WARSAW UNIVERSITY OF TECHNOLOGY

FACULTY OF ELECTRONICS
AND INFORMATION TECHNOLOGY

Ph.D. Thesis

Xin Chang, M.Sc.

Human Emotion Recognition from Image and Speech using Deep Neural Networks

Supervisor
Professor Władysław Skarbek, Ph.D., D.Sc.

WARSAW 2021

Acknowledgements

I would like to express my special gratitude to my supervisor prof. dr hab. inż. Władysław Skarbek, who guided me with patient, inspiration, and opened the door of academic research.

I would also like to thank my family who supported me in all their way so that I could focus on my research without any worry in my mind.

Finally, I thank to acadamic colleges in Warsaw University of Technology who helped me through all the obstacles for the five years of study in Poland.

Human Emotion Recognition from Image and Speech using Deep Neural Networks

Streszczenie. Rozpoznawanie emocji to ważny obszar badawczy dotyczący interakcji człowiek komputer. Pomimo, że komputerowa analiza danych sensorycznych takich jak obraz twarzy i głos, osiąga spektakularne wyniki, w wielu przypadkach lepsze od wyników osiąganych przez ludzi, to automatyczna dwu-modalna analiza emocji na podstawie obrazu i dźwięku jednocześnie, tak jak to faktycznie realizuje mózg człowieka, daleka jest jeszcze od mierzalnego poziomu ludzkich możliwości. Niniejsza rozprawa doktorska jest próbą zbliżenia się do tej granicy. Przedstawione wyniki dotyczą czterech typowych scenariuszy badawczych stosowanych klasyfikacji emocji, dokonywanej na podstawie danych ekstrahowanych z: (a) pojedynczego zdjęcia twarzy, (b) nagrania wideo, tj. z temporalnej sekwencji obrazów, (c) nagrania audio, tj. nagrania mowy, (d) klipu filmowego, tj. zsynchronizowanego nagrania wideo i audio. W systemach rozpoznawania wyróżnia się komponenty służące wydobywaniu cech i komponenty klasyfikujące te cechy. W scenariuszu (a) w niniejszej pracy pokazano wyższość rozwiązania neuronowego nad klasycznym już podejściem, w którym cechy geometryczne i animacyjne modelu Candide-3, uzyskuje się na podstawie detekcji punktów szczególnych modelu FP68, a następnie klasyfikuje w modelu tzw. maszyny wektorów nośnych (SVM). Właściwa strategia uczenia się cech głębokich przez inne zadania związane z twarzami, tj. technika transferu modelu neuronowego sprawiła, że proponowany model jest skuteczny nawet przy stosunkowo ograniczonych zasobach zdjęć w zbiorze uczącym. W scenariuszu (b) zauważono, że temporalne urozmaicanie danych uczących znacząco poprawia skuteczność klasyfikatora emocji na podstawie sekwencji obrazu. Z kolei analiza sygnału mowy w scenariuszach (c) i (d) prowadzona jest na podstawie jego spektrogramu. Scenariusz (d), a więc dwu-modalna analiza emocji, z możliwym jej rozszerzeniem na przypadek wielo-modalny, jako najbardziej zbliżona do zachowań człowieka, zajmuje w pracy prominentne miejsce. Wykorzystując komponenty opracowane w realizacji scenariuszów (b) i (c), skupiono się na zagadnieniu fuzji rozwiązań jedno-modalnych. Zaproponowana architektura MRPN (Multimodal Residual Perceptron Network) eliminuje niedoskonałości rozwiązań stosujących tzw. późną fuzję i prowadzi do aktualnie najlepszych wyników osiąganych w klasyfikatorach emocji łączących dane wideo i audio, tj. na następujących, powszechnie stosowanych zbiorach danych testowych: RAVDESS, Crema-d, FER2013, RaFD, MUG oraz CK+.

Słowa kluczowe: rozpoznawanie emocji twarzy, rozpoznawanie mowy, rozpoznawanie emocji audio-wideo, multimodalna sieć neuronowa, głęboka fuzja funkcji

Human Emotion Recognition from Image and Speech using Deep Neural Networks

Summary. Recognizing emotions is an important research area of human-computer interaction. Although computer analysis of sensory data such as face image and voice achieves spectacular results, in many cases better than human results, automatic bi-modal analysis of emotions based on image and sound simultaneously, as is actually done by the human brain, is still far from a measurable level of human capacity. This doctoral dissertation is an attempt to get closer to this border. The presented results concern four typical research scenarios for the applied classification of emotions, made on the basis of data extracted from: (a) a single photo of a face, (b) a video recording, i.e. from a temporal sequence of images, (c) audio recordings, i.e. speech recordings, (d) a movie clip, i.e. synchronized video and audio recording. Recognition systems distinguish components for extracting features and components that classify these features. In scenario (a) in this paper, the superiority of the neural solution over the classic approach, in which the geometric and animation features of the Candide-3 model are obtained on the basis of the detection of special points of the FP68 model, and then classified in the model, the so-called support vector machines (SVMs). The proper strategy of learning deep features through other face-related tasks, i.e. the neural model transfer technique, made the proposed model effective even with relatively limited resources of images in the training set. In scenario (b) it was noticed that temporal augmentation of the training data significantly improves the effectiveness of the emotion classifier based on the image sequence. In turn, the analysis of the speech signal in scenarios (c) and (d) is carried out on the basis of its spectrogram. Scenario (d), i.e. the two-modal analysis of emotions, with a possible extension to the multi-modal case, as being the closest to human behavior, occupies a prominent place at work. Using the components developed in the implementation of scenarios (b) and (c), the focus was on the issue of fusion of one-modal solutions. The proposed MRPN (Multimodal Residual Perceptron Network) architecture eliminates the imperfections of solutions using the so-called late fusion and leads to the currently best results in emotion classifiers combining video and audio data, i.e. on the following commonly used test data sets: RAVDESS, Crema-d, FER2013, RaFD, MUG and CK+.

Keywords: facial emotion recognition, speech emotion recognition, audio-video emotion recognition, multi-modal neural network, deep feature fusion

Contents

Acknowledgements . . . 3
1. Introduction . . . 9
1.1. Emotion Recognition from Face Expression and Voice Timbre . . . 10
1.2. Multi-modal emotion recognition . . . 11
1.3. Hypothesis of the doctoral thesis and contributions . . . 12
2. Artifical Neural Network . . . 13
2.1. Feedforward of neural network . . . 13
2.2. Backpropagation of neural network . . . 14
2.3. Model optimization . . . 16
2.3.1. Data scaling . . . 16
2.3.2. Activation function . . . 17
2.3.3. Weights Normalization . . . 18
2.4. Model generalization . . . 18
2.4.1. Underfitting, overfitting and misfitting . . . 19
2.4.2. Data augmentation . . . 20
2.5. Convolution Neural Network . . . 21
2.5.1. Artifical convolution kernels . . . 21
2.5.2. Classical Convolution Architecture . . . 22
2.6. Recurrent Neural Network . . . 23
2.6.1. Naive Recurrent Neural Network . . . 23
2.6.2. Long Short-term Memory . . . 24
2.7. Transformer . . . 25
3. Literature Review . . . 27
3.1. Face detection . . . 27
3.1.1. Traditional methods for Face detection . . . 27
3.1.2. Neural approach for Face detection . . . 29
3.2. Facial Emotion Recognition . . . 30
3.2.1. PCA and LDA for Facial Emotion Recognition . . . 30
3.2.2. 3D Modeling for Emotion Recognition . . . 31
3.3. Transfer learning . . . 32
3.4. Emotion recognition from streaming video . . . 33
3.5. Emotion recognition from streaming audio . . . 34
3.6. Multi-modal solution for Audio-Video Emotion recognition . . . 34
4. Proposed methods . . . 36
4.1. Facial Emotion Recognition via 3D modeling . . . 36
4.2. Traditional solution versus neural solution for facial emotion recognition . . . 43
4.2.1. Neural solution replacing SVM classifier . . . 43
4.2.2. End-to-end Neural facial emotion classification framework . . . 44

4.3. Transfer learning from facial emotion recognition . . . 47
4.3.1. Source task: VGG face descriptor . . . 47
4.3.2. Target task: Emotion Recognition . . . 49
4.3.3. Experimental Dataset . . . 49
4.3.4. Data preparation . . . 50
4.3.5. Training strategy . . . 51
4.3.6. Impact of learning rate . . . 52
4.3.7. Impact of fine-tuning strategies . . . 52
4.3.8. Comparable results with others . . . 53
4.4. Emotion recognition from streaming video in neural approach . . . 54
4.4.1. Model specification . . . 55
4.4.2. Experimental results . . . 56
4.5. Proposed methods of SER . . . 57
4.5.1. Model specification . . . 57
4.5.2. Preprocessing of audio data . . . 58
5. Multi-modal Residual Perceptron Network for AVER . . . 60
5.1. Hypothesis . . . 60
5.1.1. Potential failures in the existing solutions . . . 60
5.1.2. Within-modal information can be missing or fuzzy . . . 61
5.1.3. End-to-end modeling for multi-modal data can be distorted . . . 61
5.1.4. Late fusion modeling for multi-modal data can be insufficient . . . 63
5.2. Proposed methods . . . 63
5.2.1. Functional description of analyzed networks . . . 63
5.2.2. MRPN components' role in multi-term optimization . . . 66
5.2.3. MRPN in general multi-modal applications . . . 67
5.3. Computational Experiments and their Discussion . . . 67
5.3.1. Datasets . . . 67
5.3.2. Model organization and computational setup . . . 69
5.3.3. Data augmentation cannot generalize multi-modal feature patterns . . . 69
5.3.4. Discussion on inferior multi-modal cases . . . 69
5.3.5. Improvement of MRPN . . . 70
5.3.6. Comparing baseline with SOTA . . . 71
6. Conclusion . . . 73
List of Symbols and Abbreviations . . . 74
List of Figures . . . 76
List of Tables . . . 79
References . . . 80
List of Xin Chang publications . . . 87

1. Introduction

The objective of this paper is to demonstrate the creation of an unique end-to-end Deep Neural Network (DNN) framework for solving the Audio-Video Emotion Recognition (AVER) problem.

Long before DNN became practical, emotion recognition (ER) was researched. P. Ekman[1] began investigating human emotions in 1965 and suggested six fundamental emotions; he asserts that these six emotion classes are ubiquitous across cultures, nationalities, and sexual orientations. P. Ekman and WV. Friesen[2] create the Facial Action Coding System (FACS) to define actions from muscle groups based on this concept.

Visual information is not the only source for understanding human emotions; the human voice, texture meaning from language, posture, and even Electroencephalography (EEG) signals have all come into the researcher's view in order to gather and analyze human beings' emotional states.

The investigation of alternative modes of emotional expression, makes Human Computer Interaction (HCI) much more realistic in a variety of applications. By using speech analysis, software may assist physicians in diagnosing illnesses such as depression and dementia. Gartner client enquiries indicate that demand for employee safety solutions is increasing. Emotion AI can aid in the analysis of workers who do physically demanding professions, such as first responders. Automotive manufacturers can monitor the driver's emotional state using computer vision technologies. A driver's alert system may be triggered by an intense emotional condition or sleepiness. Insurance firms utilize speech analysis to determine whether or not a consumer is being truthful while filing a claim. Independent studies indicate that up to 30% of users confess to lying to their auto insurance provider in order to get coverage. An irate client may be identified early on and sent to a well-trained person who can also monitor and modify the discussion in real time.

The conventional approach to emotion detection in computer vision is often divided into two parts: feature extraction and feature categorization. The goal of feature extraction is to extract descriptive characteristics that are numerically significant from the raw data source. Certain classifiers make use of these characteristics to categorize emotional experiences.

The recent invention of Artificial Neural Networks (ANN) has transformed the landscape since then. As a tool for bioengineering signal processing, ANN demonstrates its ability to handle any kind of signal using a universal function composed of millions of basic unit neuron functions. Granted by the GPU's strong parallel computing, the ANN has shown state-of-the-art (SOTA) achievements in a wide variety of study fields. Additionally, ANN solutions may accept extremely raw data and generate the desired outputs directly, transforming many applications' solutions into end-to-end solutions.

1.1. Emotion Recognition from Face Expression and Voice Timbre

In the Facial Emotion Recognition (FER) system, information is collected by the camera and then spread over many frames, as shown in Figure 1.1. A single frame's discrete information is initially supplied to pattern extracting units for feature extraction. Traditionally, Fisherfaces and Eigenfaces [3] were extracted using Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA), however they have been superseded by deep features retrieved using Convolution Neural Networks (CNN) [4]. To completely retain the information contained in the discrete signals, some kind of Sequence Aggregation Component (SAC), such as a Long Short-term Memory (LSTM) or Transformer [5], is then required to process the retrieved features in the neural method. Traditionally, Markov chains have been employed to analyze temporal data [6]. Finally, a classifier such as a Support Vector Machine (SVM) or a neural dense layer classifies the combined characteristics.

Figure 1.1. Video frames of visual facial expressions selected from RAVDESS (Ryerson Audio-Visual Database of Emotional Speech and Song) dataset.

The raw voice inputs for Speech Emotion Recognition (SER) are typically between 10,000 to 44,100 samples per second, whereas the visual frame rate is about 25-30 picture frames per second. Acoustic characteristics such as Mel-frequency cepstral coefficients (CFCCs), linear prediction cepstral coefficients (LPCCs), and fundamental frequencies (F0) have long been used as descriptors of speech characteristics. While raw digital signals in the temporal domain closely resemble the original signal, their spectral representations, such as the Spectrogram frame, Mel-spectrogram coefficients, or Log Mel-spectrogram frame, proved to be more successful for sound identification. Despite certain restrictions, spectral transformed voice signals demonstrated substantial gains in a variety of classification tasks. Because the time

of expression events fluctuates, so does the breadth of the Spectrogram frames, which is inconvenient for CNN pattern extractors. As a result, the extracted features need further processing by SAC, which produces integrated features. The Figure 1.2 illustrates expression events belonging to various types and with varying durations.

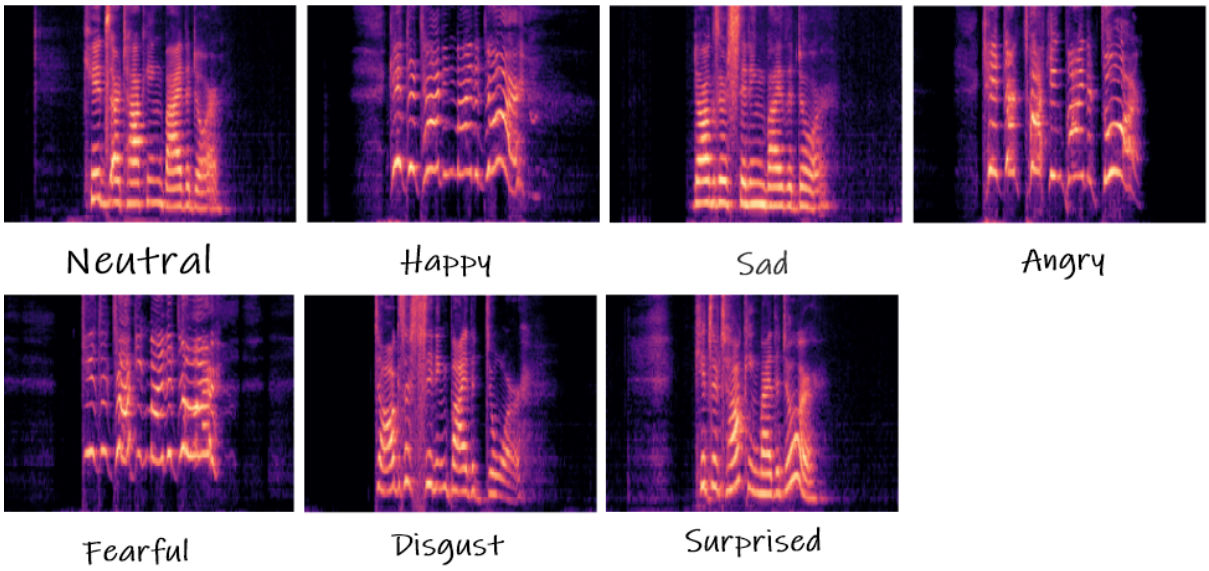


Figure 1.2. Mel Spectrograms of vocal timbres selected from RAVDESS (Ryerson Audio-Visual Database of Emotional Speech and Song) dataset.

1.2. Multi-modal emotion recognition

Human beings rely on a variety of senses, or modes, such as the auditory, visual, and textual, to interpret emotional expressions in our everyday social interactions. For multi-modal emotion identification systems, inputs from multiple modes should be analyzed and integrated. Intelligent artificial detectors including neural processing units are playing key roles in activities that relate to emotion detection. HCI takes use of its advanced sensors, which collect information for comprehending multi-modal information the same way human beings are able to process emotional emotions.

The AVER solution is likewise based on human perception. Many individuals may "hear" a sound while gazing at sheet music, "smell" a scent when remembering a memory, or "see" the ocean from the scent of the air. Our cerebral cortex evaluates information through processing movement, hearing, and sight, among other senses. Certain additional brain areas are intimately tied to this knowledge. As a result, the decision is based not only on identifying unique sensations, but also on taking them all into account.

The neural sensor learning mechanism should be similar to how people learn. The neurons train in the same way as the human brain cortex does, by adjusting to their environment and monitoring their actions throughout the supervised neural network sensor training process.

1.3. Hypothesis of the doctoral thesis and contributions

The purpose of this thesis is to increase the rate of emotion recognition in the FER, SER, and AVER dimensions. In the FER, we hypothesize that using neural networks in lieu of conventional ER systems may address the problem of poor system adaptability to new data and environmental conditions.

The pervasiveness of generalization Capacity of the neural network requires a huge quantity of data, however human-related activities struggle to retrieve significant amounts of data owing to security concerns. We seek to improve the recognition rate for FER and SER by using transfer learning of the neural network's general knowledge.

AVER, as multi-modal solutions to the ER problem, has been shown to be superior to alternative uni-modal methods. However, based on our tests, we observed that the AVER system can have a lower recognition rate than uni-modal solutions in some cases. We hypothesize that this is due to a weakness in the current multi-modal ER system and are working to develop a universal solution that avoids such situations.

Addressing the aforementioned study directions, we have several contributions to the ER problems answer the following hypotheses:

1. Basic neural network solutions for emotion recognition can be superior than traditional even complex machine learning solutions like Candide model/SVM.
2. Transfer learning technique applied for the initialization of deep features extraction stage, in case of emotion recognition systems while reducing model training costs can provide comparable recognition performance.
3. Developing relevant time augmentation techniques for AV data used for learning multi modal emotion recognition systems (AVER) can improve recognition model performance with marginal time complexity overhead.
4. While the late fusion of uni-modal speech based (SER) nad video based (VER) emotion recognition neural systems can give the inferior results for some counterexample AV data, its replacing by the MRPN (Multimodal Residual Peceptron Network) component results in consistent performance improvement for all those counterexample AV data.

The paper is structured as follows. The second part goes into depth on the overall ANN system, training, and optimization, with emphasis on the process of ANN development and the foundations of the final ER system. Following this, a discussion of existing literature on the FER, ER, and AVER ER solutions will appear. We'll detail the conventional and neurological remedies to each issue we identify, while also detailing the pros and cons of each.

Our experimental findings are presented for each aspect of the ER system, illustrating the benefits of our suggested solutions. As our last AVER solution, Multimodal Residual Perceptron Network (MRPN) is detailed in its own section. At last, we conclude our work for the ER task and propose future possibilities, with all we've got.

2. Artifical Neural Network

A neural network (NN), sometimes called an artificial neural network, is a computer system modeled after the human brain that is capable of intelligent information processing and generalization. It was initially created to address AI issues. A breakthrough in the field of Biological Engineering was made, inspired by the human brain, which has hundreds of billions of nerve cells, or neurons, connected together to form neural networks.

