in [0, 1]$ is exponential decay rates for the moment estimates. f denotes stochastic objective function. Default setting: $\alpha = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$,$\beta_1\beta_2 \in [0, 1)$, and $\epsilon = 10^{-8}$ is recommended.

$$g_t \leftarrow \nabla_\theta f_t(\theta_{t-1}) \tag{2.45}$$

$$m_t \leftarrow \beta_1 \cdot v_{t-1} + (1 - \beta_2) \cdot g_t \tag{2.46}$$

$$v_t \leftarrow \beta_2 \cdot v_{t-1} + (1 - \beta_2) \cdot g_t^2 \tag{2.47}$$

$$\hat{m}_t \leftarrow m_t/(1 - \beta_1^t) \tag{2.48}$$

$$\hat{v}_t \leftarrow v_t/(1 - \beta_2^t) \tag{2.49}$$

$$\theta_t \leftarrow \theta_{t-1} - \alpha \cdot \hat{m}_t/(\sqrt{\hat{v}_t} + \epsilon) \tag{2.50}$$

The algorithm keeps iterating until the stop condition is reached[30].

3. DEVELOPMENT OF VIDEO AUDIO BASED DEEP LEARNING NEURAL NETWORKS

3.1 Experiment environment

Colaboratory known as ”Colab”, it was developed by Google Research team. Colab is adopted for our research. Using Colab, user can edit and execute Python code, for machine learning projects, data analysis, and education purpose. Colab has hosted a Jupyter notebook service that user can run their code without setting up the environment. Colab also offers free Graphics processing units(GPU)[31] and Tensor Processing Unit(TPU)[32] computing resources.

3.2 Dataset

3.2.1 FER2013

The FER2013 dataset has approximately $30,000$, 48×48 pixel grayscale images of 7 types emotions: angry, disgust, fear, happy, sad, surprise and neutral[33].

Figure 3.1. JAFFE example

3.2.2 JAFFE

The JFAAE(Japanese Female facial expression) data set has 213 images from 10 Japanese female subjects. Each person makes 7 expressions : sad, happy, angry, disgust, surprise, fear, neutral[34].

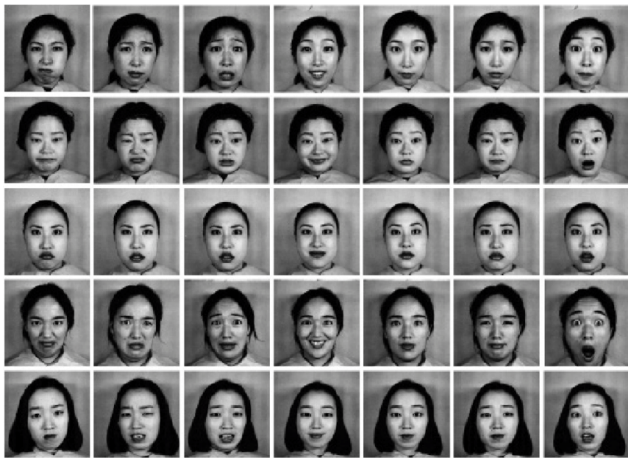


Figure 3.2. FER2013 example

3.2.3 RAVDESS

The Ryerson Audio-Visual Database [35] of Emotional Speech and Song (RAVDESS), which contains 7356 files, including audio and visual data from 24 actors consisting of 12 males and 12 females. Speech data contains eight emotions, that is, neutral, calm, happy, sad, angry, fearful, disgust, and surprise whereas song data contains calm, happy, sad, angry, and fearful emotions. We only used the speech-video data in this thesis, which comprises 1440 audio-visual files (96 files for neutral, 192 calm, 192 happy, 192 sad, 192 angry, 192 fearful, 192 disgust, and 192 surprise). Each audio-visual file has video recording format with a scan resolution of 1920x1080 pixels at a frame rate of 30 frames per second (fps) and speech recording format at a sampling rate of 48 kHz at 16-bit resolution[36].

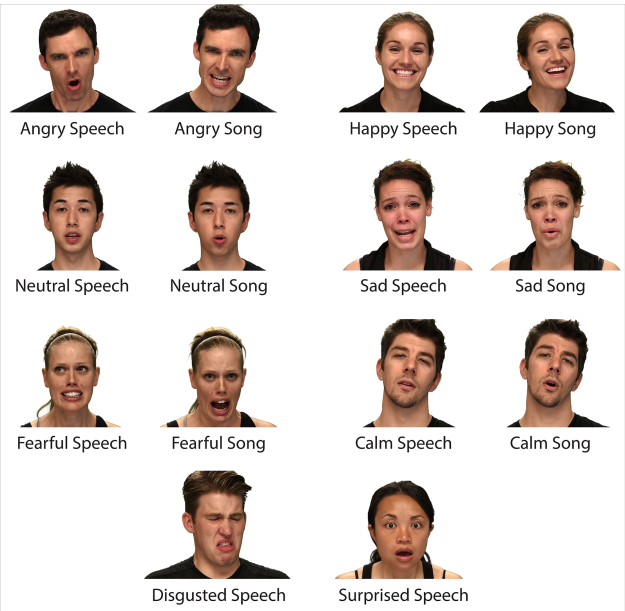


Figure 3.3. Example of RAVDESS data set

3.3 Training Process

The RAVDESS speech subset contains 1440 samples, we applied 10 fold cross-validation to evaluate our methods.