2.1. Feedforward of neural network

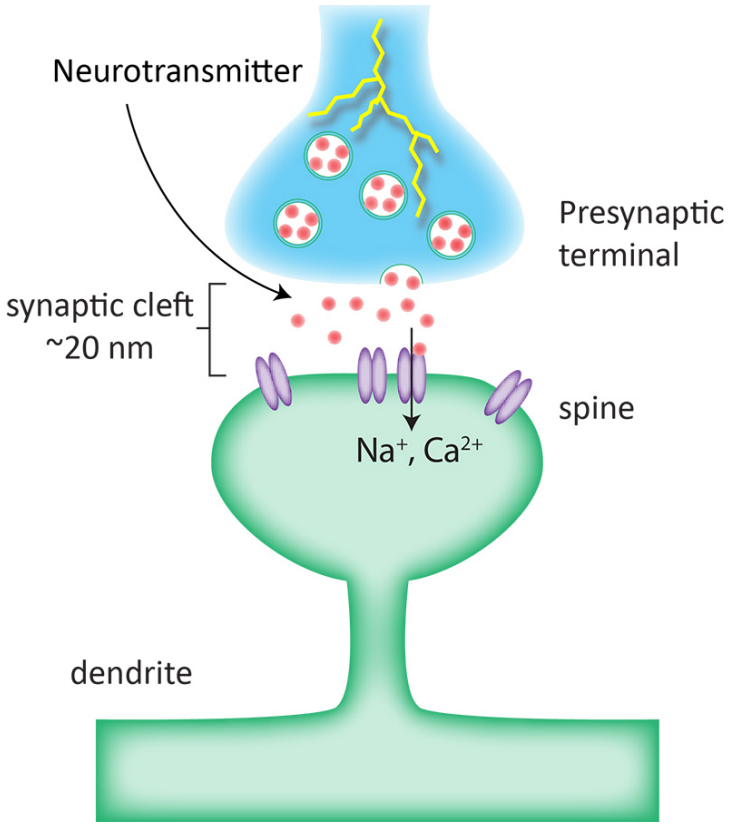


Figure 2.1. Neuron element diagram from a biological perspective. (Image source: Alan Woodruff / QBI)

Figure 2.1 depicts how one neuron links to many others and has the ability to make new connections or change existing ones. Neurons are cells that communicate with one another via neurotransmitters. Synapses, also known as "synaptic clefts," link neurons. When a neuron fires, it sends nerve impulses, releases neurotransmitters, and perhaps communicates this information to neighboring neurons through neurotransmitters across the synaptic cleft. The intensity of the neurotransmitter has a direct impact on the strength of the signals.

The ANN does brain cell art. The interconnected neurons of the ANN create a network made up of many layers, each of which has numerical weights to store information. The cap is placed on neurons so that they are prepared to fire when needed, much like synaptic clefts.

In neurons, information is represented as a bias. In a mathematical explanation, the ANN is just a simple cascade summation of:

$$A = \sigma(\sum_i \omega_i x_i + b_i) \tag{1}$$

In Equation 1, the ω_i stands for the weight number in every neuron, x_i represent the information from the last neuron, b indicate the bias in the current neuron and A means the next neuron which receives the information or the final information of certain sense.

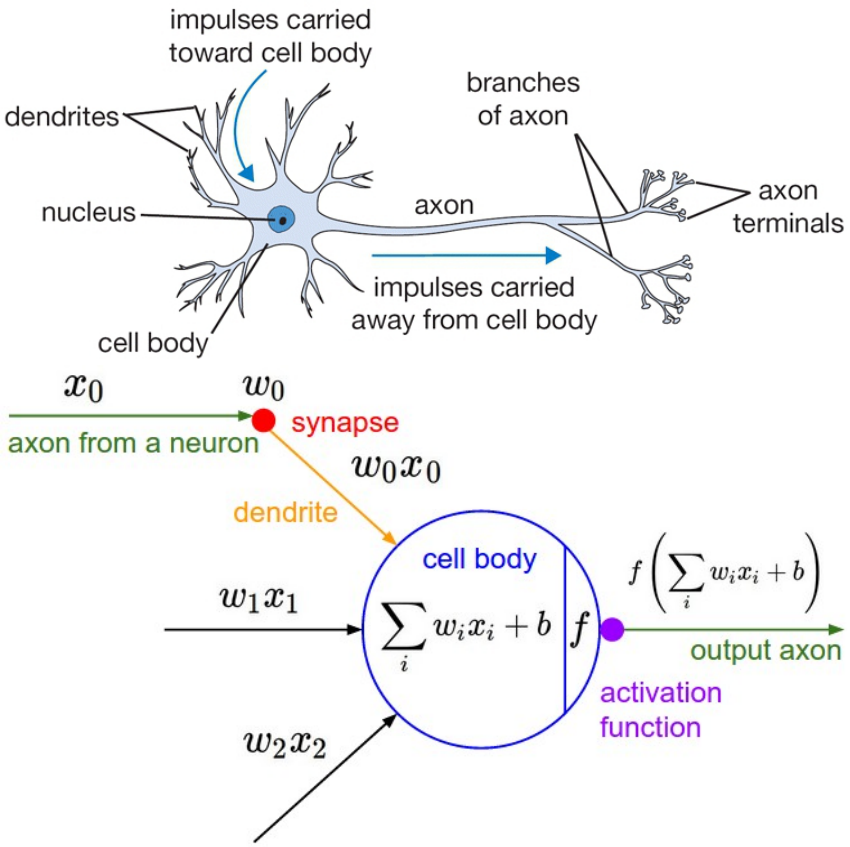


Figure 2.2. Artificial neural element versus biological neural element (Image source: `https://cs231n.github.io/neural-networks-1/`)

The ANN is built up from the artificial neurons as Figure 2.2 illustrates, the general defined mapping Function 1 is possible to mimic the chemical information stored and passed among the brain neurons. The procedure of the function is called forward propagation, where the structure of the processing network is established. However at this point, the ANN only has the ability to process the information, the functionality, or the role of the neurons are not specified yet.

2.2. Backpropagation of neural network

Neural networks are known as a black box system (see Figure 2.3). While we know that the output of the neural system is a description of the input we provide, we are unclear about the

specific mapping operations used. The design of such a system ignores everything but the output and the stimulus inputs; the underlying working of the system is a complete mystery.

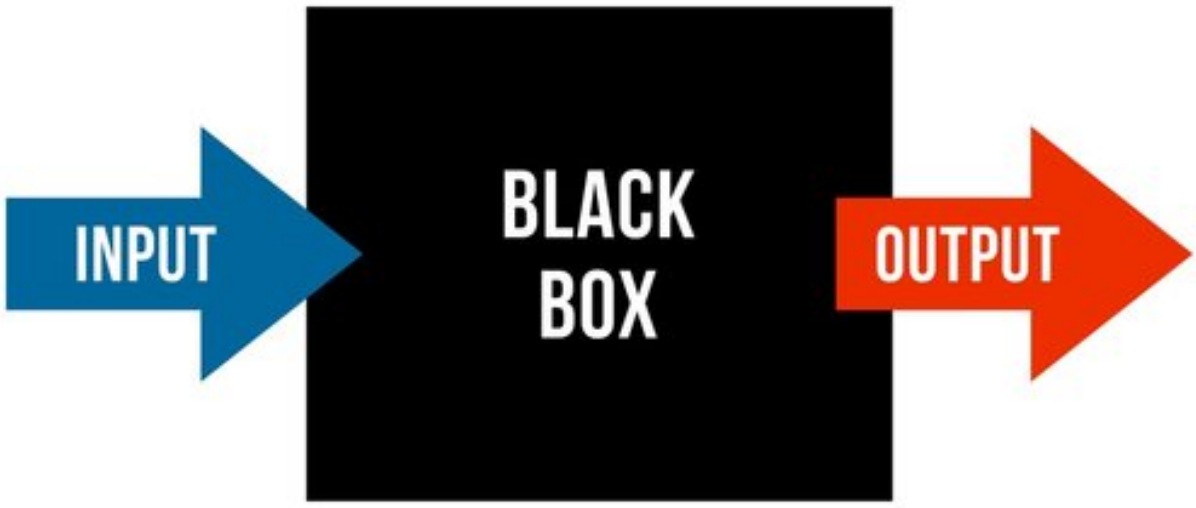


Figure 2.3. Block box system as system modeling concept.

The training of the neural network system, to respond properly according to the inputs, is based on the auto gradient backpropagation (BP) mechanism. BP mechanism is also general and simple. By computing the gradient of the loss function wrt the weights in the neural network system in a chain rule, the weights in each neuron of the system are updated accordingly through each stimulus.

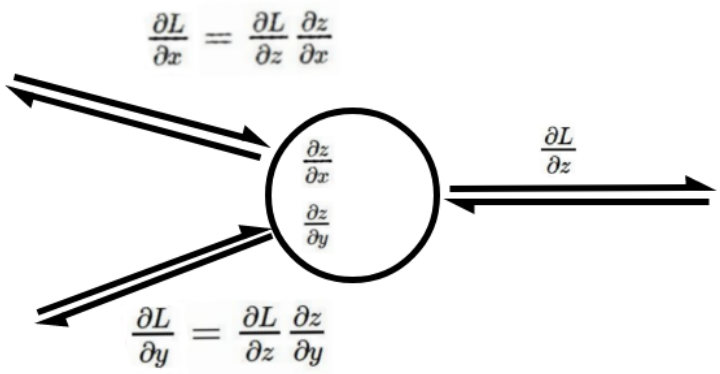


Figure 2.4. The general idea of error gradient backpropagation through the neural component.

In the illustration provided by Fig. 2.4, the gradient flow in a single neuron and the nested neural system is represented. The premise that the layered neural system is consistent with the flow is important to why the system may respond appropriately to inputs, as seen in the diagram.

When compared to other traditional systems, BP's learning is progressive rather than all at once. All at once refers to traditional machine learning algorithms such as PCA, SVM,

LDA, etc., where the algorithms learn the mapping functions from the analyzed database. BP, however, updates the system neuron weights step by step according to each sample in the database.

As a result, ANN requires the training database to be balanced in numbers and delivered in a shuffled order, otherwise, the weights of the network are updated towards the direction of the nature of the majority samples, while the nature of the small number samples is not understood well. The traditional machine learning mechanism does not have problems in such cases.

ANN need a balanced training database, supplied in a shuffled sequence; otherwise, the weights of the network tend to shift toward the bulk of the sample distribution, and so are poorly equipped to learn from outliers. In situations like what's in Figure **??**, the neural machine learning process won't work well.

Neural networks, despite the requirement for balance in the database, nevertheless have the benefit of lessening the impact of anomalous data on the training. A tiny number of bad data will only increase the neural network's overall inaccuracy by a little amount, which will not significantly impact the training procedure. Though conventional models include data collected from the whole database, they may misjudge some samples. This behavior helps to save time on assessing data in the preparation phase by reducing the amount of time needed to go through a lot of data.

2.3. Model optimization

Gradients are important in training for improved results. Model optimization considers them throughout the training process. For this issue, data, activation function, and network design, many variables are at play. Everything in the most fundamental neural function $A = \sigma(\sum i\omega ixi + bi)$, stems from these variables.

2.3.1. Data scaling

Because of the weighting values being numerical, the data scale is crucial. In order to maintain smaller values for the gradients, they must be normalized. Otherwise, when high weight values arise from large input values, the weight values may be widely changing and the loss will never diminish. To prevent very varying values, such as 0.01, 0.1, and 100, The data need standardization. There are many methods to deal with data kinds and sizes.

1. Min-max scaling: Min-max scaling will scale all data values in to the range $[0,1]$, which is exactly what we wanted to restrict the range of the values.

$$X' = \frac{X - X_{\min}}{X_{\max} - X_{\min}} \tag{2}$$

2. Value range specified: For some certain raw data types, such as an image, the pixel values are originally ranging from $[0,255]$, thus the values can be normalized by their maximum value.

3. Statistically specified: In the neural method to learning RGB pictures, the mean and standard deviation are statistically relevant. The mean and standard deviation are estimated as [0.485, 0.456, 0.406] and [0.229, 0.224, 0.255] correspondingly, based on millions of pictures.

$$x' = \frac{x - \mu}{\sigma} \tag{3}$$

2.3.2. Activation function

There is another kind of control that also remaps the weights' value range: the activation function. Some nonlinear activation functions, such as tanh and sigmoid, are used early on. In addition to data scaling, the activation functions standardized the feature values to be within a normal range, allowing for a smoother gradient optimization.

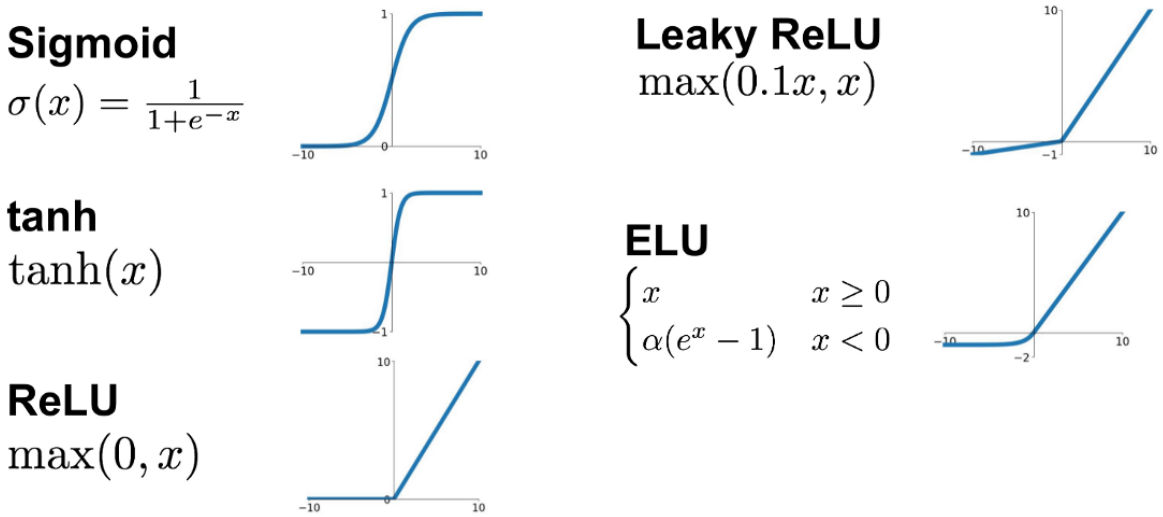


Figure 2.5. Examples of commonly used activation functions in neural networks.

However, researchers have realized that although the nonlinear activation functions can remap the values to the range of $[-1, 1]$ or $[0, 1]$, their gradient, as a component of the gradient for the neuron weight, can vanish according to their activation values. As Figure 2.6 shows, when the values of the weights are at a certain range, the gradient of the activation will be so closed to zero, thus, following the chain rule, the whole network will not be learning anything from the specific samples.

Although the nonlinear activation functions may transfer values in the range of [-1,1] or [0,1], researchers have shown that, when mapping values, their gradient can vanish. The main problem is shown in 2.6, where when the weights are in a particular range, because the gradient of the activation is almost zero, learning ceases and the network is unable to generalize to new circumstances.

To address this issue, the Rectified Linear Unit (ReLU) activation function was used. The ReLU activation function forces the gradient of value one when it has input values that are greater than zero, as illustrated in Figure 2.5 and Figure 2.6. When it sees a negative value, the

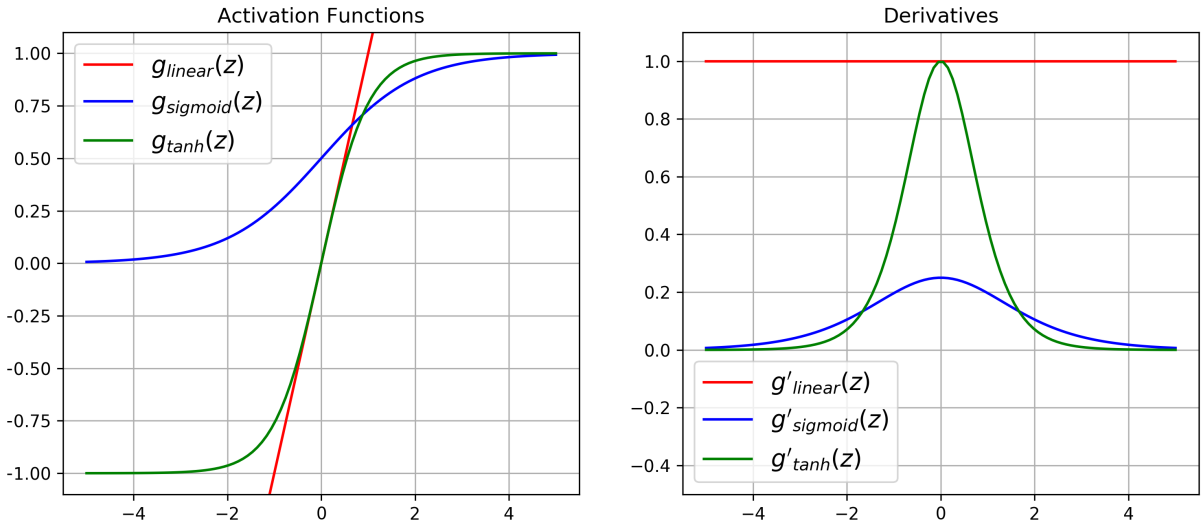


Figure 2.6. Examples of gradients for commonly used activation functions.

activation function zeros out the whole neural network chain. This causes the neural network to gain no information.

In spite of its lack of resilience to zero gradients for some cases, the ReLU activation function is more efficient than nonlinear activation functions for use in neural network optimization.

2.3.3. Weights Normalization

Weights optimization is necessary to make sure that the value of the neuron weights is consistent. The ReLU family of activation functions solved some of the gradient vanishing issue, but they also had the disadvantage of not remapping feature values in the intermediate neural layers, which leaves the potential for larger feature weights to be learned.

There are several techniques performing data normalization, based on the mean and standard deviation of the input data with the selected axis, namely Batch Norm [7], Layer Norm [8], Instance Norm [9], and Group Norm [10]. This procedure aims to recover the global statistics of the input database. Since the gradient also considers the value of the old weights, thus by normalizing the old weights in the inputs, the gradient can be handled more stable for the loss to converge.

2.4. Model generalization

Model generalization is another part of the optimization problem. The neural approach to the empirical tasks is powerful due to its generalized behavior to the unseen new data. Benefited from the large set of functions described by the neural network, the extracted deep features describe patterns more precisely and generally than other traditional pattern features. Because of the vast number of iterations required by the functions and the coefficients that must be tuned inside the functions, the objective of neural network model generalization necessitates immense processing power and a significant amount of data.

2.4.1. Underfitting, overfitting and misfitting

Approximating the target function using a neural network model is achieved via the training process. The neural network mapping functions have coefficients that may be adjusted using the gradient information that's applied back via the layered functions. When it comes to generalization, there has to be an equal amount of data and parameters.

Intuitively, having a greater number of estimating parameters enables the expression of more complex patterns. Thus, the depth of the network system can be increased by stacking layers, and the breadth may be increased by increasing the number of neurons in the neuron layers. The performance of the network does not improve merely by increasing its depth. They must be given similar amounts of data in order to adjust the settings throughout the training stages. If the training samples are statistically insufficient in comparison to the

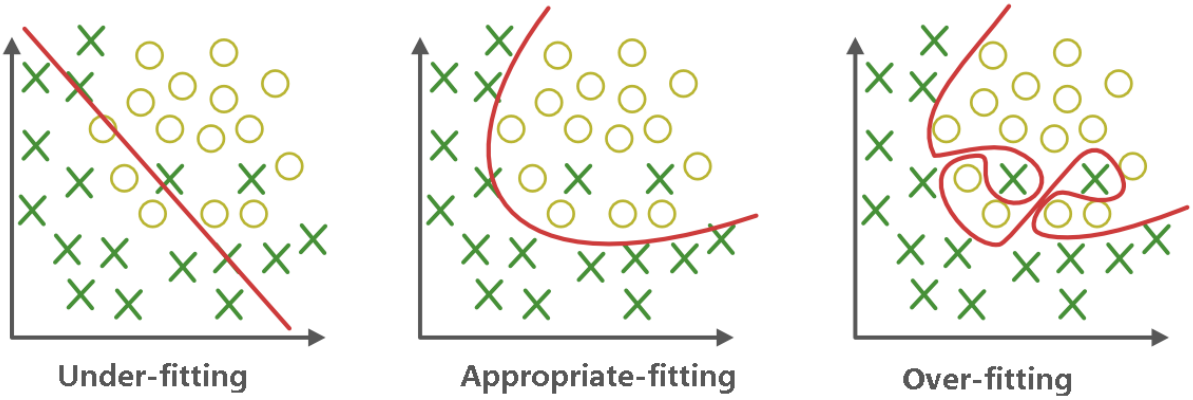


Figure 2.7. Graphical illustration of model fitting to data. (Image source: `geeksforgeeks`

network complexity, the network will be prone to overfitting due to the limited quantity of data that is statistically dissimilar to the general characteristic of the information we wish to capture. Because the patterns learnt on the training data are highly skewed, we may anticipate excellent performance on previously observed training data and poor performance on previously unknown testing data. On the other hand, if the network is too simplistic, it

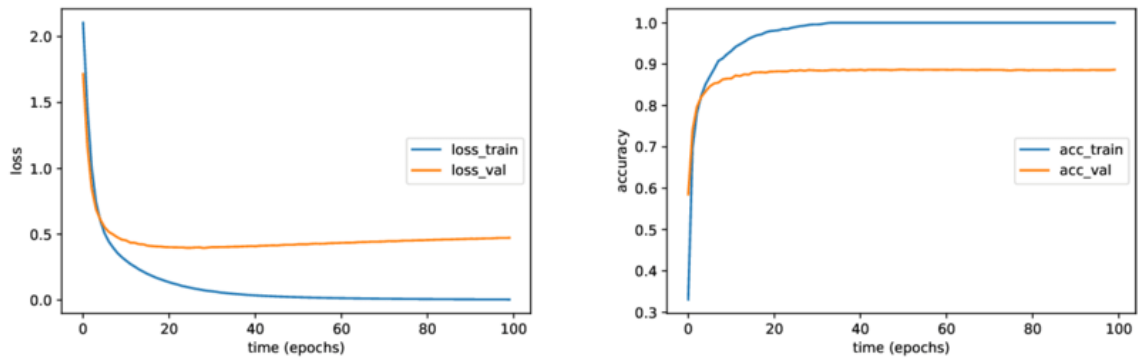


Figure 2.8. Typical model over-fitting observed for errors while the model is being trained.

lacks sufficient parameters to approximate the target function from which the generalized pattern must be extracted. In such a situation, the network will perform badly regardless of whether it is exposed to previously learnt material or previously unknown new data.

Figure 2.9. Typical model under-fitting observed for errors while the model is being trained.

Misfitting occurs when the training set does not include enough patterns to be sufficiently generalized; as a result, the model may function on certain testing samples but not on others. As shown by the recorded training curves, the validation curve may converge on one database while fluctuating or even diverging on another. This behavior is comparable to that of a model with an excessively high learning rate, however lowering the learning rate to a low number will not resolve the problem.

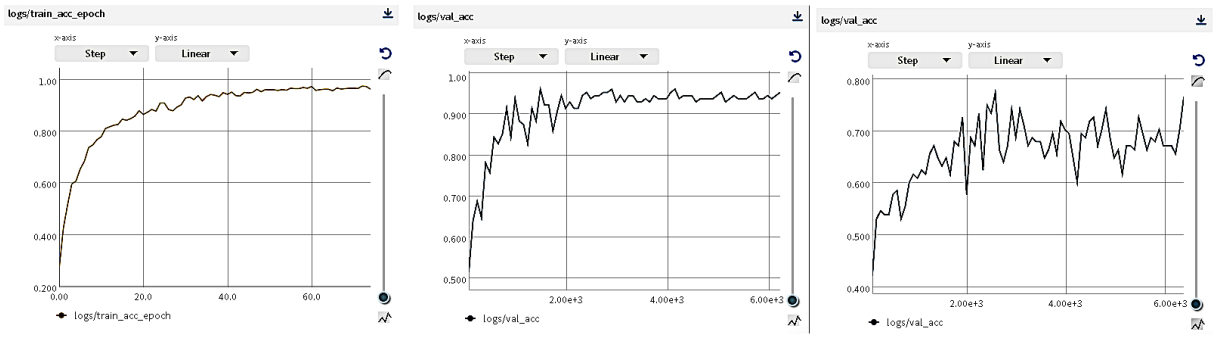


Figure 2.10. Misfitting in the training. left: training curve of the same training data.middle: validation curve on one validation dataset. right: validation curve on the other validation dataset.

2.4.2. Data augmentation

Data augmentation is a technique that allows for the creation of mutant data from the original. Because the direction of the gradients is uncontrolled during the BP process, the extracted features are still fuzzy in the early stages, the network's learning process seeks common information while comprehending unrelated different information, the mutant data fit the purpose and must retain some key information.

When it comes to computer vision tasks, data augmentation is primarily concerned with spatias visual information. Because even a single pixel change in the data may be interpreted as a new sample by the network, the network can determine whether or not this pixel is significant. Thus, spatial modification of data that are relevant to our knowledge will aid in the network's generalization throughout the learning phase.

Cropping, shifting, rotating, and mirroring are examples of typical spatial augmentation. Additionally, the mutant samples' brightness, sharpness, and RGB values may be changed.

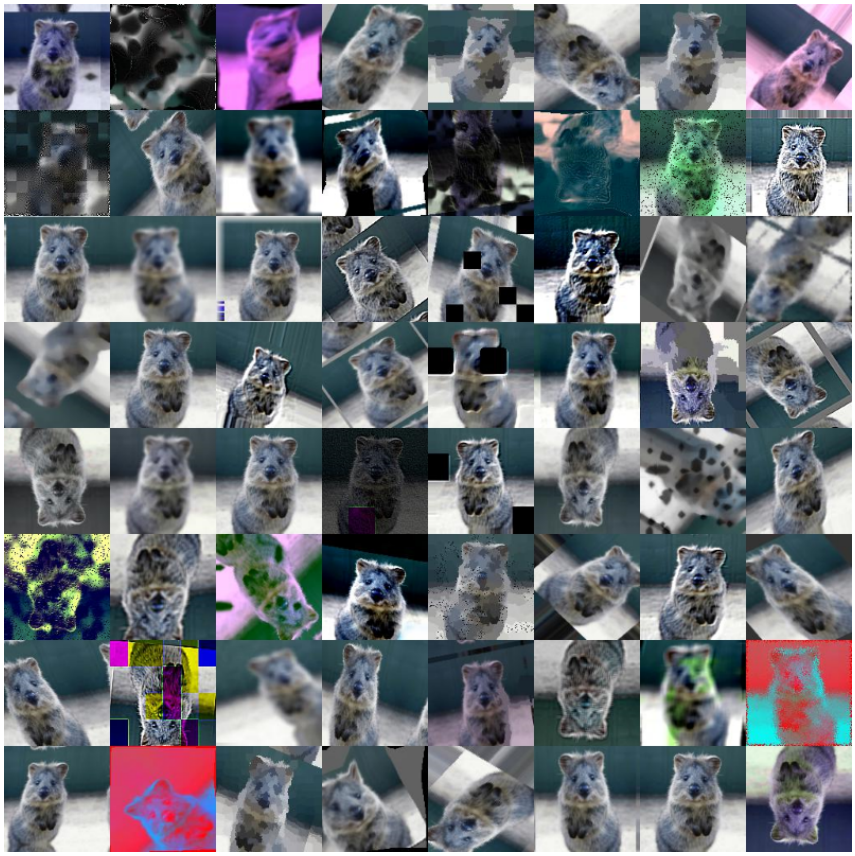


Figure 2.11. Data augmentation in image pixel domain and image color space.(Image source: `https://github.com/aleju/imgaug`)

2.5. Convolution Neural Network

Convolutional neural networks were created by combining convolutional kernels with deep machine learning. Taking use of convolution operations' pattern recognition capability, the concept of training deep filters that extract patterns adaptable to the target function rather than preset filters was suggested.

2.5.1. Artifical convolution kernels

In contrast to the preset convolution kernels (see Figure 2.12 for artificial kernels), the initialized convolution kernels have values that are random depending on certain distributions. Their weights are updated throughout the BP process using the trained samples. Through

the use of several kernels operating in parallel, we are able to extract features from various channels containing varying amounts of data using mapping functions. After generalizing the model, certain artificial convolution kernels extract some patterns that humans recognize, such as edges and colors, but some of these patterns are only numerically relevant to the objective function.

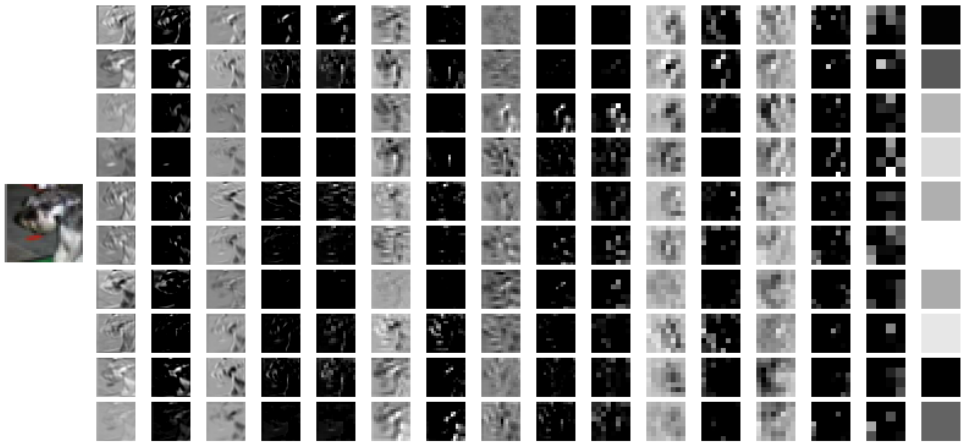


Figure 2.12. An example of trained CNN and the middle features extracted by the trained kernels.(Image source: `http://cs231n.stanford.edu/`)

2.5.2. Classical Convolution Architecture

CNN has grown significantly during the last several decades. Each every CNN family proposes a new method for increasing the efficiency and robustness of CNN.

1. Alexnet [11] won the 2012 ImageNet Visual Recognition Large Scale Challenge. Other successors have been enlightened by the convolution layers coupled with Maxpooling layers, dropout regulation, and ReLU activation function. Convolution kernels operating at various feature resolutions, along with pooling operations, make it not just deep, but also computationally executable on the GPU.
2. VGG [12] is unique in that it uses a 3 by 3 convolution kernel and padding on the input matrices; the combination of these parameters maintains the same resolution before and after convolution. As a result, the VGG network can stack much more convolutional and pooling layers than the Alexnet. VGG-19 has 19 convolution layers, while Alexnet contains just 6. Following that, the specific configuration of the convolution kernels was extensively adopted by other CNNs.
3. Resnet [13] adheres to the VGG architecture and pushes the network's depth even further; they discovered that simply stacking the convolution layers on top of the VGG network degrades network performance; the authors hypothesize that the network degenerated due to the convolution layers' lengthy mapping paths. They suggested a residual link between certain convolution layers as a workaround. The shortcuts allow for the avoidance of duplicate learning in the intermediate levels while maximizing the number of layers. As a consequence, Resnet-152 stacked a total of 152 levels.

4. Instead of using the same size kernel in various feature channel operations, the Inception network [14] uses a mix of different size kernels in distinct feature channels. As a result of the expanded kinds of convolution kernels used in the Inception network, the deep features retrieved by the convolution layers are more divergent. Additionally, the authors argue that by using 1x3 kernels followed by 3x1 kernels, the convolution operations may generate the same size deep features while using less computing resources than a single 3x3 kernel convolution operation.

2.6. Recurrent Neural Network

2.6.1. Naive Recurrent Neural Network

The capacity of the recurrent neural network (RNN) to process temporal sequences and display temporal information is well-known. In comparison to a conventional ANN, RNNs include extra internal memory weights that interact with all input sequences. Each iteration modifies the state of the RNN cell, which acts as an extra input during the RNN's forward function.

$$\begin{aligned} c_t &= \sigma\left(x_t \times W_{xt} + b\right) \\ c_t &= \sigma\left(W_{rec} \cdot c_{t-1} + W_{in} \cdot x_t + b\right) \end{aligned} \tag{4}$$

Due to its flexibility and broad definition, RNN has been used to a variety of applications. The RNN cell may either accept many sequences as input and create a single output describing the temporal events (many to one) or it can be used as an online solution, using each output produced from the input sequences (many to many). RNNs have been used to recognize speech [15, 16], and handwriting [17]

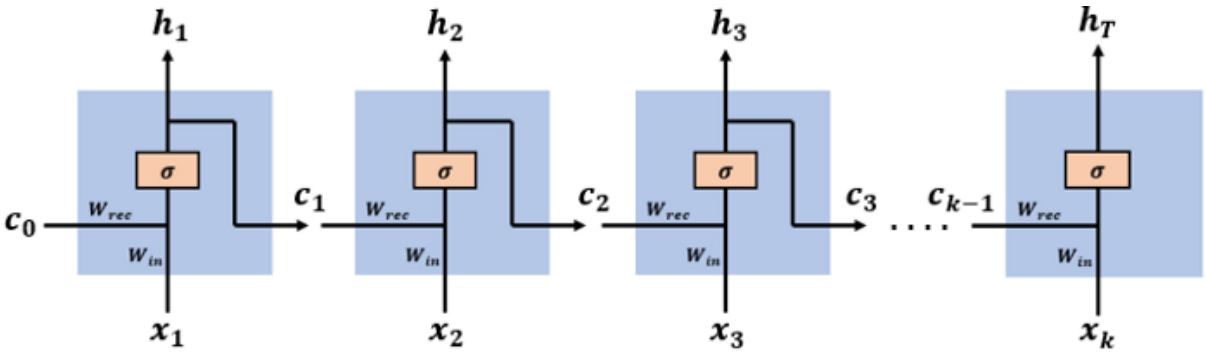


Figure 2.13. Recurrent neural network. (Image source: `https://medium.datadriveninvestor.com/`)

The deficiency of RNN has been noticed that it doesn't handle long sequences data well by its backpropagation. The backpropagation of RNN is called Backpropagation Through Time (BPTT). From the Equation 5, we can understand that if the sequences are long enough and $\frac{\partial C_t}{\partial C_{t-1}}$ are likely to be smaller than 1, the gradient will vanish.

The shortcoming of RNN is that it does not perform well with lengthy sequences of data due to its backpropagation. Backpropagation Through Time (BPTT) is the term used to

describe RNN backpropagation. We may deduce from Equation refbptt that if the sequences are sufficiently lengthy and $\frac{\partial C_t}{\partial C_{t-1}}$ is likely to be less than 1, the gradient will disappear.

$$\begin{aligned}
\frac{\partial E_k}{\partial W} &= \frac{\partial E_k}{\partial h_k}\frac{\partial h_k}{\partial c_k}\cdots\frac{\partial c_2}{\partial c_1}\frac{\partial c_1}{\partial W} \\
&= \frac{\partial E_k}{\partial h_k}\frac{\partial h_k}{\partial c_k}\left(\prod_{t=2}^{k}\frac{\partial c_t}{\partial c_{t-1}}\right)\frac{\partial c_1}{\partial W} \\
\frac{\partial C_t}{\partial C_{t-1}} &= \sigma'(W_{rec}\cdot c_{t-1}+W_{in}\cdot x_t)\cdot\frac{\partial}{\partial C_{t-1}}\left[W_{rec}\cdot c_{t-1}+W_{in}\cdot x_t\right] \\
&= \sigma'(W_{rec}\cdot c_{t-1}+W_{in}\cdot x_t)\cdot W_{rec}
\end{aligned} \tag{5}$$

2.6.2. Long Short-term Memory

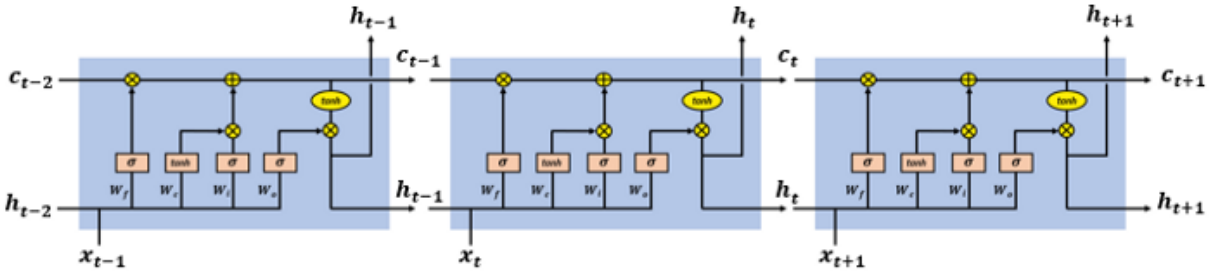


Figure 2.14. LSTM. (Image source: `https://medium.datadriveninvestor.com/`)

LSTM is designed to solve the gradient vanishing problem inside the naive RNN. It's then widely used in Natural Language Processing (NLP) area [18, 19], LSTM has shown superior performance combining with other architectures, such as CNN-LSTM to solve image captioning tasks [20, 21] or video processing tasks [22, 23].

The LSTM was created to address the gradient vanishing issue inside a naive RNN. It is then widely used in the field of Natural Language Processing (NLP) [18, 19]. LSTM has demonstrated superior performance when combined with other architectures, such as CNN, to solve image captioning tasks [20, 21] or video processing tasks [22, 23].

To address the gradient vanishing issue, LSTM included additional pathways for the gra$\frac{\partial C_t}{\partial C_{t-1}}$ is less than one.

$$c_t = c_{t-1}\otimes f_t \oplus \tilde{c}_t \otimes i_t \tag{6}$$

Forget gate:

$$f_t = \sigma\left(W_f\cdot\left[h_{t-1,x_t}\right]\right) \tag{7}$$

Input gate:

$$\begin{aligned}
&\tanh(W_c\cdot[h_{t-1},x_t])\otimes\sigma(W_i\cdot[h_{t-1},x_t]) \\
&\tilde{c}_t = \tanh(W_c\cdot[h_{t-1},x_t]) \\
&i_t = \sigma(W_i\cdot[h_{t-1},x_t])
\end{aligned} \tag{8}$$

Output gate:

$$\begin{aligned} o_t &= \sigma\left(W_o \cdot [h_{t-1}, x_t]\right) \\ h_t &= o_t \otimes \tanh\left(c_t\right) \end{aligned} \tag{9}$$

2.7. Transformer

The transformer is inspired by the cognitive attention process. Similarly to how human brain prioritizes essential information while fading out irrelevant data, attention enables the NN to comprehend the critical characteristics behind our intended system response. The attention mechanism is initially applied in RNN in order to aid in addressing the gradient vanishing issue for the input of the lengthy sequence, by searching for the most critical temporal information for a single input sequence across the remainder of the input sequences.

The mechanism is implemented in such a way that the importance of the information is distributed and denoted by α_i, where h_j denotes the RNN's output sequences and $e_i j$ denotes the output scores that determine the relationship between the input at sequence j and the output at sequence i.

$$e_{ij} = a\left(s_{i-1}, h_j\right) \tag{10}$$

There are two methods to calculate α_i. The first method uses the weighted sum of the annotations to generate the context vector c_i:

$$c_i = \sum_{j=1}^{T_x} \alpha_{ij} h_j \tag{11}$$

where c_i is the processed context vector considering the weights from the original vectors. We consider the sum of the α_{ij} to be one, thus:

$$\alpha_{ij} = \frac{\exp\left(e_{ij}\right)}{\sum_{k=1}^{T_x} \exp\left(e_{ik}\right)} \tag{12}$$

The second way, proposed by the authors of Transformers, defines three matices for the attention units, where query W_Q, key W_K, and value W_v matrices are specified. Then each input context vector is producing their own value:

$$\begin{aligned} q_i &= x_i W_Q \\ k_i &= x_i W_K \\ v_i &= x_i W_V \end{aligned} \tag{13}$$

The attention weight α_{aj} is then calculated by the dot product of q_i and k_j, also suggested by the authors, the attetion weights will be more stablelized after the division of the square root

of the demension of the key vectors $\sqrt{d_k}$. The attention mechanism has the final form:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^{\text{T}}}{\sqrt{d_k}}\right)V \tag{14}$$

The transformer has extended the issue of context vector relationships to include not only temporal but also geographical vectors. This concept was subsequently included into the Vision Transformer (ViT) [24], in which the authors eliminated convolutional kernels in favor of simple linear kernels with attention mechanisms for visual pattern recognition tasks.

3. Literature Review

3.1. Face detection

Face detection is a critical pre-processing step that allows for the extraction of Regions of Interest (ROI) from picture samples. Because neural solutions involve a great deal of computing, a correctly recovered ROI may decrease computation proportionately to the face area across the whole picture region. However, the ROI extraction process must be precise; misprediction or partial omission of the facial region may result in skewed data from the samples.

3.1.1. Traditional methods for Face detection

In the face-related applications, the describing features are given the goal. Historgram Oriented Gradients (HOG) features are extracted based on the histogram of the appeared vectors. Eigenface [25, 26] used in face recognition looks for the characteristics of the face from the known database by its eigenvectors, the vectors are then projected to a lower-dimensional space using PCA. Fisherface [27, 28], improved from Eigenface, furtherly exploits LDA to maximize the ratio of between-class scatter matrix and the within-class scatter matrix.