3.3.1 Cross-validation

We can separate the data set by K folds. we use each subset of data as a validation set, and use the remaining $K-1$ subsets of data as the training set, K models will be obtained as shown in Figure 3.4. Average of K validation error is the cross-validation error. Cross-validation effectively uses in the limited data and the evaluation result can be more accurate.

3.4 Neural Network Implementation

3.4.1 Feature Fusion Model

The feature fusion model contains two subsystems: video and audio. The video subsystem takes video as input, then perform face detection to video frames, and uses the pre-trained

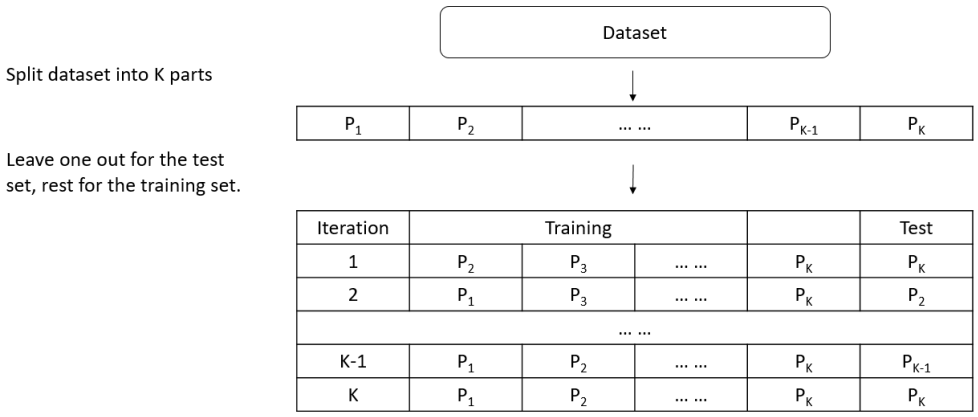


Figure 3.4. K-fold cross validation

VGG16 model to obtain features. Then we use gated recurrent unit(GRU) layer to combine with attention mechanism. Similarly, the audio subsystem takes audio signal as input, generates Mel frequency cepstral coefficients(MFCC) for audio features. The sequence of MFCC features is fed to the GRU layer combining with attention mechanism. The video and audio features from the fully connected layer of both subsystems are fused by concatenation operation. Finally, using batch normalization and dropout layer, we get the classification results by the softmax operation. The feature fusion model is shown in Figure 3.5.