The aim of face-related applications is to identify the descriptive characteristics. On the basis of the histogram of the appeared vectors, Historgram Oriented Gradients (HOG) characteristics are retrieved. Eigenface [25, 26], which is used in face recognition, searches for the features of a face in a known database using its eigenvectors; the vectors are then projected to a lower-dimensional space using PCA. Fisherface [27, 28], an enhanced version of Eigenface, further uses LDA to optimize the ratio of the between-class and within-class scatter matrices.

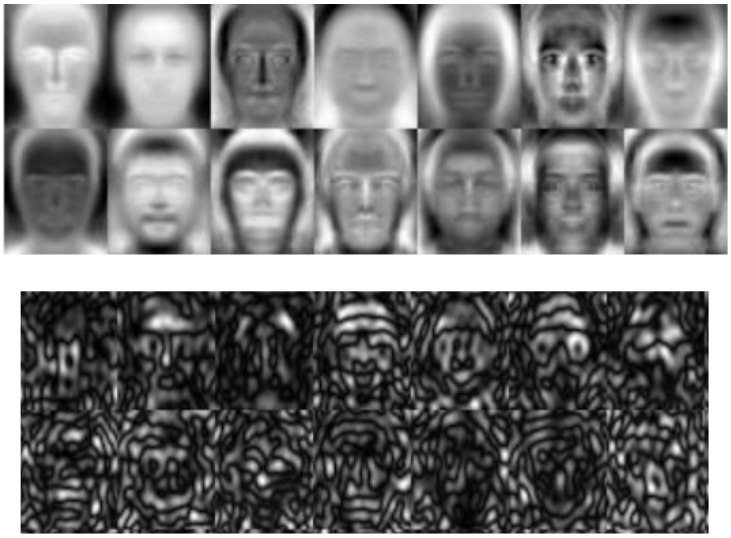


Figure 3.1. Examples of eigenface (top) and fisherface (bottom).

Face detector of King[29] available in `dlib` library[30], which use the bag of Histogram of HOG features, [31] combined with linear SVM for image rectangles of pixels. The HOG

features are widely used for many detections tasks. Dalal, N. and Triggs, B. [32] extract HOG features for human detection in the images. Baumann.F [33] used the HOG features for action classification.

King's[29] face detector is provided in the `dlib` library [30], and it utilizes the bag of Histogram of HOG features [31], in conjunction with linear SVM for picture rectangles of pixels. HOG features are extensively utilized in a broad variety of detecting applications. Dalal, N., and Triggs, B. B. [32] extract HOG features from pictures for person detection. Baumann.F [33] classified actions using HOG characteristics.

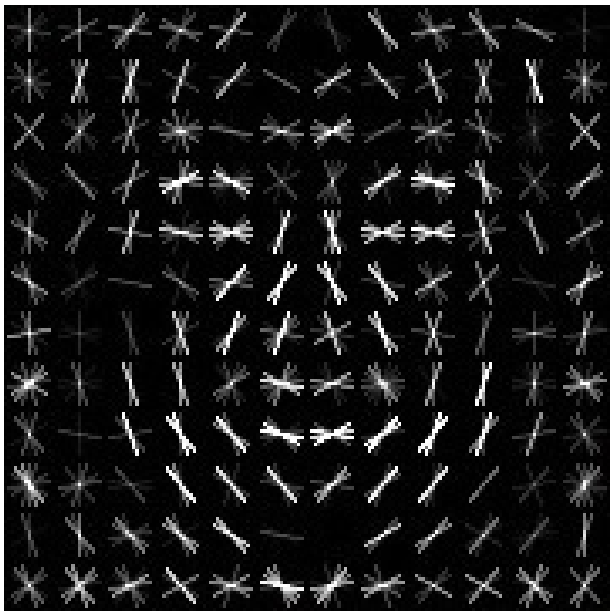


Figure 3.2. HOG features. [32]

In order to avoid the combinatorial explosion, the sophisticated method of selecting image rectangles with objects of interest uses an image pyramid and SSVM trick where only the worst constraints are subjected to relevant quadratic optimization, as well as a greedy heuristic that allows us to obtain suboptimal rectangle configurations for a complex but convex risk function

The SVM model is then trained using a convex optimization problem developed for training collections of picture rectangles called the max-margin convex optimization problem. SVM model demonstrates its benefit in high-dimensional space, which is appropriate for HOG features; it also suffers little from the small sample size in relation to the vector dimensions. SVM determines the optimal width of the gap between two clusters by mapping the high-dimensional samples in the space. The unseen samples are then mapped to the same space depending on their location on each side of the gap.

However, the greedy searching method is computationally intensive. Additionally, the non-frontal HOG features collected may be misclassified to the non-face cluster by the SVM. Certain coverings, such as caps, spectacles, or the individuals' hands, may potentially have an effect on the derived HOG characteristics, resulting in misclassification.

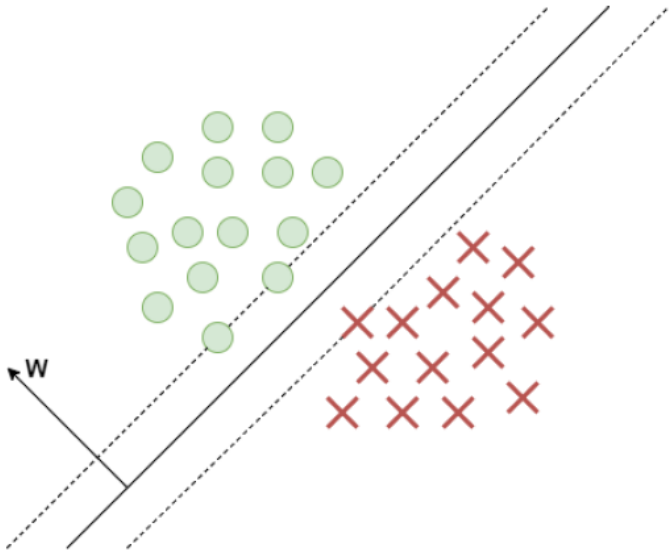


Figure 3.3. Illustration of SVM working as the classifier maximizing the margin between two categories.

3.1.2. Neural approach for Face detection

The most important distinction between the neural method and the conventional solution is that the characteristics retrieved for face identification are more generic, or "deeper", than those derived using preset mapping functions. Due to their robustness for pattern extraction, CNNs are often employed for face detection [34, 35].

Because the features for face-related tasks, such as face detection, facial landmark detection, and face identification, all share the same raw inputs, the CNN extractors are powerful enough to extract deep features to fulfill all of the above-mentioned proposals simultaneously in a real-time solution [36].

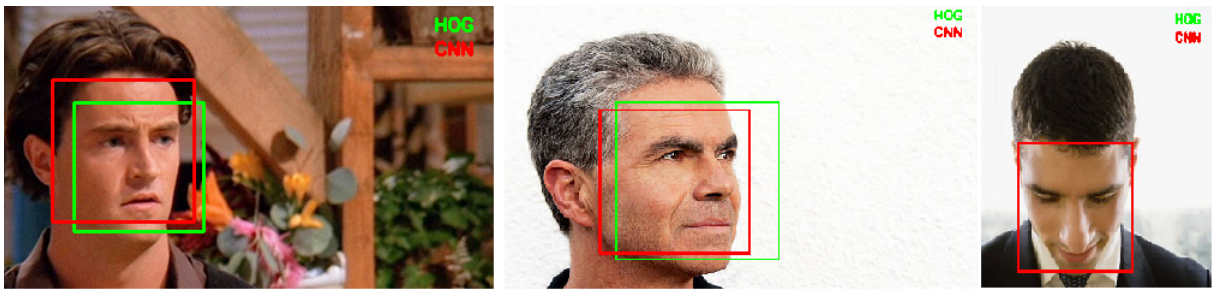


Figure 3.4. Face detection: deep features vs HOG features. Image source: `towardsdatascience`

Face detection neural solutions are capable of more than simply detection; they can also anticipate faces. In conventional methods, such as HOG features, the issue is not just where the face HOG features are, but also where they are not; the covering objects of the faces or non-facial features, having a detrimental effect on face prediction. However, in the

neural solutions, we can train the network to predict the face area based on the incomplete information by simply erasing random [37] rectangles on the faces in the training sets. Rather than being an impediment, this covering aided in the generalization of the CNN face detectors.

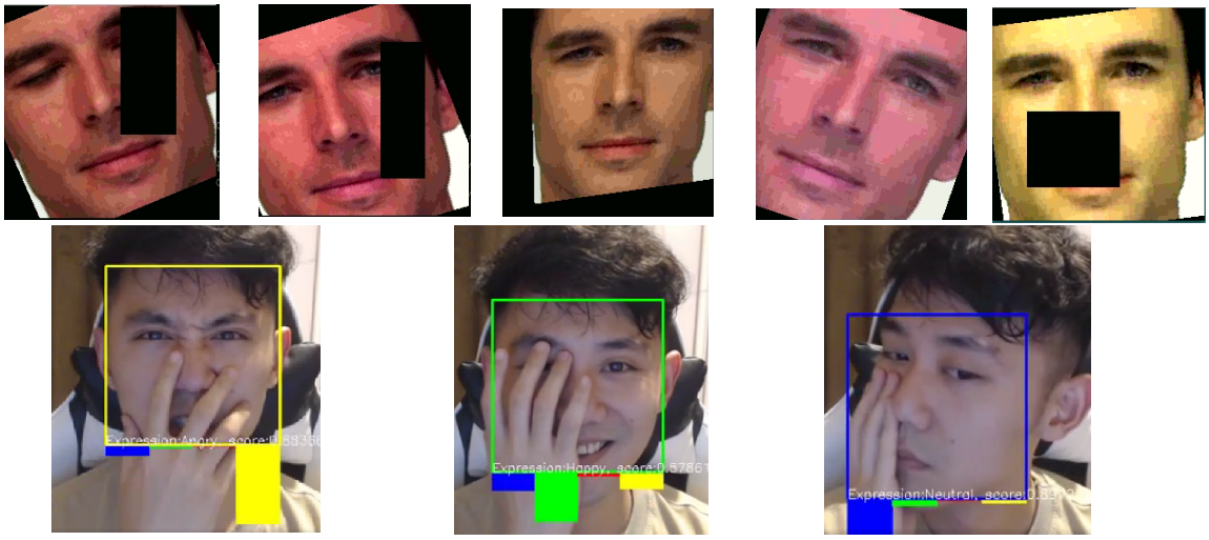


Figure 3.5. Random erasing in the data augmentation and its benefited results.

As shown in Figure 3.5, deep features as descriptors are capable of detecting non-frontal faces, while conventional features are not sufficiently generalized to do so. Due to the superiority of this behavior, CNN systems are much more resilient than conventional techniques in real-time face identification applications.

3.2. Facial Emotion Recognition

Emotion recognition utilizes spatial information extracted from the picture input, namely the ROI after face identification. In essence, recognition is a categorization of emotional categories. The emotion recognition process assigns a particular category to the input picture based on the number of preset classes.

3.2.1. PCA and LDA for Facial Emotion Recognition

Similar to face detection, traditional emotion recognition algorithms can also exploit the extracted facial features. [38, 39, 40] have all adopted the PCA algorithm to extract the features for emotion recognition. Since facial emotion recognition also takes similar inputs, human face images, thus it's not surprising that Fisherface [41, 42] have also been adopted for facial emotion recognition.

In a similar fashion to face detection, conventional emotion identification algorithms may make use of retrieved facial characteristics. [38, 39, 40] all used the PCA method to extract emotion recognition features. Given that facial emotion identification likewise requires comparable inputs, human face pictures, it's unsurprising that Fisherface has been used for facial emotion recognition.

3.2.2. 3D Modeling for Emotion Recognition

Three-dimensional modeling attempts to depict the intricacies of face components in order to identify facial expression using the information contained in three-dimensional vertices. To do this, FACS is used to define movements based on the muscle types established by P. Ekman and WV. Friesen[2].

The FACS deconstruct expressions into Action Unit (AU) combinations; each AU is subsequently produced by a series of muscular movements. Figure 3.6 illustrates many facial muscles that govern various facial movements.

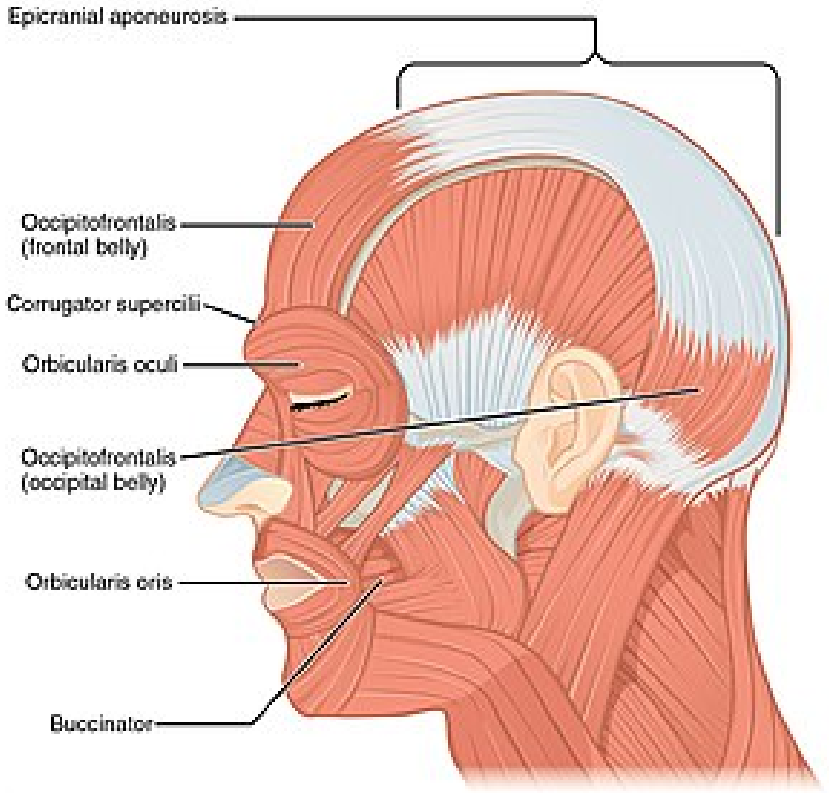


Figure 3.6. Facial muscles important in forming facial actions relevant to visual emotions.(Image source: `https://openstax.org/books/anatomy-and-physiology/pages/preface`)

FACS enables us to interpret face emotion expressions into meaningful muscle groups, as shown in Figurerefauexample. For example, a grin may be interpreted as the movement of AU6 + AU12, where AU6 is the cheek raiser and AU12 is the Lip corner puller.

With the FACS, the job of facial expression recognition was transformed into an AU detection task. We integrated 3D modeling and FACS to solve the AU identification problem, not only detecting the kind of AUs, but also specifying the scatter values of the discovered AUs as the final feature vectors for emotion categorization.

To represent FACS in computer vision, a three-dimensional vertex model, Candide-3 [43], is used as Figure 3.8 shows. Not only the model, but also the user's subjective information, such as facial breadth, mouth position, and nose position, may be used to modify these settings. Additionally, it has preset AU vectors for the three-dimensional vertices. We want to

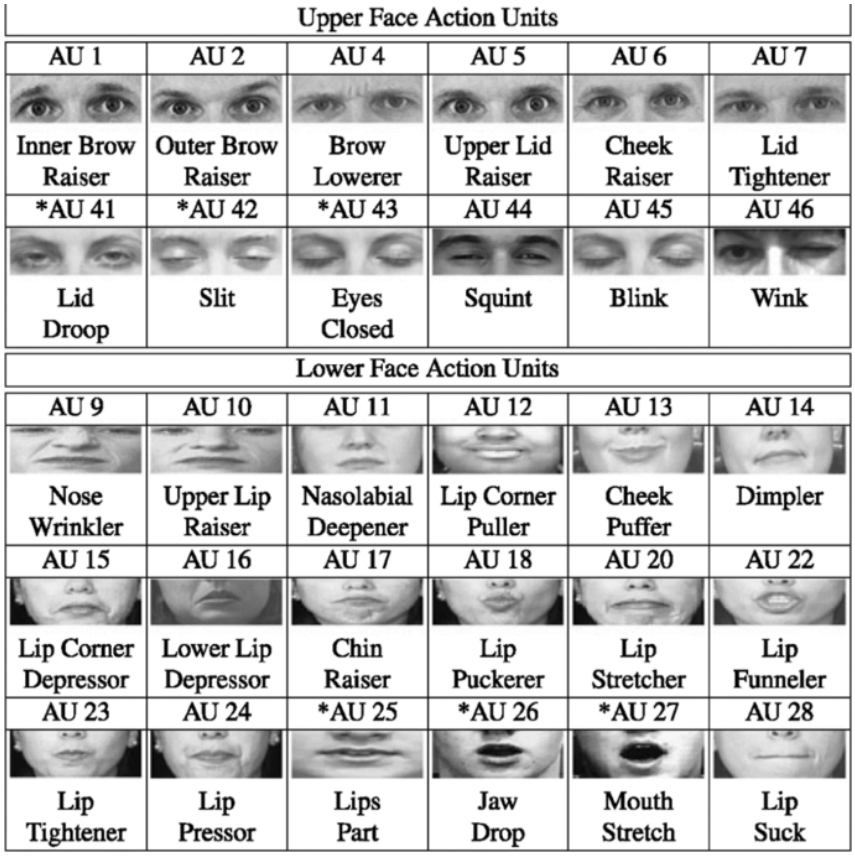


Figure 3.7. Facial Action Units extacted from Cohn and Kanade dataset.

rebuild 3D information from 2D picture data and extract AU information for face emotion detection using the Candide-3 model.

3.3. Transfer learning

With a similar idea of Neural Networks mimicking the brain's neural cells in order to learn, transfer learning may be achieved. Transfer learning of neural networks alters the underlying knowledge of the network and uses it to enhance the learning of a new task in the same way that a person recognizes and applies relevant information from prior learning experiences when confronted with new tasks.

According to the results of many neural networks trained with natural images, the features learned from lower layers of networks are similar in nature regardless of the task or data categories, whereas the features learned from higher-level layers are specific to the tasks [44, 45]. Transfer learning is a technique in which a network is trained on a source task and source data, and then the acquired characteristics are re-purposed to a target task and target data.

Assume we have a source domain $D_S = \{X_S, f_S(X)\}$, and a source task T_S, as well as a target domain $D_T = \{X_T, f_T(X)\}$, and a target task T_T. Transfer learning may significantly aid in the learning of the target prediction function $f_T(\cdot)$ in D_T using knowledge of D_S and T_S, where $D_S \neq D_T$ and $T_S \neq T_T$, respectively. Transfer learning has shown significant benefits in

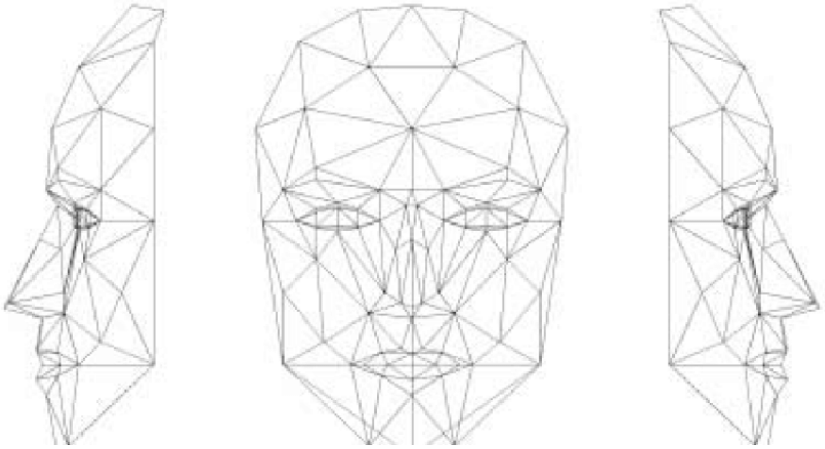


Figure 3.8. Candide-3 model.

reality, since few individuals train a deep network from scratch, as they seldom come across a database that is adequate for a particular task.