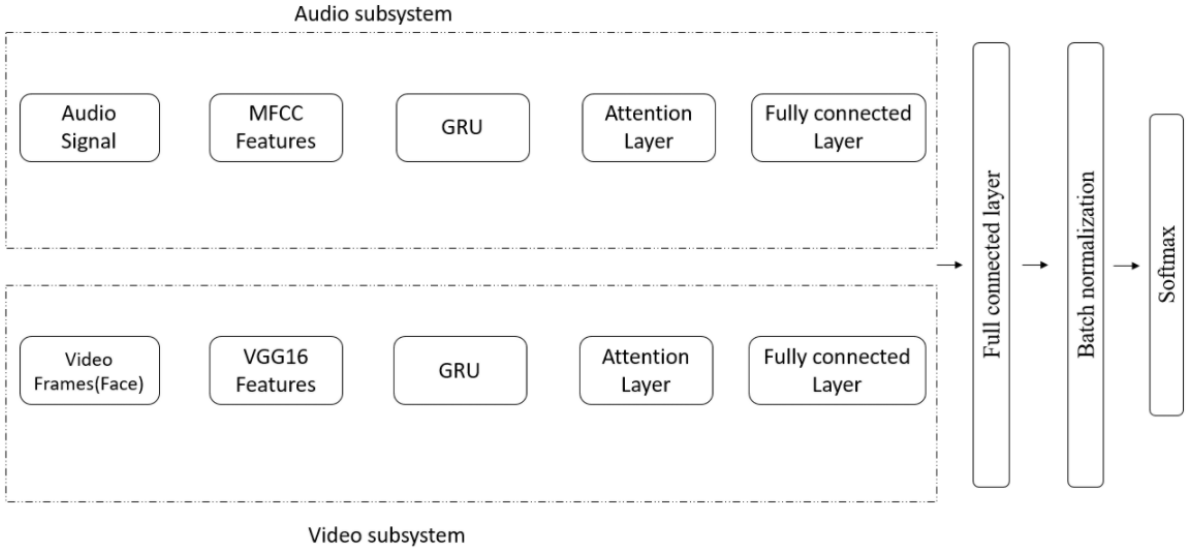


Figure 3.5. Feature Fusion model

3.4.2 Video Subsystem

The video subsystem is shown in Figure 3.6. For the pre-processing, the frame rate of the input video clips was reduced from 30 FPS to 7.5 FPS. Then, A face detector was applied to the video frames and resized to $48 \times 48 \times 3$. The VGG16-GRU subsystem has a pre-trained VGG16 model, one GRU layer followed by an attention layer, a fully connected layer, batch normalization layer, and softmax layer. The VGG16 model was described in Table 2.1. The transfer learning model was pre-trained on the ImageNet [37] dataset. During the training weights of all convolutional blocks in the VGG16 pre-trained model are frozen.

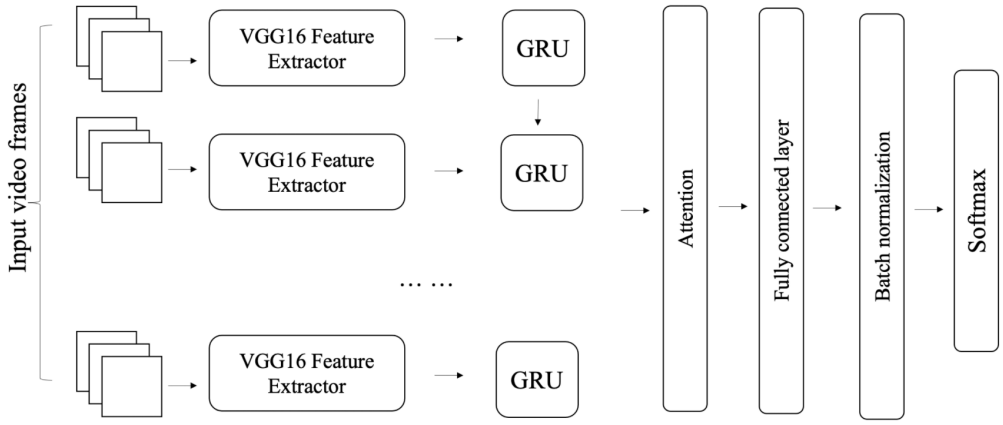


Figure 3.6. Video subsystem

3.4.3 Audio Subsystem

In the MFCC-GRU subsystem, the Mel frequency cepstral coefficients(MFCC) features are extracted at the pre-processing stage of the audio signal shown in the Figure 3.7. Thirteen MFCC coefficients of speech signals are calculated. The length of each speech signal clip is 4096. The details about obtaining MFCC features is described in Chapter2. We obtain MFCC features of audio clips as input features, similar to our video subsystem, the audio subsystem is cascaded with stacked GRU layers and attention layer, followed by a fully connected layer, batch normalization layer, and softmax layer.

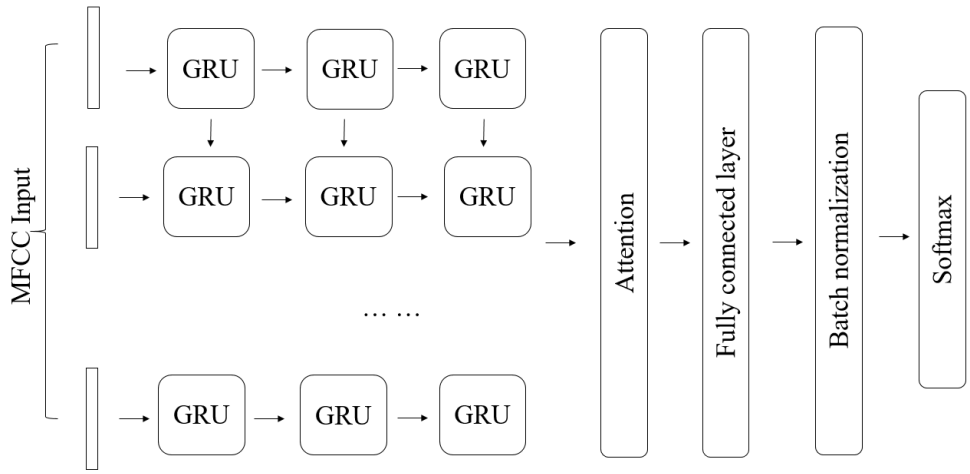


Figure 3.7. Audio subsystem

3.5 Performance Evaluation

3.5.1 Accuracy

Accuracy is the most common evaluation index and it is defined as:

$$Accuracy = \frac{\text{Number of correct prediction}}{\text{Total number of samples}} \tag{3.1}$$

Accuracy is a very intuitive evaluation index. However, in the situation of an unbalanced data set. High accuracy does not mean high performance. For example, given a data set that has $n\%$ positive samples and $1 - n\%$ negative samples, even the model predicts all samples as positive, the final accuracy will be higher than $n\%$. Therefore, using accuracy as the only metric to evaluate a machine learning model is not sufficient.

3.5.2 Confusion Matrix

Each column of the confusion matrix represents the predicted category, while each row represents the true category of the data. The sum of each row denotes the actual sample size of the category, the sum of each column denotes the number of samples predicted to be the category. Table 3.1 shows an example of confusion matrix.

Table 3.1. Confusion matrix example

		True		
		Category1	Category2	Category3
Predicted	Category1	-	-	-
	Category2	-	-	-
	Category3	-	-	-

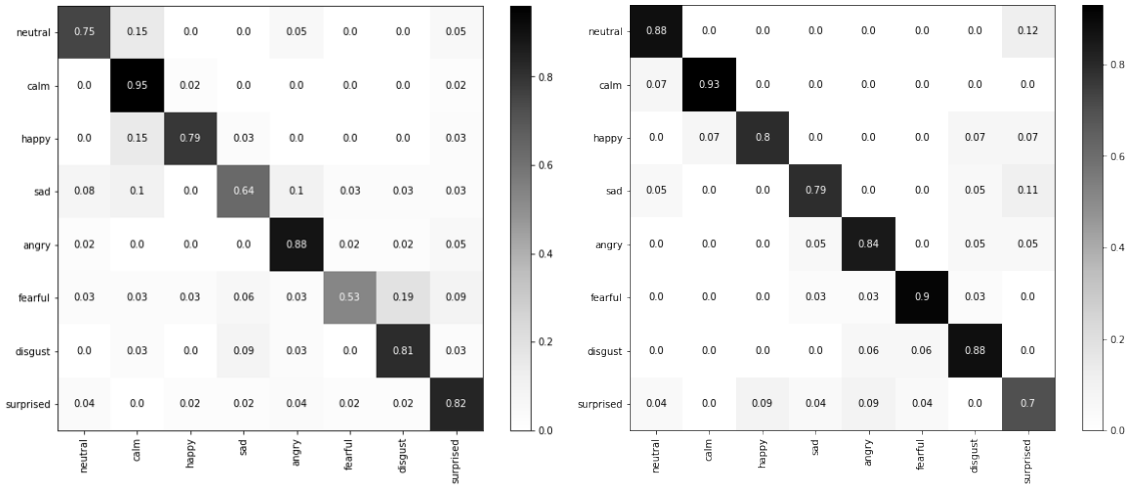
4. Performance Evaluation and Comparison

4.1 Video Subsystem Result

We used the video subsystem alone to test the performance of our system. The result is shown in Table 4.1, which contains the test accuracy of different hyper-parameters: number of stacked GRU layers, number of units in each GRU layer, the option for batch normalization, and the option for attention mechanism for GRU [36].

Table 4.1. Video Subsystem Result

GRU layers	GRU units	Batch normalization	Attention	Test accuracy
1	256	Yes	No	78%
3	256	Yes	No	74%
1	512	No	No	76%
1	512	Yes	No	79%
3	512	No	No	74%
3	512	Yes	No	76%
1	1024	No	No	77%
1	1024	Yes	No	77%



(a) Previous video subsystem result confusion matrix

(b) Video subsystem with attention result confusion matrix

Figure 4.1. Video subsystem results comparison

From the experiments shown above, VGG16 with 1-layer 512 units GRU obtains 79% accuracy, applying attention mechanism obtains 83% accuracy. The previous model confu-

Table 4.2. Video subsystem with attention result(10-Fold CV)

Average Accuracy	Best Accuracy	Lowest Accuracy	STD
78.19%	83.33%	73.61%	0.03

sion matrix is shown in the Figure 4.1a. The emotion "claim" has the highest accuracy of 95%, "fearful" has the lowest accuracy of 53%. There are 19% "fearful" was classified to disgust and 15% "neutral" was classified to "clam". After applied attention mechanism the confusion matrix is shown in the Figure 4.1b. The emotion "claim" has the highest accuracy of 93%, "surprised" has the lowest accuracy of 70%.

4.2 Audio Subsystem Result

The results from the audio subsystem with out applying attention mechanism shown in the Table 4.3[36].