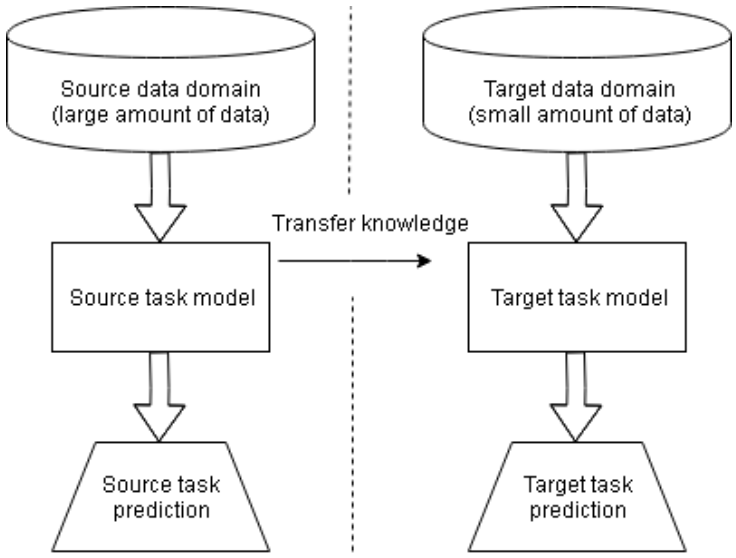


Figure 3.9. General concept of transfer learning.

3.4. Emotion recognition from streaming video

The method for emotion identification from streaming video incorporates temporal information derived from the discrete data collected. Numerous neural architectures attempt to accomplish this objective. One possibility is to utilize 3DCNN [46, 47], which extracts spatial-temporal information through four-dimensional tensors from networks. The frames are stacked and handled as the time dimension, which shapes the data as $[H, W, C, T]$, where H denotes the height, W denotes the width, C denotes the channel, and T is the time sequence. However, increasing the size of the network tensors and inputs significantly increases the computing cost and makes generalization more difficult.

Another solution is based on 2DCNNs; unlike single-frame predictions, temporal information can be extracted by averaging the deep features extracted from each frame of the 2DCNN [48]. While such temporal segment networks demonstrate their potential for video event classification, they do not always perform better when dealing with events where details matter, such as facia.

People also try to combine LSTM with 2DCNN, separating their role of processing the data, thus the 2DCNN doesn't need to extract any new patterns but focusing on spatial features only while leaving the temporal information to the LSTM. Such a combination has been proved from many works. [7, 49]

Additionally, some people attempt to integrate LSTM and 2DCNN by decoupling their roles in data processing, such that the 2DCNN does not need to extract any new patterns but may instead concentrate on spatial characteristics while leaving the temporal information to the LSTM. Numerous works attest to this mix. [7, 49]

3.5. Emotion recognition from streaming audio

Apart from visual face information, aural data processing also attracts researchers' interest.

Similarly to face emotion recognition, conventional SER systems use feature extraction and categorization methods. Traditional classification methods such as the Gaussian mixture model (GMM) [50, 51], the support vector machine (SVM) [52, 53] and the hidden Markov model (HMM) [54] make use of the retrieved characteristics to accomplish the ultimate objective.

The neurological approach to the SER has also flourished during the last several decades. RNNs [55, 56], LSTMs [57], and gated recurrent units (GRUs) [58] have been used to push the SER SOTA findings in the time domain.

Others, inspired by CNN's better ability to uncover patterns, find the spectrogram and Mel-spectrogram matrices helpful as straight raw inputs. CNNs treat the spectrogram matrices as pictures; the temporal information was converted to pure spatial information and stored in the matrices after conversion to the time-frequency domain. [59, 60] have all used similar SER solutions. The complex spectrogram has shown significantly better performance in speech-related applications such as speech enhancement [61].

3.6. Multi-modal solution for Audio-Video Emotion recognition

Many studies have shown a substantial increase in the effectiveness of multi-modal solutions. For example, N. Neverova et al. [62] propose progressive fusion including the random dropping of individual channels, and this technique was used by V. Vielzeuf and colleagues [63] in their AVER solution to get the best possible outcome.

Others have raised the question of whether fusion should occur early or late in the process. When it comes to multi-modal feature fusion, R. Beard and colleagues [64] proposed it at the end of the process; on the other hand, E. Ghaleb and colleagues [65] attempted it at the beginning and provided external loss functions to minimize the distance between features

from different modalities. The Multi-view Gated Memory presented by A. Zadeh et al. [66] was designed to gate the multi-modal information from LSTM into the time series. In their paper [67], E. Mansouri-Benssassi and J. Ye describe how they archive early fusion by forming separate multi-modal neuron groupings.

In their paper, S. Zhang et al. [68] extract features from CNN and 3D-CNN models for voice and visual sources, and then use global averaging to produce video features. NC. Ristea et al. [69] combine the features collected by CNNs from both modalities and utilize the resulting fused features to classify objects in the environment. E. Tzinis and colleagues [70] use cross-modal and self-attention modules in their research. Y. Wu et al. [71] locate occurrences that span several modes of transportation. E. Ghaleb et al. [72] propose multi-modal emotion recognition metric learning to build a robust representation for both modalities, with the goal of improving overall performance.

4. Proposed methods

4.1. Facial Emotion Recognition via 3D modeling

The reconstruction of 3D information from a 2D picture requires 2D knowledge about the 3D vertices. Thus, we presented a method for detecting 2D landmarks for 2D face points based on the Candide-3 model's 3D facial points. The development of the 3D modeling are described in detail in [112, 113].

To do this, we construct 68 facial salient points (fp68) (cf. Fig.4.4), from which we extract the identified facial landmarks information from the camera data. The point detection is comparable to face detection and facial expression identification in that it uses the same raw data from the input picture.

The `dlib` library's fp68 detector is used. Similarly to the face recognition algorithm in the `dlib` package, the detector makes use of the HOG features mapping from 68 HOG feature vectors to pixel level position prediction.

HOG to plane mapping is defined via regression trees designed for all 68 fp68 as Figure 4.1 using cascade approach [73, 74]. The use of many small regression trees gives a more effective detector than using one large regression model. The trees are built using stochastic gradient boosting of Friedman [75].

The HOG to plane mapping is created using regression trees constructed for every 68 fp68 as shown in Figure 4.1 utilizing a cascade method [73, 74]. The employment of a large number of tiny regression trees results in a more effective detector than the use of a single big regression tree. The trees are constructed using Friedman's stochastic gradient boosting technique [75].

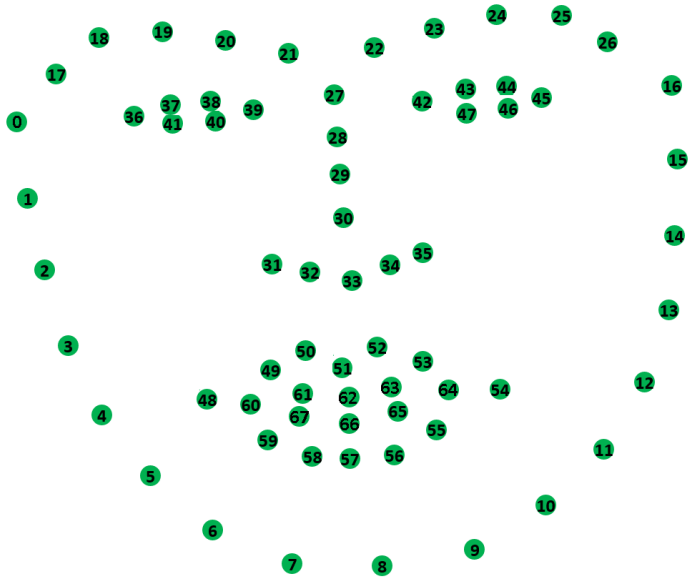


Figure 4.1. Facial feature points indexed in FP68 categorization.

The 3D reconstruction is then carried out by estimating the 3D vertices using the 2D landmarks as a source of information and restrictions on the model's spatial position and deformation. 46 of the 182 3D vertices are chosen because they provide information on certain face prominent spots.

```
# 46 selected 2D indexes of dlib fp68
idxs2 = [35, 31, 30, 33, 28,     # nose
         36, 37, 38, 39, 40, 41,         # left eye
         42, 43, 44, 45, 46, 47,         # right eye
         17, 19, 21,                     # left brow
         22, 24, 26,                     # right brow
         48, 51, 54, 57, 53, 49, 55, 59, # outer lip outline
         60, 62, 64, 66,                 # inner lip outline
         ]
# corresponding 3D indexes of Candide-3
idxs = [26, 59, 5, 6, 94,
        53, 98, 104, 56, 110, 100,
        23, 103, 97, 20, 99, 109,
        48, 49, 50,
        17, 16, 15,
        64, 7, 31, 8, 79, 80, 85, 86,
        89, 87, 88, 40,
        ]
```

The model's deformation is controlled by preset AU vectors extracted from MPEG-4 related papers [76]. The following are the entities that exchange information with our 2D information provider:

1. Action units :
   a) AU 26/27: jaw drop
   b) AU 4: brow lower
   c) AU 13/15: lip corner depressor
   d) AU 10: upper lip raiser
   e) AU 20: lip stretcher
   f) AU 7: lid tightener
   g) AU 9: noise wrinkler
   h) AU 42/43/44/45: eye closed

   The model's deformation is then used to define the expressions, which are parameterized using a mix of AU vectors and their numerical values. We anticipate that the AU values will serve as our ultimate descriptors of emotional features.
2. Shape units :

   a) SU 1:Eye brows vertical position
   b) SU 2: Eyes vertical position

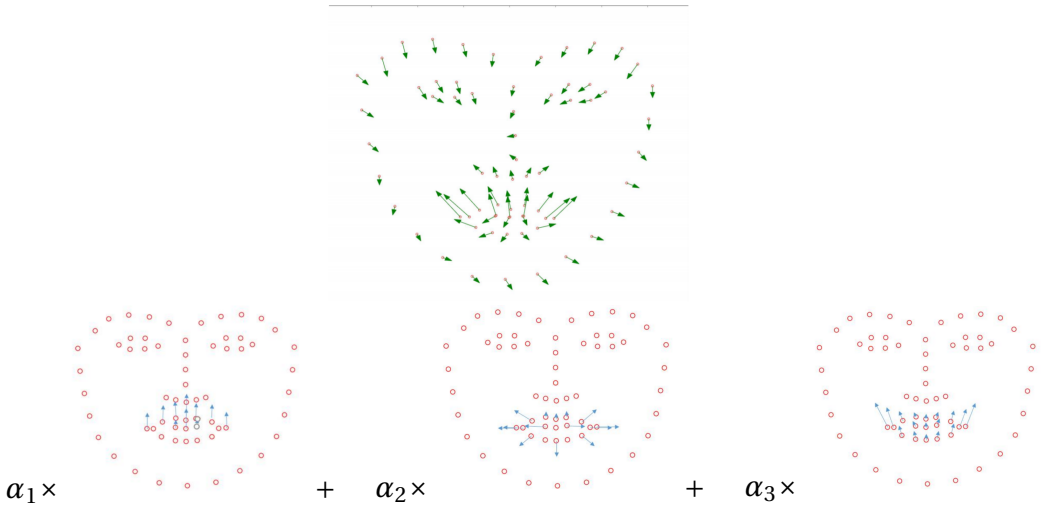


Figure 4.2. Facial action units of mouth for smiling: (left) AU10 – upper lip raiser; (middle) AU20 – lip stretcher; (right) AU13/15 – lip corner depressor. The linear combination $\alpha_1 \times \text{AU10} + \alpha_2 \times \text{AU20} + \alpha_3 \times \text{AU13/15}$ roughly approximates the mouth motion of smile.

c) SU 3: Eyes width
d) SU 4: Eyes separation distance
e) SU 5: Mouth vertical position
f) SU 6: Mouth width
g) SU 7: Eyes vertical differences

Apart from the AU deformation used to depict muscle movements from the 2D fp68 points, the Shape Unit (SU) distortion is used to determine the subjective information about the face components. The SU deformation gathers user input and transforms the original 3D model into a user-specific one, which aids in the optimization stage's 3D reconstruction.

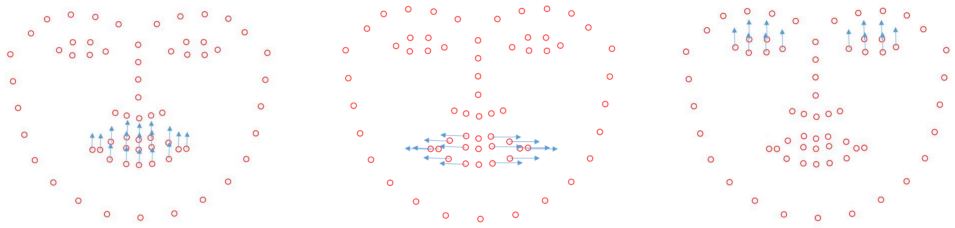


Figure 4.3. Shape units: (left) – mouth vertical position; (middle) mouth width; (right) eyes vertical position.

Our suggested techniques do a two-step calculation of the affine coefficients. SU collects personal information from those who have a neural expression. Thus, the Candide-3 model's deformation is represented only via the SU that customize the model. The distance between the projected 3D model and fp68 markers is then minimized using an optimizer based on the

Affine transform and AU deformation of the 3D model.

$$P_i^g(\tau) \doteq \begin{bmatrix} X_i^g(\tau) \\ Y_i^g(\tau) \\ Z_i^g(\tau) \end{bmatrix} = s^g(\tau)R^g(\tau)\left(\begin{bmatrix} X_i^g \\ Y_i^g \\ Z_i^g \end{bmatrix} + \sum_{d\in[D]:i\in I_d} \alpha_d(\tau)a_i^d\right) + t(\tau),\ i \in [G] \tag{15}$$

where i is the index of the point in Candide-3 model with G points of global estimation; I_d is the index set of points for the deformation $d \in [D]$, where $D \in G$ selected for pose and individualization; $a_i^d \in \mathbb{R}^3$ is the unit deformation vector[1] being the column of the deformation matrix A_d^c which is assigned to the point i at the deformation d, $A_d^c \in \mathbb{R}^{3\times|I_d|}$; the notation $d \in [D] : i \in I_d$ selects for the summation only those deformations d which refer to the point i.

$$P_i(\tau) \doteq \begin{bmatrix} X_i(\tau) \\ Y_i(\tau) \\ Z_i(\tau) \end{bmatrix} = \begin{bmatrix} X_i^g(\tau) \\ Y_i^g(\tau) \\ Z_i^g(\tau) \end{bmatrix} + s^g(\tau)R^g(\tau)\left(\sum_{f\in[F]:i\in I_f} \alpha_f(\tau)a_i^f\right),\ i \in [ht] \tag{16}$$

Where H stands for all 35 points we selected, the scale parameter and rotation matrix are the same as those used in global estimation. I_f is the index set of points for the deformation $f \in [F]$, where $F \in H$ is selected for motion expression. $a_i^f \in \mathbb{R}^3$ is the unit deformation vector.[2]

1. The optimization function's primary purpose is to determine the transformation parameters of the Candide model onto the current face model (local deformations for action and shape units, global scaling, rotation, and translation). To accomplish this objective:
   a) Active point indexes for the 2D and 3D case are established:
      i. Core 3D points which have referenced points in fp68 are selected: J.
      ii. Points for global estimation and individualization are selected from core 3D points:$J_g \in J$
      iii. Indexes of deformation points for shape units are joined to core points: $J_{gd} \doteq \bigcup_{d\in[D]} I_d \cup J_g$
      iv. Active 2D points J_s^2 of facial salient points fp68 having corresponding points in J_{gd} core and deformation points, are selected.
      v. Active 3D points are specified as those points of J_{gd} which correspond to active 2D points: J_s^3.
      vi. Number of active points is registered: $N_s = |J_s^2| = |J_s^3|$.
      vii. The centroid for Candide model is computed: $\bar{P}^g = \begin{bmatrix} \bar{X}^g \\ \bar{Y}^g \\ \bar{Z}^g \end{bmatrix} \doteq \frac{1}{N_s}\sum_{i\in J_s^3} \begin{bmatrix} X_i^g \\ Y_i^g \\ Z_i^g \end{bmatrix} = \frac{1}{N_s}\sum_{i\in J_s^3} P_j^g$.

[1] In Candide-3 model, a_i^d is called the shape unit vector and the matrix A_d^c gathers all vectors for the given action unit.

[2] In the Candide-3 model, a_i^f is called the action unit vector, and the matrix A_d^c gathers all vectors for the given action unit.

b) For the current fp68 shape $p_j(\tau) \in \mathbb{R}^2$, $j \in J_s^2$, the initial values of motion parameters with respect to Candide-3 shape P_i^g, $i \in J_s^3$, are found:

i. Distortion coefficients and rotation:

$$\alpha_d = 0, d \in [D],\ R = I_3 \tag{17}$$

ii. Scaling s:

$$\begin{gathered} s = \arg\min_s \left[\sum_{i \in J_s^3} (x'_{j(i)} - s(X_i^g)')^2 + (y'_{j(i)} - s(Y_i^g)')^2 \right] \longrightarrow \\ s = \frac{\sum_i \left[x'_{j(i)}(X_i^g)' + y'_{j(i)}(Y_i^g)' \right]}{\sum_i \left[(X_i^g)'(X_i^g)' + (Y_i^g)'(Y_i^g)' \right]} \end{gathered} \tag{18}$$

where the 2D/3D centered shapes are defined as follows:

$$\begin{bmatrix} \bar{x} \\ \bar{y} \end{bmatrix} \doteq \frac{1}{N_s} \sum_{j \in J_s^2} \begin{bmatrix} x_j \\ y_j \end{bmatrix} \longrightarrow \begin{bmatrix} x'_j \\ y'_j \end{bmatrix} = \begin{bmatrix} x_j - \bar{x} \\ y_j - \bar{y} \end{bmatrix}, \quad \begin{bmatrix} (X_i^g)' \\ (Y_i^g)' \end{bmatrix} \doteq \begin{bmatrix} X_i^g - \bar{X}^g \\ Y_i^g - \bar{Y}^g \end{bmatrix} \tag{19}$$

iii. Translation t :

$$\begin{bmatrix} x_{j(i)} \\ y_{j(i)} \end{bmatrix} \simeq s \begin{bmatrix} X_i^g \\ Y_i^g \end{bmatrix} + t,\ \ i \in J_s^3 \longrightarrow t = \begin{bmatrix} \bar{x} \\ \bar{y} \end{bmatrix} - s \begin{bmatrix} \bar{X}^g \\ \bar{Y}^g \end{bmatrix} \tag{20}$$

c) Error function is defined:

$$E_\tau(s, w, t, a) = \sum_{i \in J_a^3} \left\| P_i(\tau)\big|_{xy} - p_{j(i)}(\tau) \right\|^2 \tag{21}$$

where $s \in \mathbb{R}$ – scaling parameter; $w \in \mathbb{R}^3$ – the vector representation of the rotation matrix (see the inverse Rodrigues formulas below (23)); $t \in \mathbb{R}^2$ – the translation vector in the xy plane; $a \in \mathbb{R}^D$ – parameters of local deformations; $j(i) = j$ such that $J_s^3[k] = i \longrightarrow J_s^2[k] = j$, i.e. it is the active index of 2D point corresponding to the active index of 3D point; $|_{xy}$ – denotes the orthographic projection onto xy plane.

d) Levenberg Marquardt Method (LMM) optimization procedure is performed for the error function $E(s, w, t, a)$ defined by equation 21 with initialization described above.

2. The function to compute the orthographic projection uses the current transformation parameters. The rotation is represented by 3D vector $w \in \mathbb{R}^3$ representing the rotation angle α in radians $\alpha \doteq \|w\|$, and the rotation axis $u \doteq \frac{w}{\alpha}$. The rotation matrix R is found from the Rodrigues formula.