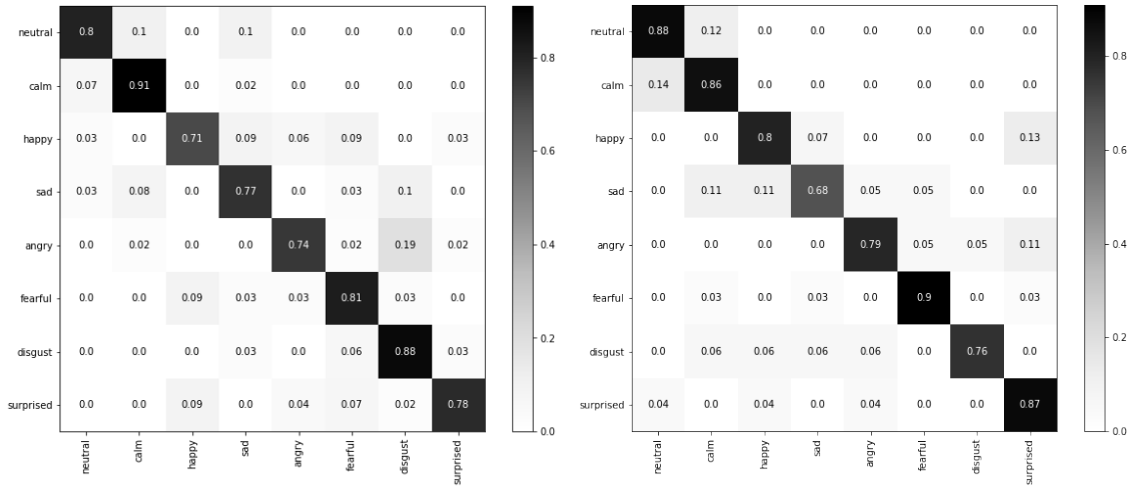
Table 4.3. Audio subsystem result

GRU layers	GRU units	Batch normalization	Attention	Test accuracy
1	256	No	No	69%
1	256	Yes	No	75%
3	256	No	No	75%
3	256	Yes	No	80%
3	512	No	No	75%
1	512	Yes	No	79%

Table 4.4. Audio subsystem result(10-Fold CV)

Average Accuracy	Best Accuracy	Lowest Accuracy	STD
76.52%	81.25%	68.75%	0.04

From the experiments shown in Table 4.3 , applying MFCC features with 3-layers and 256 units GRU and batch normalization obtains 80% accuracy. In Table 4.4, applying attention mechanism on our model obtains 76% average accuracy, 81% best accuracy, and 69% lowest accuracy and 0.04 standard deviation. The accuracy of emotions "neutral", "happy", "sad", "fearful", and "disgust" are increased , while the accuracy of other emotions is decreased.



(a) Previous audio subsystem result confusion matrix

(b) Audio subsystem with attention result confusion matrix

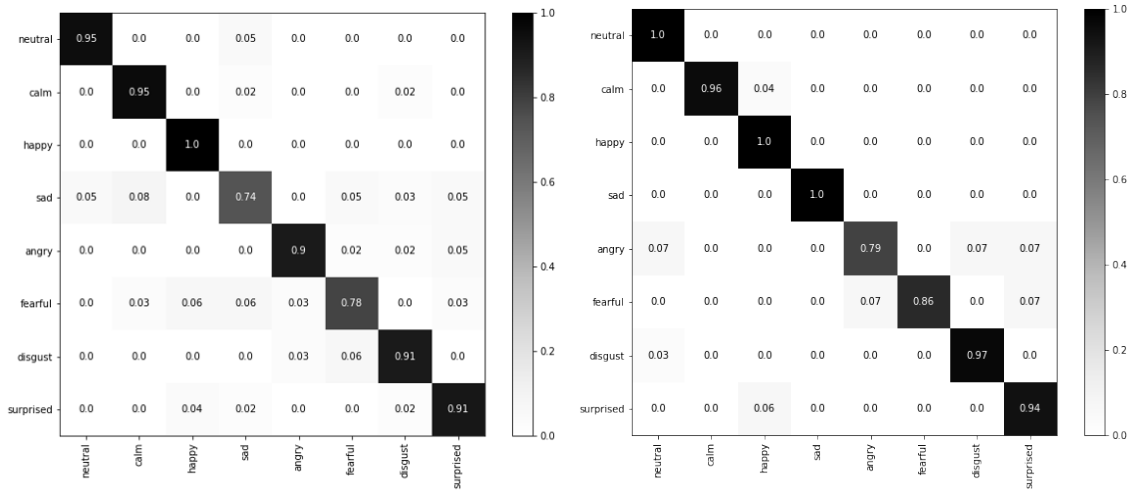
Figure 4.2. Audio subsystem results comparison

The final accuracy increases. From the confusion matrix shown in Figure 4.2b. The emotion "fearful" has the highest accuracy of 90%, "sad" has the lowest accuracy of 68%.