Namely, let R be the rotation matrix for rotation axis u and rotation angle α. If $x^\top u = 0$ then $Rx = \cos\alpha \cdot x + \sin\alpha \cdot (u \times x)$, otherwise $Rx = R(u^\top x u + x - u^\top x u) = u^\top x R u + R(x - u^\top x u) = u^\top x u + \cos\alpha(x - u^\top x u) + \sin\alpha(u \times (x - u^\top x u))$. Hence $Rx = u u^\top x + \cos\alpha \cdot x -$

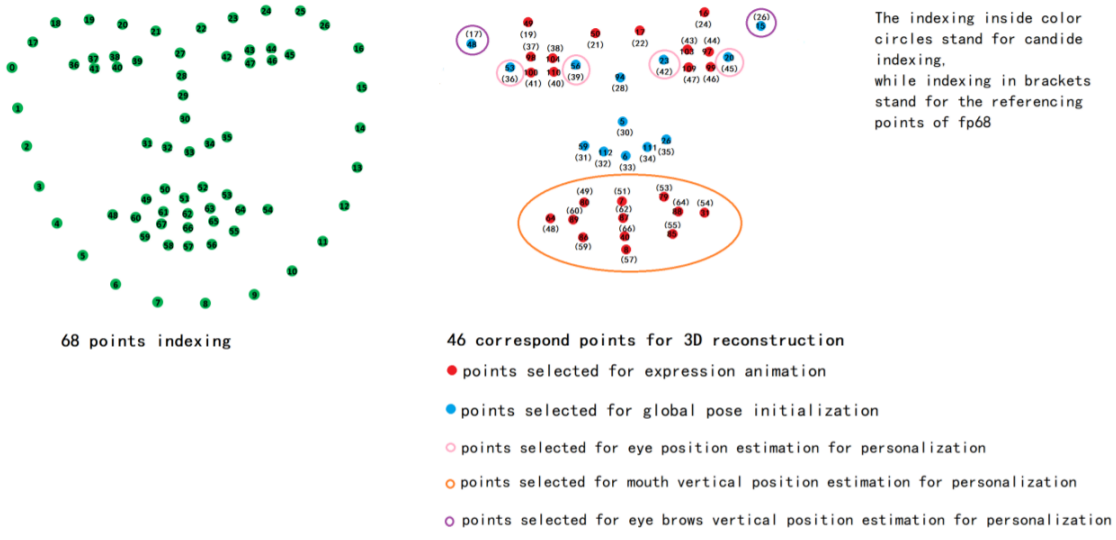


Figure 4.4. Facial points fp68 model and their groups used for animation and personalization.

$\cos\alpha \cdot uu^{\mathsf{T}}x + \sin\alpha(u \times x) = \cos\alpha \cdot x + (1-\cos\alpha)uu^{\mathsf{T}}x + \sin\alpha(u \times x)$. However,

$$u \times x = \begin{bmatrix} 0 & -u_z & u_y \\ u_z & 0 & -u_x \\ -u_y & u_x & 0 \end{bmatrix} x \longrightarrow R = \cos\alpha \cdot I_3 + (1-\cos\alpha)uu^{\mathsf{T}} + \sin\alpha \begin{bmatrix} 0 & -u_z & u_y \\ u_z & 0 & -u_x \\ -u_y & u_x & 0 \end{bmatrix} \tag{22}$$

Note that the rotation angle α and the rotation axis u can be recovered from the rotation matrix by the inverse Rodrigues formulas. They follow directly from the linearity of trace and transposition operations for matrices.

$$\mathtt{tr}\,[R] = 2\cos\alpha + 1, \quad R - R^{\mathsf{T}} = 2\sin\alpha \begin{bmatrix} 0 & -u_z & u_y \\ u_z & 0 & -u_x \\ -u_y & u_x & 0 \end{bmatrix} \tag{23}$$

We evaluate the recognition of the 3D modeling mechanism on the Cohn-Kanade Dataset (CK+)[77] dataset, the visual represention for the online solution is shown in Figure 4.5. The decision is shown by the color frame of the ROI, which is made by the SVM classifier taking the advantage of just 8 AU parameters. The detailed metrices evaluation is presented in the subsection 4.2. The 3D modeling solution for FER solution has visually noticeable deficiency that the 3D reconstruction is based on the 2D facial landmarks information. The performance of the whole system heavily depends on the fp68 detection, while the traditional solution of the fp68 can fail having bad light condition, non-frontal face image, etc. Replacing the

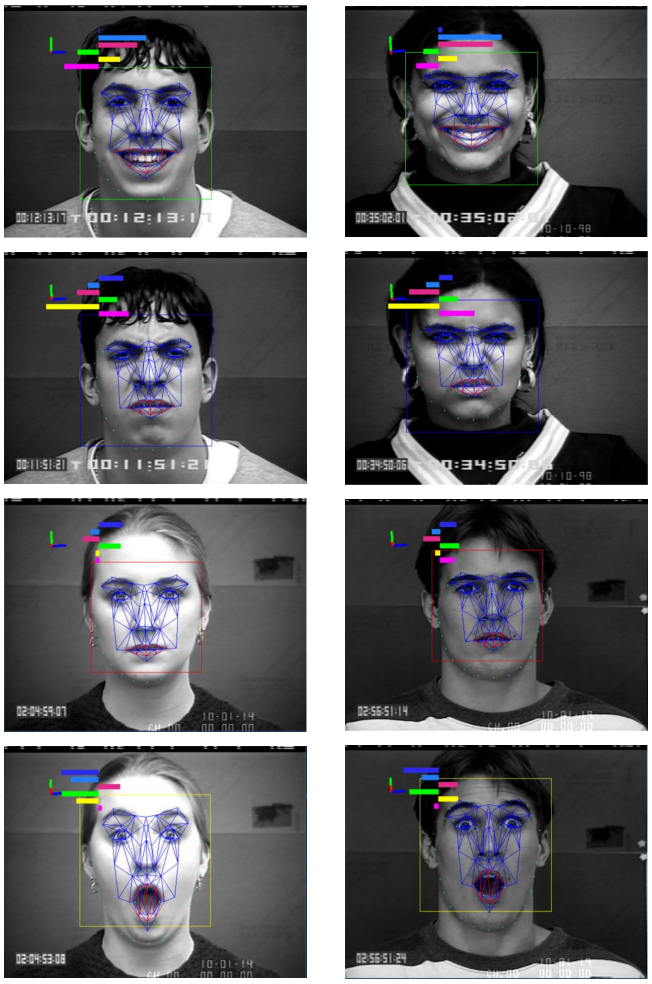


Figure 4.5. Samples from Cohn and Kanade dataset selected to extract facial action units for SSVM classification: the bar length represents the action unit weight while the bar color stands for the class label.

landmarks detection can greatly help for better 3D reconstruction, but the personalization of the Candide-3 model by the SU still make the solution miscellaneous. When the neural fontal face of the subject is unavailable, the recognition rate drop significantly.

On the Cohn-Kanade Dataset (CK+) [77], we assess the identification of the 3D modeling method; a visual representation of the live solution is given in Figure 4.5. The color frame of the ROI indicates the choice, which was determined by the SVM classifier with just eight AU parameters. The subsection 4.2 has a thorough assessment of the metrics. The 3D modeling approach for FER has a visually apparent shortcoming in that the 3D reconstruction is based on 2D face landmark data. The performance of the whole system is highly dependent on the fp68 detection, and the conventional fp68 solution may fail due to poor lighting conditions, non-frontal face images, and so on. While replacing the landmark detection may significantly improve 3D reconstruction, the result remains inconsistent due to the SU's customization of the Candide-3 model. When the subject's neural fontal face is missing, the recognition rate decreases substantially.

4.2. Traditional solution versus neural solution for facial emotion recognition

In this subsection, we present our experiments published in [114]. The experiments aim to compare the solutions of traditional SVM classification peroformance, adopting landmarks and 3D modeling facial features with neural network solutions.

4.2.1. Neural solution replacing SVM classifier

The first part of the experiment examines the classifiers' performance; we investigate for any potential benefits of neural classifiers that can be gained by utilizing traditional facial features. Another point of contention is the feasibility of our application; we conducted experiments to determine its viability. We aim to determine whether recognition systems can adapt appropriately to previously unknown settings, light, image resoltution, camera angle, etc., a third dataset, The Radboud Faces Database (RaFD) [78] is extended to our experimental data to complicate the datasets.

RaFD added 67 new participants and varied the camera angles used to capture the subjects; nevertheless, the non-frontal data made it more difficult for AU and FP68 to properly extract the 2D information, as seen in the lower panel of Figure 4.6. As described in subsection 4.1, these environmental circumstances can make it considerably more difficult to improve landmark identification and 3D reconstruction.

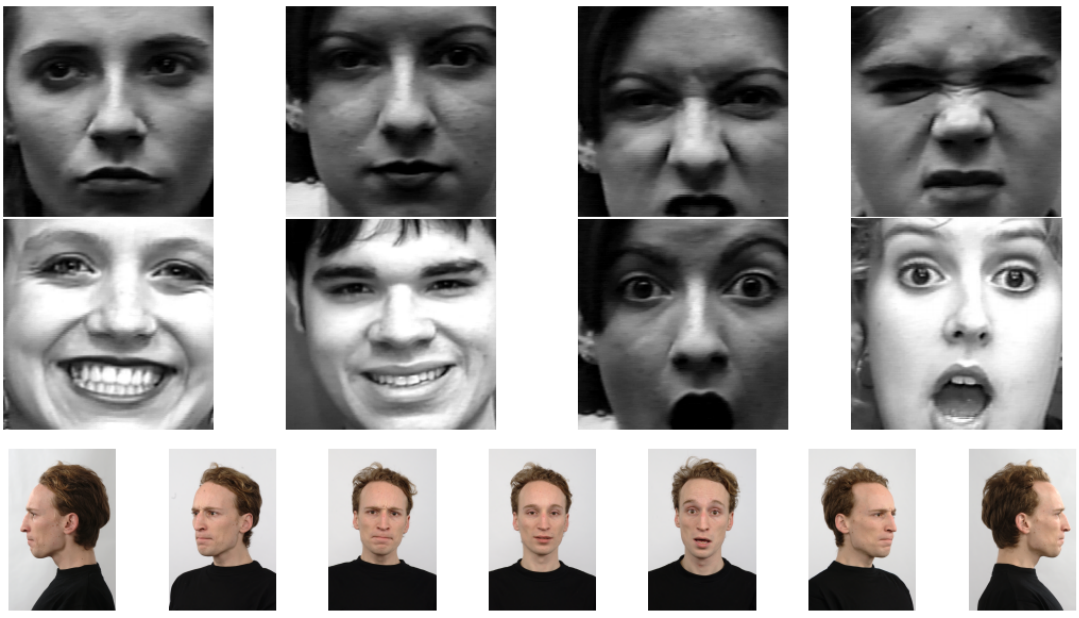


Figure 4.6. Samples from the training dataset (top: Cohn and Kanade dataset.) and testing dataset (bottom: RaFD).

For the extracted features, two DNNs are given. Due to the low dimension of the AU features, only two dense layers are included in the DNN models.

$$\boxed{au8} \mapsto \mathscr{J}_{l}^{a}\ \mathbb{F}_{r}^{8}\ \mathbb{F}^{4}$$
$$\overset{AU8}{\mathscr{NET}}\left(\ au8 := 8_{a};\ optima := [loss, AdaM, SoftMax]\ \right)$$

For FP68, experimental evidence indicates that dropout regularization with a probability of 80% avoids overfitting and achieves the highest recognition rate; the architecture is described as follows:

$$\boxed{fp68} \mapsto \mathscr{J}_{l}^{a}\ \mathbb{F}_{br}^{16}\ \mathbb{D}_{80}\ \mathbb{F}^{4}$$
$$\overset{AU8}{\mathscr{NET}}\left(\ au8 := 8_{a};\ optima := [loss, AdaM, SoftMax]\ \right)$$

The trained DNN classifiers adapted well to the new data; their recognition rate on the testing dataset was almost 34% and 28% higher than the answer provided by SVM classifiers for the AU and FP68 features, respectively.

4.2.2. End-to-end Neural facial emotion classification framework

While DNN solutions shown their generalizability in mapping extracted characteristics, CNN demonstrated its overwhelming pattern recognition capabilities for pictures. We anticipate the same excellent performance from CNN when it comes to extracting face picture patterns. To investigate the performance of networks with varying topologies and complexity, we specify three distinct architectures in our experiments.

CNN-1 is a convolutional network with batch normalization and a non-linear ReLU activation unit that expects images to be 50x50. Following the last convolution layer, global average pooling is used. Due to the previous convolution layer that creates four feature maps, the network does not include a completely linked layer.

$$\boxed{img50} \mapsto \mathscr{J}_{yx}\ {}_{p}\mathbb{C}_{5}^{16}\ {}_{p}\mathbb{C}_{2\sigma5}^{br16}\ {}_{p}\mathbb{C}_{5}^{32}\ {}_{p}\mathbb{C}_{2\sigma5}^{br32}\ {}_{p}\mathbb{C}_{3}^{64}\ {}_{p}\mathbb{C}_{2\sigma3}^{br64}\ {}_{p}\mathbb{C}_{1}^{64}$$
$${}_{p}\mathbb{C}_{2\sigma3}^{br128}\ {}_{p}\mathbb{C}_{1}^{256}\ {}_{p}\mathbb{C}_{2\sigma3}^{br128}\ {}_{p}\mathbb{C}_{1}^{256}\ {}_{p}\mathbb{C}_{2\sigma3}^{4}\ {}_{a}\mathbb{P}_{g}$$
$$\overset{cnn-1}{\mathscr{NET}}\left(\ img50 := 50_{yx};\ optima := [loss, AdaM, SoftMax]\ \right)$$

CNN-2's architecture is inspired by xception [79]. It includes cast adder blocks with separable convolution layers on a depth-wise basis. Global average pooling is also used in the same manner as it is in the CNN-1 network.

Net-2: Unit definitions and instancing:

$$\overset{xcept}{\mathbb{U}} \longleftarrow \boxed{\left\langle \mathbb{C}_{2\sigma1}^{r1\$} \mid {}_{ps_d}\mathbb{C}_{3}^{br1\$}\ {}_{ps_d}\mathbb{C}_{3}^{br1\$}\ {}_{m}\mathbb{P}_{2\sigma3} \right\rangle}\ \overset{xcept}{\underset{1}{\mathbb{U}}}(16)\ \overset{xcept}{\underset{2}{\mathbb{U}}}(32)\ \overset{xcept}{\underset{3}{\mathbb{U}}}(64)\ \overset{xcept}{\underset{4}{\mathbb{U}}}(128)$$
$$\boxed{img75} \mapsto \mathscr{J}_{yx}\ {}_{p}\mathbb{C}_{3}^{br8}\ {}_{p}\mathbb{C}_{3}^{b8}\ \overset{xcept}{\underset{1}{\mathbb{U}}}\ \overset{xcept}{\underset{2}{\mathbb{U}}}\ \overset{xcept}{\underset{3}{\mathbb{U}}}\ \overset{xcept}{\underset{4}{\mathbb{U}}}\ {}_{p}\mathbb{C}_{3}^{4}\ {}_{a}\mathbb{P}_{g}$$
$$\overset{ImageEncoder}{\mathscr{NET}}\left(\ img75 := 75_{xy};\ optima := [loss, AdaM, SoftMax]\ \right)$$

CNN-2 performs similarly to CNN-1 when testing data is used. However, its design is more complex, which results in improved generalization as evaluated by the difference between training and testing data performance.

CNN-3 is composed of convolutional, max pooling, and dense layers, with the first layer being followed by the dropout layer during the training step.

$$\boxed{img150} \mapsto \mathscr{I}_{yx}\ \mathbb{C}^{32}_{br\,3}\ {}_{m}\mathbb{P}_{2}\ \mathbb{C}^{32}_{br\,3}\ {}_{m}\mathbb{P}_{2}\ \mathbb{C}^{64}_{br\,3}\ {}_{m}\mathbb{P}_{2}\ \mathbb{C}^{64}_{br\,3}\ {}_{m}\mathbb{P}_{2}\ \mathbb{F}^{64}_{r}\ \mathbb{D}_{50}\ \mathbb{F}^{4}$$
$$\overset{cnn-3}{\mathscr{NET}}\left(img150 := 150_{yx};\ optima := [loss, AdaM, SoftMax] \right)$$

The original picture training dataset is enhanced via the use of affine transformations, scaling, cropping, lighting, contrast, and Gaussian noise. The augmentation makes models more resistant to altering the head position — as shown in the test set. (Figure 4.7)

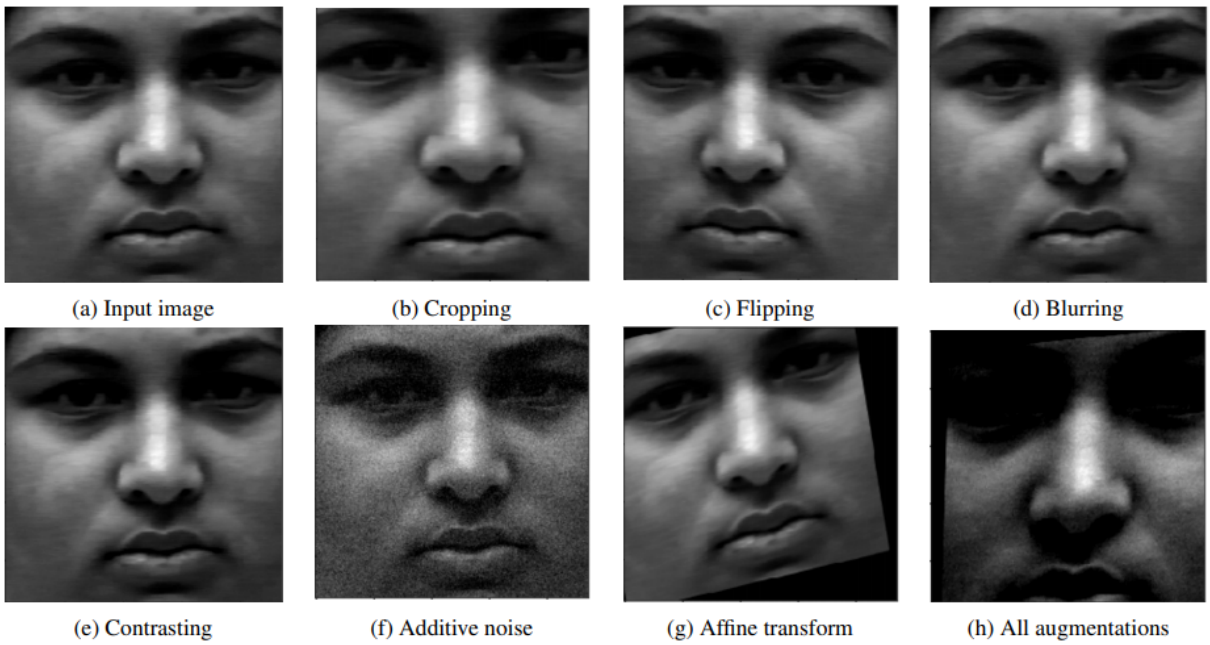


Figure 4.7. Image augmentation results from Cohn and Kanade dataset, final augmentation consists of all operations applied in random order.

The detailed parameters for the augmentation are list below:

1. Vertical axis symmetry is applied with probability 0.5;
2. Cropping randomly 0 - 10% rows and columns of the image;
3. Gaussian blur $\mathcal{N}(0, \sigma)$ is randomly applied with probability 0.5 for $\sigma \in \mu(0, 0.5)$;
4. Contrast normalization $\alpha \leftarrow \mathcal{U}(0.75, 1.5)$; $I'_i \leftarrow \max(0, \min(255, \alpha \cdot (I_i - 128) + 128))$
5. Additive Gaussian Noise $Z_i \leftarrow \mathcal{N}(0, 25)$; $I'_i \leftarrow \max(0, \min(255, I_i + Z_i))$
6. Affine transform with random matrix in the uniform pixel coordinates representing the composition of the following basic transformations:
   - scaling by $s \in \mathcal{U}(0.9, 1.1)$
   - translating $t_x \in \mathcal{U}(-x_{\text{res}}/10, x_{\text{res}}/10)$, $t_y \in \mathcal{U}\left(-y_{\text{res}}/10, y_{\text{res}}/10\right)$,

- rotating by $\theta \in \mathcal{U}(-25°, 25°)$,
- shearing by $\alpha \in \mathcal{U}(-8°, 8°)$.

Figure 4.8. Cropped faces from Cohn and Kanade dataset used for training phase after augmentation procedure.

According to 4.1 statistics, when the same discriminative features are used for both AU and FP68, DNN solutions are massively more accurate than SVM solutions. Additionally, we develop a cross-validation technique for evaluating and selecting the best SVM classifiers. We provide the standard deviation and mean of accuracy for SVM classifiers in Table 4.2. Thirty distinct experiments are analyzed statistically. We notice that the standard deviation is modest for statistics, implying that the performance of each model is comparable.

The findings of AU are about 15% more accurate than the pure geometric FP68, and the classification methods of DNN are nearly as good as the basic CNN-1, 50x50 result. DNN's findings even peak at 87.7 percent when RGB pictures are used as input, but utilizing classical features as input leads in a lesser accuracy of 75.4 percent.