4.3 Feature Fusion System Result

The proposed feature fusion model is shown in Figure 3.5, confusion matrix are shown in Figures 4.3a and 4.3b. By using VGG16 attention GRU video subsystem, MFCC attention GRU audio subsystem, and concatenative feature fusion, our proposed method achieves 92% average accuracy, 94% best accuracy, 88% lowest accuracy, and 0.04 standard deviation of 10-fold cross-validation. The confusion matrices shown in Figure 4.3 demonstrate that in our previous model emotion "happy" achieves 100% accuracy, "sad" has the lowest accuracy of 74%. In the new model, emotion "neutral", "happy", and "sad" achieve 100% prediction accuracy, "angry" has the lowest accuracy of 79%. The results show that the feature fusion model significantly improves the accuracy compared with our subsystems. For the emotion classes in which our subsystem did not perform well, the feature fusion system achieved a better accuracy.

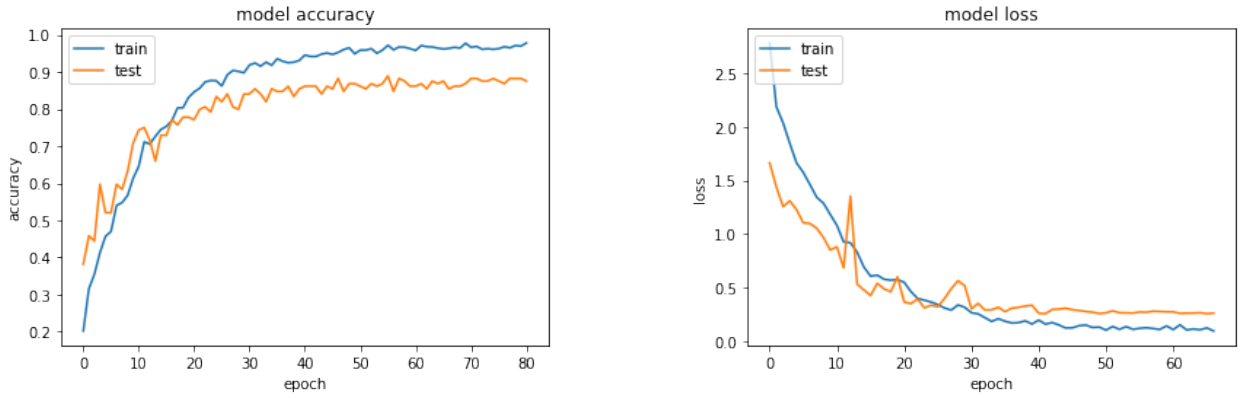
Table 4.6 summarizes our video subsystem, audio subsystem, and feature fusion system results compared with other's published methods, using human volunteers' accuracy as a



(a) Feature fusion system result confusion matrix

(b) Feature fusion system with attention result confusion matrix

Figure 4.3. Feature fusion system results comparison



(a) Training and test accuracy

(b) Training and test Loss

Figure 4.4. Feature Fusion Model Accuracy and Loss

baseline. Our video subsystem has the accuracy of 83% and including the human volunteers achieve 72%, while our audio subsystem has the accuracy of 81%, human volunteers only achieve 62%. Our video and audio feature fusion system obtain the highest accuracy of 91%. Our model outperforms the other methods using both audio and video data proposed by Ghaleb et al [38], Mustaqeem et al. [39], and Issa et al. [40]. Note that all the accuracy and confusion matrix results shown above are the best accuracy of 10 fold cross-validation.

Table 4.5. Feature Fusion model result(10-Fold CV)

Average Accuracy	Best Accuracy	Lowest Accuracy	STD
91.25%	94.44%	87.50%	0.02

Table 4.6. Comparison with other methods

Method	Accuracy
Multi-modal and temporal perception(Audio & Video)[38]	68%
Incorporating learned features and deep BiLSTM(Audio & Video)[39]	77%
Deep convolutional neural networks(Audio)[40]	72%
Human volunteers(Audio)[35]	62%
Human volunteers(Video)[35]	72%
Human volunteers(Audio and Video)[35]	80%
Ours(Audio)	81%
Ours(Video)	83%
Ours(Audio & Video)	92%

4.4 Future Work

In the future, we can test our system on more datasets especially data in real-world scenarios to validate the ability of generalization such as AFEW[41]. Furthermore, we will be in cooperation with our research results with psychology professionals. Besides, we can use vision transformer on the visual input, instead of applying face detection to video frames. Also, the attention mechanism could be used to replace RNN to process sequential inputs.

5. CONCLUSION

In this thesis, we have reviewed the machine learning concepts used in this thesis research.

We proposed a multi-modalities system using visual and audio data for emotion recognition.

We have developed a VGG16 transfer learning model to extract visual features and MFCC for speech audio features.

Then, we applied GRU with attention mechanism to process sequential inputs from both channels.

Moreover, we adopted concatenative feature fusion to fuse outputs from video and audio channels.

Finally, we evaluate our system and compare the performance with others' methods and discuss our future work to improve the proposed system.

REFERENCES

[1] W. Minker, J. Pittermann, A. Pittermann, P.-M. Strauß, and D. Bühler, "Challenges in speech-based human–computer interfaces," *International Journal of Speech Technology*, vol. 10, pp. 109–119, 2007.