	Train Data		Test Data	
Vectorized Data	AU	FP-68	AU	FP-68
SSVM	0.838	0.800	0.411	0.335
SVM (poly)	0.824	0.611	0.442	0.404
DNN*	0.830	0.642	0.754	0.611
Images				
CNN-1 50x50x1	0.838		0.763	
CNN-1 75x75x1	0.927		0.847	
CNN-2 75x75x1	0.865		0.836	
CNN-3 150x150x1	0.932		0.877	

Table 4.1. Accuracy results for selected features

According to the results of experiments on the classification performance of FER (raw images, FP68 landmarks, and action units), when dealing with each of those discriminative features individually, DNN as the classification algorithm produces the most promising results; even when classifying only the eight-dimensional data, it maintains approximate classification accuracy.

	Train Data		Test Data	
Mean of SR				
Vectorized Data	AU	FP68	AU	FP68
SVM (poly)	0.799	0.605	0.426	0.388
SSVM	0.835	0.746	0.404	0.311
Standard deviation of SR				
Vectorized Data	AU	FP68	AU	FP68
SVM (poly)	0.011	0.003	0.008	0.008
SSVM	0.002	0.038	0.004	0.027

Table 4.2. Standard deviation and mean of accuracy for SVM classifiers

Specific to this study, when models are trained on frontal pictures of human faces and then assessed on random head postures and geometric features, the success rate (accuracy) of CNN classifiers almost triples when compared to SVM classifiers under challenging conditions. CNN outperforms its geometric counterpart (AU/CNN) in terms of accuracy by about 30 percent for raw images, while the best SVM solutions outperform CNN by almost four times. The raw/CNN approach has a considerable advantage over geometric/CNN and geometric/SVM when it comes to F-score.

Also discovered is that CNN-based emotion classifiers outperform SVM-based emotion classifiers in terms of generalization to human head position when compared to CNN-based emotion classifiers.

4.3. Transfer learning from facial emotion recognition

In this subsection, we take a new approach to emotion recognition by using transfer learning from the face identification neural network solution, reported in [115]. We demonstrate how transfer learning from such solutions may aid in the initialization of the network for emotion detection, resulting in more efficient learning for our target task and allowing us to utilize more complicated networks while maintaining a higher performance. Additionally, we show that the face identification data domain is more suited to emotion recognition data domain than the emotion recognition data domain, implying that using the same architecture, one may get improved performance from the transfer learning mechanism.

4.3.1. Source task: VGG face descriptor

By choosing face identification as our source task, we adhere to the concept that the source task and target task should exhibit the greatest adaptability in the data domain. With regards to the tasks of face identification and emotion recognition, the deep features we are interested in are identical to those of human faces. In comparison to generic pre-trained networks from object categorization, it is potentially more successful in mapping source data domain knowledge to the target domain with less target samples.

To do this, we used the VGG-16 architecture-based VGG face descriptor from Parkhi, Vedaldi, and Zisserman [80], who train the VGG-16 network from scratch for face recognition

utilizing 2.6 million training data corresponding to 2622 unique people. With 138 million parameters and 15.5 billion multiply-add operations, the VGG-16 architecture is very large. The network's primary functionality is comprised of convolutional layers followed by a maximum pooling layer. Three times down sampling results in the output of 7 by 7 pixels from the original 244 by 224 three channel picture input. The number of filters, beginning with 64 at the base and up to 512 at the top. The deep features are then mapped via two thick layers with 4096 kernels each, followed by a 50% chance dropout and ultimately completely linked to the 2622 classification outputs. The VGG-16 is explained in detail below using a symbolic representation created by Professor Władysław Skarbek[81].

$$\boxed{img224} \mapsto \mathscr{I}_{yx}\ \mathbb{C}_{3}^{64}\ \mathbb{C}_{3}^{64}\ {}_{m}\mathbb{P}_{2}\ \mathbb{C}_{3}^{128}\ \mathbb{C}_{3}^{128}\ {}_{m}\mathbb{P}_{2}\ \mathbb{C}_{3}^{256}\ \mathbb{C}_{3}^{256}\ \mathbb{C}_{3}^{256}\ {}_{m}\mathbb{P}_{2}$$
$$\mathbb{C}_{3}^{512}\ \mathbb{C}_{3}^{512}\ \mathbb{C}_{3}^{512}\ {}_{m}\mathbb{P}_{2}\ \mathbb{C}_{3}^{512}\ \mathbb{C}_{3}^{512}\ \mathbb{C}_{3}^{512}\ \mathbb{F}_{r}^{4096}\ \mathbb{D}_{50}\ \mathbb{F}_{r}^{4096}\ \mathbb{D}_{50}\ \mathbb{F}^{2622}$$
$$\overset{VGG-16}{\mathscr{NET}}\left(img224 := 224_{yx};\ optima := [loss, AdaM, SoftMax] \right)$$

The notation above is defined as follows:

1. $\mathbb{C}_{3}^{64}$ denotes the convolutional layer with 64 convolutional kernels of shape 3×3. The ReLU nonlinearity is used by the inside notation r .
2. ${}_{m}\mathbb{P}_{2}$ denotes the Maxpooling layer with 2×2 kernel.
3. $\mathbb{D}_{50}$ denotes that dropout 0.5 is adopted.
4. $\mathbb{F}^{4096}$ denotes the fully connected layer with 4096 outputs.

The Face Descriptor's stated result is 98.95 percent accurate in real-world conditions. It's interesting that even when the same individual's photos are taken with and without a beard, with and without makeup, and with and without shadowing on the face, the model correctly recognizes the faces, demonstrating an extraordinary ability to extract the details of face components, which is also required for an emotion recognition model.

Figure 4.9. Image samples from VGG-Face dataset used for training Face Descriptor model.

4.3.2. Target task: Emotion Recognition

We seek to identify seven distinct facial expressions in the emotion identification challenge, including happy, sad, startled, furious, fearful, disgusting, and neutral. It is generally recognized that the critical elements for classification tasks are obtaining the most accurate characteristics defining distinct categories and separating them using a strong algorithm.

Deep neural networks have demonstrated significant benefits for end-to-end human face recognition tasks. Generally, a network's performance is proportional to the complexity of its architecture, in which case the number of training samples required grows exponentially, primarily because end-to-end solutions require enough samples to generate the filtering kernels, map the deep features, and compute the final result. We suggest emotion recognition as the goal objective for transferring face identification information in order to aid in the network's initialization and therefore to effectively assist the training process.

Apart from improving the model's initialization, this is a realistic approach if we are enthusiastic about achieving emotion detection in the wild. In comparison to the samples obtained from real-world pictures or video clips for video analysis, the samples collected from real-world images or video clips will require an enormous amount of time. If the model is more efficient in its learning, we save a great deal of time preparing the target training samples.

Transfer of knowledge The knowledge of the face identification model is obtained by fine-tuning the weights of the source model kernels and continuing to train it with target samples. In this manner, the model is modified appropriately for the final full connection layers to perform a 7 class classification task rather than 2622 as shown below.

$$\boxed{img224} \mapsto \mathscr{J}_{yx}\ \mathbb{C}_{3}^{64}\ \mathbb{C}_{3}^{64}{}_{m}\mathbb{P}_{2}\ \mathbb{C}_{3}^{128}\ \mathbb{C}_{3}^{128}{}_{m}\mathbb{P}_{2}\ \mathbb{C}_{3}^{256}\ \mathbb{C}_{3}^{256}\ \mathbb{C}_{3}^{256}{}_{m}\mathbb{P}_{2}$$
$$\mathbb{C}_{3}^{512}\ \mathbb{C}_{3}^{512}\ \mathbb{C}_{3}^{512}{}_{m}\mathbb{P}_{2}\ \mathbb{C}_{3}^{512}\ \mathbb{C}_{3}^{512}\ \mathbb{C}_{3}^{512}\ \mathbb{F}_{r}^{4096}\ \mathbb{D}_{50}\ \mathbb{F}_{r}^{4096}\ \mathbb{D}_{50}\ \mathbb{F}^{7}$$
$$\overset{VGG-16}{\mathscr{NET}}\left(img224 := 224_{yx};\ optima := [loss, AdaM, SoftMax] \right)$$

4.3.3. Experimental Dataset

We choose the FER2013 dataset for the goal task of emotion recognition. It is a publicly available collection of emotion pictures; it comprises of 48 by 48 pixel gray-scale portraits of people. It aligns the faces of all objects and labels them; in all, 28,709 samples for training, 3,589 samples for public testing, and another 3,589 samples for private testing are included in that challenge. The primary reason we chose the FER2013 dataset is because the samples themselves are more difficult, since they were gathered in real life, as opposed to some other more traditional emotion datasets obtained in the laboratory. The database is not only considerably bigger in terms of size, but also in terms of variety; the images are chosen nearly each emotion picture per person, which makes a deep neural network solution more acceptable.

In this section, we compare the FER2013 dataset to the JAFFE dataset [82], which contains just ten distinct Japanese female models performing six different emotions with no camera

angle variation. Another is the CK+ database, which has 593 sequences from 123 individuals. However, there are only 327 captioned pictures and they are all shot from a frontal camera position.

Figure 4.10. Samples from FER2013 database compared with samples from JAFFE and CK+ datasets.

According to the data from FER2013, about 30% of the training samples are polluted, meaning they are completely unrelated to human emotion yet were nonetheless tagged with emotions and included in the training process. We retained those contaminated samples (cf. Fig.4.11) throughout our trials to ensure consistency with the findings of others.

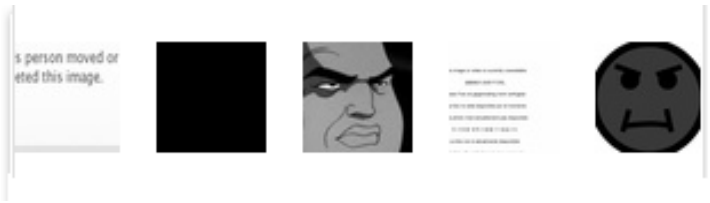


Figure 4.11. Contaminated image samples in FER2013 dataset.

4.3.4. Data preparation

Data preparation is critical for solving classification problems with deep neural networks. Although the FER2013 database already has face alignment, which saves us time by eliminating the need to locate our ROI in all samples, the database is very imbalanced. H. Paulina and M. David [83] have addressed the effect of unbalanced data on classification outcomes. In our instance, samples from the joyful class are almost 20 times as numerous as those from the disgust class, which has just 436 samples.

We firstly perform data augmentation for the categories having less data samples offline. Those samples are duplicated by performing affine transforms, scaling, vertically flipping operations, in the end the distribution of the samples are just differ less than 5%. The detailed operations are list below, examples of the duplicated samples are shown in Fig.4.13.

1. Vertical axis symmetry is applied with probability 0.5.
2. Affine transform with random matrix in the uniform pixel coordinates representing the composition of the following basic transformations:

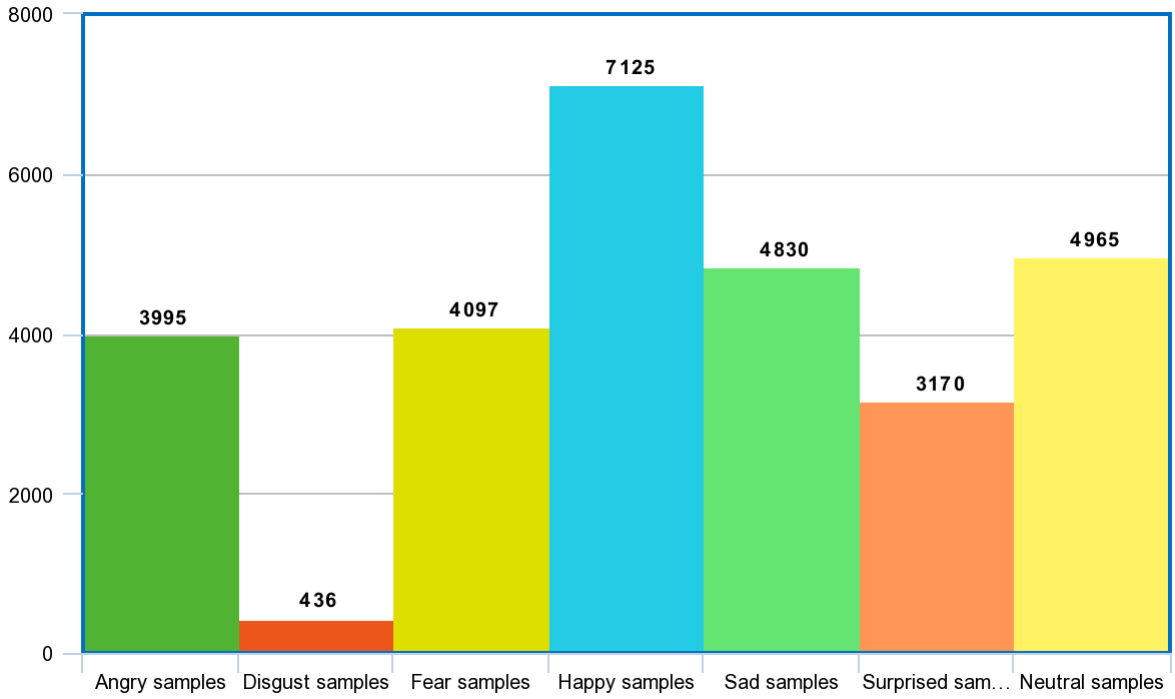


Figure 4.12. Histogram of class labels for FER2013 training dataset.

- scaling by $s \in (0.9, 1.1)$,
- translating $t_x \in (-x_{res}/10, x_{res}/10)$, $t_y \in (-y_{res}/10, y_{res}/10)$,
- rotating by $\theta \in (-25°, 25°)$,
- shearing by $\alpha \in (-8°, 8°)$.

Figure 4.13. Augmented samples from FER2013 dataset for different emotions.

4.3.5. Training strategy

The transfer learning strategy we suggest here is one that makes use of fine-tuning. We do not conduct fine-tuning on the dense layers where the deep feature mapping is taught, but on the whole model, including the convolution layers that extract the deep feature mapping. Assuming the convolution layers from the source job of face identification can extract information about detailed faces, the experimental comparison may indicate whether

we need to change our understanding of describing features, feature mapping, or both for the target goal of emotion detection.

4.3.6. Impact of learning rate

For the loss function, we used classical cross entropy and Adam for optimization. One interesting observation we made is that the learning rate of such settings has a significant effect on the performance of the prediction accuracy; when set to the classical 10^{-3} for Adam, the model does not learn at all for all of the experimental cases above; however, decreasing the learning rate by a factor of ten significantly helps the model learn. Until we set the learning rate to less than 10^{-6}, the cross-entropy loss diverges.

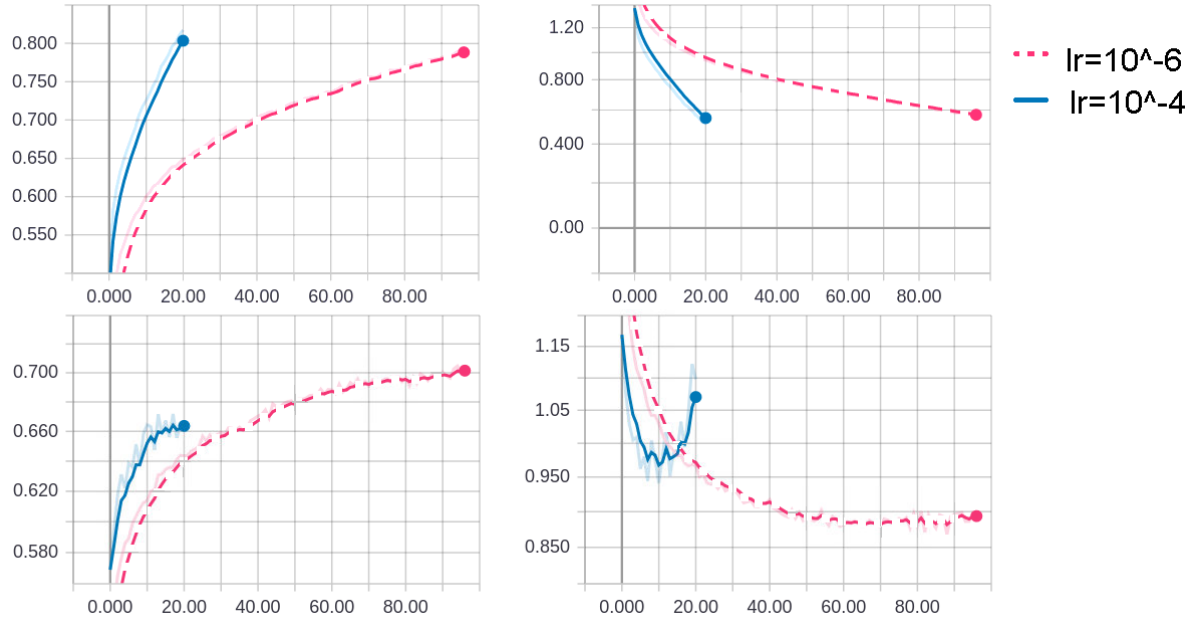


Figure 4.14. Impact of learning rate: top-left: training accuracy; top-right: training loss; bottom-left:validation accuracy; bottom-right: validation loss.

The learning rate greater than $10^{-}6$ seems to be quite high for the pre-trained source model, causing the loss to exceed even the local minimum and resulting in the divergence of the cross-entropy loss. Which is reasonable since the model from the source task has generalized well; the loss may already be at the local minimum; nevertheless, employing a low learning rate ensures that we approach the global minimum.

4.3.7. Impact of fine-tuning strategies

To investigate the effects of the "features extractor," i.e. the convolution layers filtering the input images, and the "mapping operators" of the dense layers, we use two distinct fine-tuning strategies: freezing the convolution kernels and fine-tuning the dense layers, effectively fine-tuning the entire network. If knowledge of the face recognition model is sufficient for extracting facial information, fine-tuning the "features extractor" should provide results comparable to fine-tuning the hole network. The findings in Figure 4.15 demonstrate

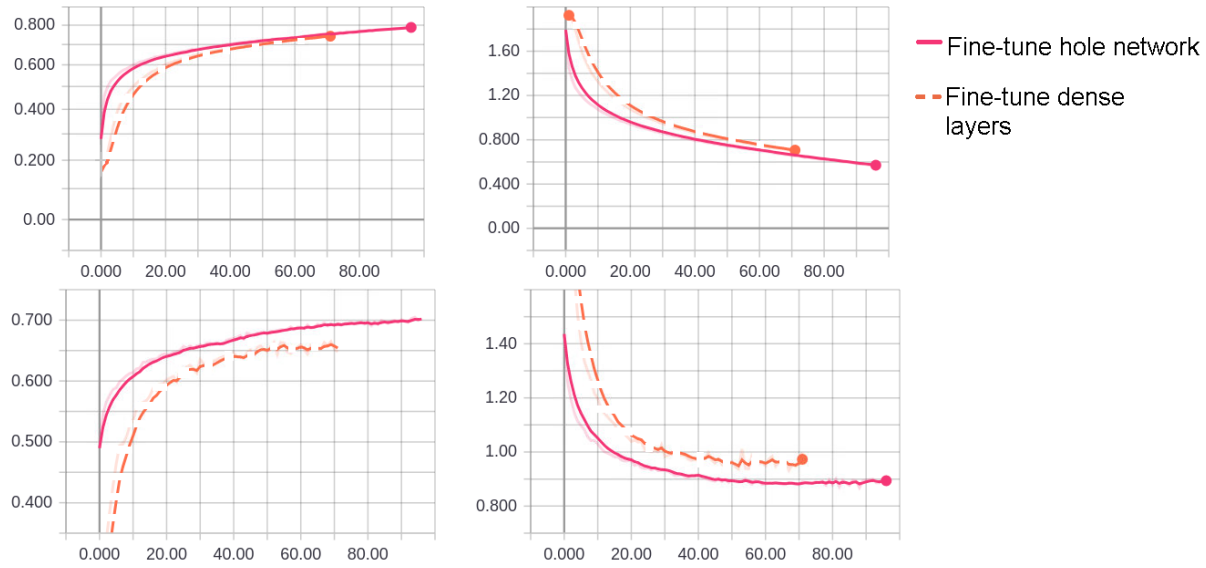


Figure 4.15. Impact of fine-tuning strategies: top-left: training accuracy; top-right: training loss; bottom-left:validation accuracy; bottom-right: validation loss.

that fine-tuning the hole network, which modifies the knowledge of the "features extractors," leads in a 5% improvement in performance. This fine-tuning approach should be used to comparable transfer learning tasks.