[2] W. Zhang, D. Zhao, X. Chen, and Y. Zhang, "Deep learning based emotion recognition from chinese speech," May 2016, pp. 49–58, ISBN: 978-3-319-39600-2.

[3] S. Jain, P. Jha, and R. Suresh, "Design and implementation of an automatic speaker recognition system using neural and fuzzy logic in matlab," in *2013 INTERNATIONAL CONFERENCE ON SIGNAL PROCESSING AND COMMUNICATION (ICSC)*, 2013, pp. 319–324. DOI: 10.1109/ICSPCom.2013.6719805.

[4] A. Graves, A.-r. Mohamed, and G. Hinton, *Speech recognition with deep recurrent neural networks*, 2013. arXiv: 1303.5778 [cs.NE].

[5] T. Young, D. Hazarika, S. Poria, and E. Cambria, *Recent trends in deep learning based natural language processing*, 2018. arXiv: 1708.02709 [cs.CL].

[6] B. Yang, J. Cao, R. Ni, and Y. Zhang, "Facial expression recognition using weighted mixture deep neural network based on double-channel facial images," *IEEE Access*, vol. 6, pp. 4630–4640, 2018. DOI: 10.1109/ACCESS.2017.2784096.

[7] C. Shan, S. Gong, and P. W. McOwan, "Facial expression recognition based on local binary patterns: A comprehensive study," *Image and Vision Computing*, vol. 27, no. 6, pp. 803–816, 2009, ISSN: 0262-8856. DOI: https://doi.org/10.1016/j.imavis.2008.08.005. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0262885608001844.

[8] M. Ali, A. H. Mosa, F. Al Machot, and K. Kyamakya, "Eeg-based emotion recognition approach for e-healthcare applications," in *2016 Eighth International Conference on Ubiquitous and Future Networks (ICUFN)*, 2016, pp. 946–950. DOI: 10.1109/ICUFN.2016.7536936.

[9] U. Mangai, S. Samanta, S. Das, and P. Roy Chowdhury, "A survey of decision fusion and feature fusion strategies for pattern classification," *IETE Technical Review*, vol. 27, Jul. 2010. DOI: 10.4103/0256-4602.64604.

[10] S. Khalid, T. Khalil, and S. Nasreen, "A survey of feature selection and feature extraction techniques in machine learning," in *2014 Science and Information Conference*, 2014, pp. 372–378. DOI: 10.1109/SAI.2014.6918213.

[11] A. Galassi, M. Lippi, and P. Torroni, "Attention in natural language processing," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 10, pp. 4291–4308, Oct. 2021, ISSN: 2162-2388. DOI: 10.1109/tnnls.2020.3019893. [Online]. Available: http://dx.doi.org/10.1109/TNNLS.2020.3019893.

[12] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, *An image is worth 16x16 words: Transformers for image recognition at scale*, 2021. arXiv: 2010.11929 [cs.CV].

[13] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, and Q. He, *A comprehensive survey on transfer learning*, 2020. arXiv: 1911.02685 [cs.LG].

[14] K. Simonyan and A. Zisserman, *Very deep convolutional networks for large-scale image recognition*, 2015. arXiv: 1409.1556 [cs.CV].

[15] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds., vol. 25, Curran Associates, Inc., 2012. [Online]. Available: https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.

[16] A. Zhang, Z. C. Lipton, M. Li, and A. J. Smola, "Dive into deep learning," *arXiv preprint arXiv:2106.11342*, 2021.

[17] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[18] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, ISSN: 0899-7667. DOI: 10.1162/neco.1997.9.8.1735. eprint: https://direct.mit.edu/neco/article-pdf/9/8/1735/813796/neco.1997.9.8.1735.pdf. [Online]. Available: https://doi.org/10.1162/neco.1997.9.8.1735.

[19] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, *Learning phrase representations using rnn encoder-decoder for statistical machine translation*, 2014. arXiv: 1406.1078 [cs.CL].

[20] L. Muda, M. Begam, and I. Elamvazuthi, *Voice recognition algorithms using mel frequency cepstral coefficient (mfcc) and dynamic time warping (dtw) techniques*, 2010. arXiv: 1003.4083 [cs.MM].

[21] A. V. Oppenheim, R. W. Schafer, and J. R. Buck, *Discrete-Time Signal Processing (2nd Ed.)* USA: Prentice-Hall, Inc., 1999, ISBN: 0137549202.

[22] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, *Bert: Pre-training of deep bidirectional transformers for language understanding*, 2019. arXiv: 1810.04805 [cs.CL].

[23] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, *Attention is all you need*, 2017. arXiv: 1706.03762 [cs.CL].

[24] D. Bahdanau, K. Cho, and Y. Bengio, *Neural machine translation by jointly learning to align and translate*, 2016. arXiv: 1409.0473 [cs.CL].

[25] M.-T. Luong, H. Pham, and C. D. Manning, *Effective approaches to attention-based neural machine translation*, 2015. arXiv: 1508.04025 [cs.CL].

[26] S. Ioffe and C. Szegedy, *Batch normalization: Accelerating deep network training by reducing internal covariate shift*, 2015. arXiv: 1502.03167 [cs.LG].

[27] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, no. 56, pp. 1929–1958, 2014. [Online]. Available: http://jmlr.org/papers/v15/srivastava14a.html.

[28] S. Ruder, *An overview of gradient descent optimization algorithms*, 2017. arXiv: 1609.04747 [cs.LG].

[29] B. McMahan and M. Streeter, "Delay-tolerant algorithms for asynchronous distributed online learning," in *Advances in Neural Information Processing Systems*, Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger, Eds., vol. 27, Curran Associates, Inc., 2014. [Online]. Available: https://proceedings.neurips.cc/paper/2014/file/5cce8dede893813f879b873962fb669f-Paper.pdf.

[30] D. P. Kingma and J. Ba, *Adam: A method for stochastic optimization*, 2017. arXiv: 1412.6980 [cs.LG].

[31] J. D. Owens, M. Houston, D. Luebke, S. Green, J. E. Stone, and J. C. Phillips, "Gpu computing," *Proceedings of the IEEE*, vol. 96, no. 5, pp. 879–899, 2008. DOI: 10.1109/JPROC.2008.917757.

[32] *Cloud tensor processing units (tpus) google cloud*. [Online]. Available: https://cloud.google.com/tpu/docs/tpus.

[33] I. J. Goodfellow, D. Erhan, P. L. Carrier, A. Courville, M. Mirza, B. Hamner, W. Cukierski, Y. Tang, D. Thaler, D.-H. Lee, Y. Zhou, C. Ramaiah, F. Feng, R. Li, X. Wang, D. Athanasakis, J. Shawe-Taylor, M. Milakov, J. Park, R. Ionescu, M. Popescu, C. Grozea, J. Bergstra, J. Xie, L. Romaszko, B. Xu, Z. Chuang, and Y. Bengio, *Challenges in representation learning: A report on three machine learning contests*, 2013. arXiv: 1307.0414 [stat.ML].

[34] M. Lyons, M. Kamachi, and J. Gyoba, *The Japanese Female Facial Expression (JAFFE) Dataset*, The images are provided at no cost for non- commercial scientific research only. If you agree to the conditions listed below, you may request access to download., Zenodo, Apr. 1998. DOI: 10.5281/zenodo.3451524. [Online]. Available: https://doi.org/10.5281/zenodo.3451524.

[35] S. R. Livingstone and F. A. Russo, "The ryerson audio-visual database of emotional speech and song (ravdess): A dynamic, multimodal set of facial and vocal expressions in north american english," *PloS one*, vol. 13, no. 5, e0196391, 2018.

[36] Y. Song, Y. Cai, and L. Tan, "Video-audio emotion recognition based on feature fusion deep learning method," in *2021 IEEE International Midwest Symposium on Circuits and Systems (MWSCAS)*, 2021, pp. 611–616. DOI: 10.1109/MWSCAS47672.2021.9531812.

[37] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255. DOI: 10.1109/CVPR.2009.5206848.

[38] E. Ghaleb, M. Popa, and S. Asteriadis, "Multimodal and temporal perception of audio-visual cues for emotion recognition," in *2019 8th International Conference on Affective Computing and Intelligent Interaction (ACII)*, 2019, pp. 552–558. DOI: 10.1109/ACII.2019.8925444.

[39] Mustaqeem, M. Sajjad, and S. Kwon, "Clustering-based speech emotion recognition by incorporating learned features and deep bilstm," *IEEE Access*, vol. 8, pp. 79 861–79 875, 2020. DOI: 10.1109/ACCESS.2020.2990405.

[40] D. Issa, M. Fatih Demirci, and A. Yazici, "Speech emotion recognition with deep convolutional neural networks," *Biomedical Signal Processing and Control*, vol. 59, p. 101 894, 2020, ISSN: 1746-8094. DOI: https://doi.org/10.1016/j.bspc.2020.101894. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1746809420300501.

[41] A. Dhall, R. Goecke, S. Lucey, and T. Gedeon, "Static facial expression analysis in tough conditions: Data, evaluation protocol and benchmark," in *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, 2011, pp. 2106–2112. DOI: 10.1109/ICCVW.2011.6130508.

PUBLICATION

Y. Song, Y. Cai and L. Tan, "Video-Audio Emotion Recognition Based on Feature Fusion Deep Learning Method," 2021 IEEE International Midwest Symposium on Circuits and Systems (MWSCAS), 2021, pp. 611-616, doi: 10.1109/MWSCAS47672.2021.9531812.