4.3.8. Comparable results with others

The results presented in Table 4.3 are without additional facial expression training samples, and we compare them to other solutions using general transfer learning from AlexNet, GoogleNet, and VGGNet, as well as a train-from-scratch solution with additional features or classification algorithm reported by various experts in the last two years.

	Methods	Accuracy
Ours	Transfer learning from face identification to emotion	70.47%
S. A and A.F[77]	CNN model combined with HOG features	65.0%
Y. G[84]	Transfer learning from AlexNet, GoogleNet, and VGGNet	65.0%
FER2013[85]	Best result reported from FER2013 challenge	69.76%
C. P and M. K[86]	CNN model with illumination correction pre-processing	75.2%
C. Li, N. M and Y.D[87]	Mutiple networks fusion	68.7%
Y. T[88]	Deep learning combined with SVM	69.4%

Table 4.3. Transfer learning versus other train from scratch methods.

We make no claim that our approach is state-of-the-art, but it does enable us to use more sophisticated design, and as a result, our model seems to be more accurate than models based on simpler architecture. In comparison to the same architecture of VGGNet, transfer learning from a pre-trained face recognition model is more resilient for learning the target task, and therefore our approach beats the same complicated architecture of VGG-16 by around 5% in accuracy. When compared to alternative solutions that include characteristics derived from the original samples, our outcome remains similar. The exception is C.Pramerdorfer and

M.Kampel's [86] solution, which utilizes the aforementioned classifiers by increasing their depth for each layer and supplementing the data set using the 10-crop method.

Our study demonstrates the effectiveness of transfer learning from face recognition to emotion recognition tasks. It demonstrates another method of deep network initialization for the emotion detection problem, which allows for more effective use of the training data. When considering time constraints, efficient learning enables the heavy network to be practical with minimal training data. It is even more practical when dealing with comparable human face-related tasks in real-world applications.

4.4. Emotion recognition from streaming video in neural approach

In this part, we attempt to tackle time-related visual emotion classification problems using video segments rather than a single-frame picture for training, our temporal study is reported in [116]. Emotions never exist in a static state due to the constant passage of time. Utilizing the face expression detection technology to analyze a streaming video pixel by frame seems to resolve the issue. However, our experimental findings indicate that such a solution has a flaw in that the facial component motions, lips, eyes, and so on, may cause the face emotion in a single frame to look as another facial emotion. The situation shown in Figure 4.16 is one in which a single frame classifier fails to identify facial emotion owing to the mobility of the facial components.

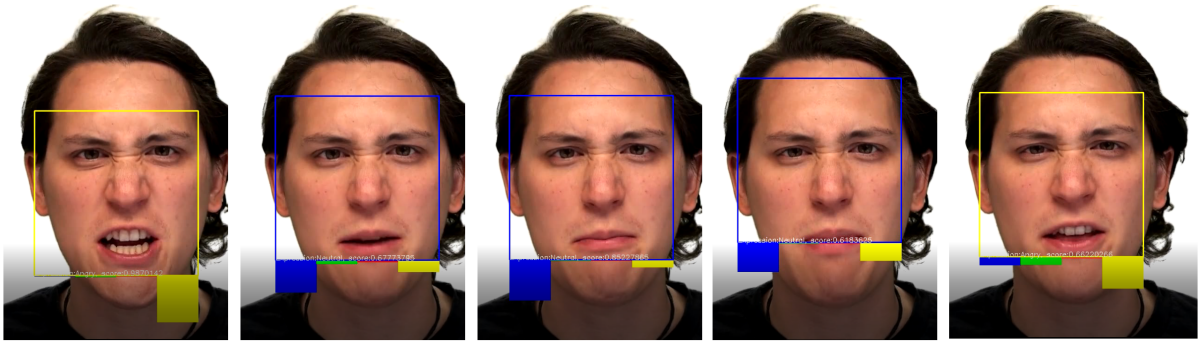


Figure 4.16. Misclassification of single frame based CNN facial emotion classifier from some frame sequences, where yellow frame stands for recognition of angry emotion and blue frame stands for the recognition of neutral emotion.

The findings indicate that FER characteristics should be spatially as well as temporally dependent. Thus, the network should be capable of remembering temporal patterns. To do this, we use CNN to extract spatial information and LSTM to extract temporal information [89]. Apart from CNN-LSTM methods, others have attempted to tackle video emotion detection problems using 3DCNN, for example, [90]. The 3DCNNs, on the other hand, need much more computing resources and data to generalize the temporal-spatial deep features. Multi-add operations may be up to 15 times faster than those required by the CNN-LSTM design. As an example, consider Resnet-18 combined with LSTM and 3D-Resnet-18:

1. Training variables and multiple adds of solution: 2D-Resnet-18 combined with LSTM for Image encoder and 2D-Resnet-18 for Audio Encoder

```
----------------------------------------------------------------------
                                 Totals
Total params                  26.813704M
Trainable params              26.813704M
Non-trainable params                 0.0
Mult-Adds                  4.200483008G
======================================================================
```

2. Training variables and multiple adds of solution:
   3D-Resnet-18 for Image encoder and 2D-Resnet-18 for Audio Encoder

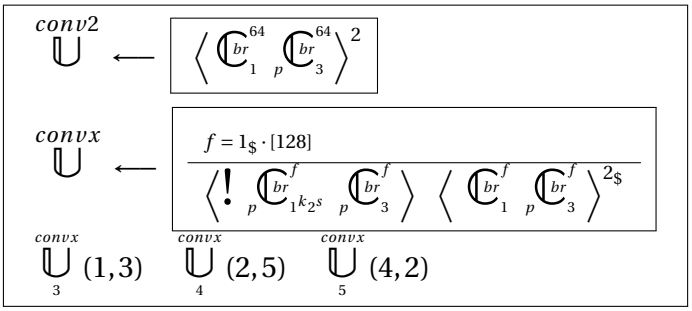
```
----------------------------------------------------------------------
                                 Totals
Total params                  44.6386M
Trainable params              44.6386M
Non-trainable params                 0.0
Mult-Adds                  56.933271744G
======================================================================
```

4.4.1. Model specification

Our approach employs the CNN-LSTM architecture, with the Resnet architecture serving as the feature extractor. Due to the spatial feature extraction from the CNN, 2DCNN makes the whole system more generalizable. Rather of focusing on temporal-spatial characteristics like 3DCNN does, our CNN concentrates only on the spatial information contained in a single frame. Thus, sequences of repeated frames, for example, when the subject maintains their posture for a few milliseconds, will generate the same deep features and will be further analyzed by the LSTM to identify any temporal deep features. While in the 3DCNN method, such brief repeated frames are handled differently, making them considerably more difficult to generalize.

The architecture is specified as follow:

Resnet Image Encoder: Unit definitions and instancing:



Resnet-LSTM: The main architecture for Image Encoder:

$$\boxed{image} \mapsto \mathscr{I}_{yx\ p}^{1}\ \mathbb{C}_{7k_2s\ m}^{br\ 64}\ \mathbb{P}_{2}\ \bigcup_{3}^{conv2}\ \bigcup_{4}^{convx}\ \bigcup_{5}^{convx}\ \bigcup^{convx}\ \mathbb{F}_{y}^{512}\ \mathbb{P}_{g\ a}\ \bigcup^{LSTM}\ \mathbb{F}^{8}$$

$$\overset{Resnet-LSTM}{\mathscr{NET}}\left(image := 224; optima := [loss, AdaM, SoftMax + CE] \right)$$

The $\mathbb{C}_{1\ m}^{br\ f}$, $\mathbb{P}_{2}$, $\mathbb{F}_{y}^{512}$ symbols stands for convolution, pooling, fully connection layers respectively. The user defined units $\bigcup^{LSTM}$ stands for the LSTM cell and $\bigcup_{3}^{convx}$ for special convolution blocks in the Resnet design.

The video data augmentation follows similar procedure of image augmentation, only the same random factors are applied to the whole frames of a file instead of to the individual frames separately.

4.4.2. Experimental results

Figure 4.17. Resnet-LSTM results for the streaming classification results, the pink frame stands for the recognition of angry expression.

By comparing Figure 4.17 and Figure 4.16, we can observe that the Resnet-LSTM solution performs much better than the pure single frame classification for streaming video classification. Although the classification results are still dominated by angry and neutral expressions, the temporal information is retained and used in the Resnet-LSTM solution, ensuring that all streaming frames are classified correctly.

4.5. Proposed methods of SER

Additionally, our method to SER makes use of CNN-extracted raw spectrogram features. While spectrogram matrices accurately represent voice information, they cannot be utilized effectively by just CNN systems. Due to the unknown duration of the audio file, the time axis of the spectrogram matrices has an indeterminate length. Not only can having a varied spectrogram resolution result in differing deep feature sizes at the output of CNN extractors, but it is also incompatible with online solutions that require constantly predicting the emotion contained inside spoken words.

4.5.1. Model specification

We chose the CNN-LSTM architecture for the SER job because to these considerations. The benefit of this combination is that it enables investigation of spectrogram segments rather than the whole matrix at once. Thus, we may establish a unit period to examine for speech occurrences; the temporal information included within this unit period will create a fixed resolution of the spectrogram and will continue to have an effect on subsequent information.

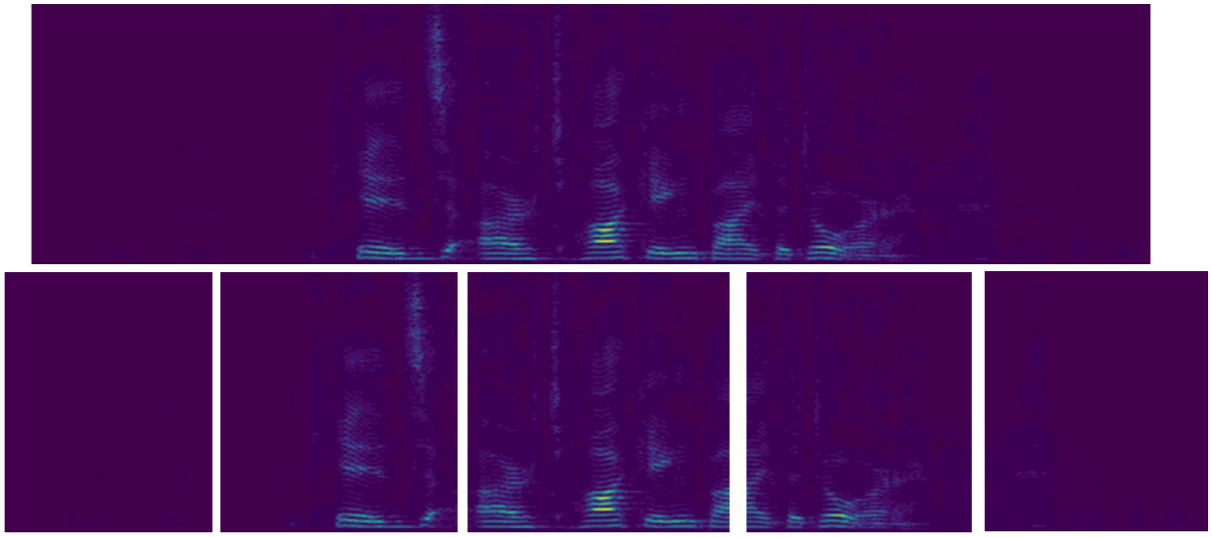


Figure 4.18. Example of segments with a fixed duration from the original spectrogram.

As with the CNN-LSTM solution for VER, we use the Resnet architecture as the CNN extractor to avoid gradient vanishing or bursting. The design is as follows:

Resnet-LSTM: The main architecture for audio deep feature extraction:

$$\boxed{\begin{array}{c} spectrogram \mapsto \mathscr{J}^{1}_{yx}\ {}_{p}\mathbb{C}^{64}_{br\,7^{k}2^{s}}\ {}_{m}\mathbb{P}_{2}\ \overset{conv2}{\cup}\ \underset{3}{\overset{convx}{\cup}}\ \underset{4}{\overset{convx}{\cup}}\ \underset{5}{\overset{convx}{\cup}}\ \mathbb{F}^{512}_{y}\ {}_{a}\mathbb{P}_{g}\ \overset{LSTM}{\cup}\ \mathbb{F}^{8} \\ \hline \overset{Resnet-LSTM}{\mathscr{NET}}\left(\begin{array}{c} spectrogram := 257_{y}250_{x}; \\ optima := [loss, AdaM, SoftMax + CE] \end{array}\right) \end{array}}$$

4.5.2. Preprocessing of audio data

Preprocessing audio data is concerned with the time-domain and spectrogram domains. Because our raw audio inputs are spectrogram frames, the conversion of the spectrogram from STFT is critical. To fine-tune the spectrogram's resolution, the window size, STFT hop size, and audio sample frequency should all be carefully chosen.

The settings should be chosen in such a way that the spectrogram information is reasonably apparent on both the time and frequency axes; otherwise, the detailed information will be obscured. All audio samples are resampled at 16kHz in our approach, a Hann window of size 512 is used, and a hop size of 64 is used. The results of the comparison of the produced spectrogram pictures are shown in Figure 4.19.

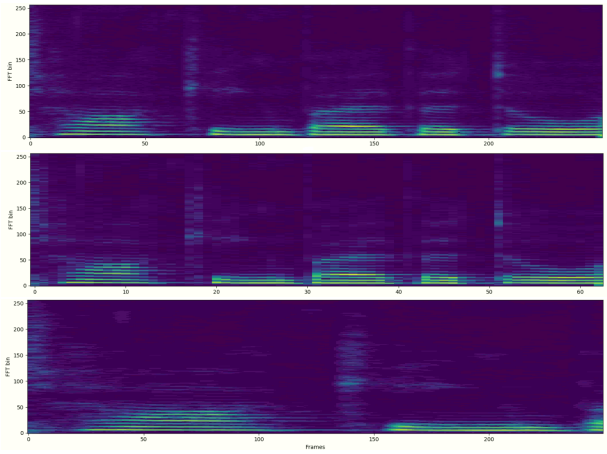


Figure 4.19. Spectrogram comparison. top: our proposed windowing; middle: having too big hopping steps at 128. button: having too small hopping steps at 32.

The STFT settings stated above provide clear spectrogram segments for the CNN extractors, while the LSTM cell processes the extracted deep features and produces the final integrated deep features. To go even deeper into the temporal information, we suggest delivering overlapping portions to the CNN.

Overlapping segments have two benefits: The jumped segments along the time axis are analogous to the shifting technique used in picture enhancement. To the CNN extractor, the jumped segments seem to be a fresh sample, which aids with generalization. On the other hand, the LSTM gets additional deep features from the same source file that has not been changed. Thus, the LSTM cell is likewise more generic as a result of the time augmentation.

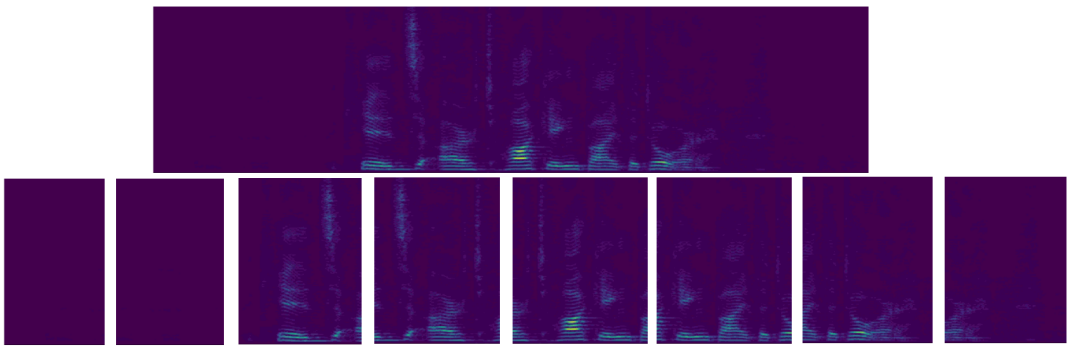


Figure 4.20. Overlapped spectrogram segments generated from the original spectrogram sample.

5. Multi-modal Residual Perceptron Network for AVER

This section discusses our contribution to AVER issues related to current shortcomings in commonly used architectures with naïve fusion strategies for multi-modal information. The contribution is pulished in [117].

5.1. Hypothesis

In this subsection, we discuss our hypothesis where fuzzy information from the uni modalities can cause chaos in not just the uni-modal neurons but also the correlation neurons, namely the fusion component.

5.1.1. Potential failures in the existing solutions

Consider the human brain's learning process once again. Assume that part of the sensory input contains incorrect information. A youngster learnt an animal that resembled a dog but had the voice of a cat through edited films, yet this child had never encountered a dog or a cat in a natural setting. He will either see a dog and misidentify it as a cat, or he will hear a cat and misidentify it as a dog. The issue may deteriorate further if the stimulus from which he learnt is likewise ambiguous within their own experience.

His identification ability is still intact to a degree, as he may sometimes properly identify visual or auditory information patterns. However, the recognized data is skewed, as is the correlation of the intermodal data. As a result, the warped uni-modal information he acquired had a detrimental effect on the other. We apply the same approach to existing multi-modal neural network systems. The learnt pattern's within-modal and inter-modal noisiness both contribute to incorrect recognition.

Despite the numerous benefits of multi-modal solutions in terms of improving recognition performance on emotion recognition tasks, we hypothesize that the uncontrolled fusion strategy used by [63, 71, 91, 92, 93, 68, 64, 66] may result in potential deficiencies in either late fusion or end-to-end fusion.

Though numerous studies have demonstrated the superior performance of late fusion strategies [94, 95, 96], for example, for audio event detection in video material, W. Wang et al. [97] demonstrate that the results of naive fusion using multi-modal features can be worse than the best uni-modal approach. They suggest combining gradient flows using multi-task loss functions, which we refer to as multi-term loss functions, from uni- and multi-modalities, which enables more accurate modeling of the whole system in a variety of different study fields. While they indicate advantages from mixing the gradient flows, multitasking may make it difficult to optimize the features for both uni- and multi-modal objectives, as many studies have proposed [98, 99, 100]. We show how, although this approach may still fail in certain inferior situations, it is resolved by the MRPN within-modal RP component.

5.1.2. Within-modal information can be missing or fuzzy

The missing or fuzziness of information may be detected in either the visual or auditory modality emotion recognition solutions, and the success rate of recognition cannot be improved much as a result of this. In the uni-modality, missing information refers to feature data in which emotion categories are confused with neutral categories, resulting in missing information. Fuzzy information refers to feature data in which one emotion category cannot be differentiated from another in a uni-modality setting.

For example, W. Wang et al. [101] found that the visual modality results from the challenge FER-2013 [85] for single picture face recognition had only increased by approximately 4 percent to 76.8 percent over the last eight years.

Furthermore, utilizing transfer learning and averaged temporal deep features, HW. Ng et al. [102] obtained 47.3 percent validation accuracy and 53.8 percent testing accuracy on the EmotiW dataset [103] using video frames.

Similarly, for vocal solutions, the results for Interactive Emotional Dyadic Motion Capture dataset (IEMOCAP) [104] with the raw inputs are reported around 76% by S. Kwon [105] and 64.93% by S. Latif et al. [106]. These situations are not recognized at a high enough rate to be considered optimum.

We are concerned with the human voting for those datasets, which is separate and distinct from the design, functioning, and training of the neural network. The instructor in supervised learning has inadequate expertise in nearly all datasets linked to emotion identification, according to the data. Human beings, who are the greatest judges of human emotions, are unable to reach a consensus on the author's labeling because they lack a majority of agreement. In general, the human rate of emotional categories is 72 percent, according to IEMOCAP, and the human accuracy on the FER-2013 [85] is 65 percent. Crema-d [107] has an accuracy of 63.6 percent, and RAVDESS [108] has an accuracy of 72.3 percent.

In every single study, it is said that the information of data in a uni-modality is never crystal clear, and as a result, the acquired knowledge of a uni-modality in emotion detection may be corrupted and uncontrolled by the network. Because the borders of the clusters are very subjective, we are unable to determine or agree on which samples are incorrect in any way.

5.1.3. End-to-end modeling for multi-modal data can be distorted

Using several modes in multi-modal solutions creates more broad patterns by expanding parameters. Despite this, there are negative side effects caused by characteristics being mixed up, which are camouflaged by its advantages.

The diagram shown in Figure 5.1 demonstrates the difference in clusterings resulting from various uni-modal and multi-modal solutions for the shuffled train/validation data. There are significant differences between the actors in the validation set and the training set. Even if the data are from the same uni-modality, the clustering of data varies due to missing and fuzzy information. As the image demonstrates, the neutral cluster is formed due

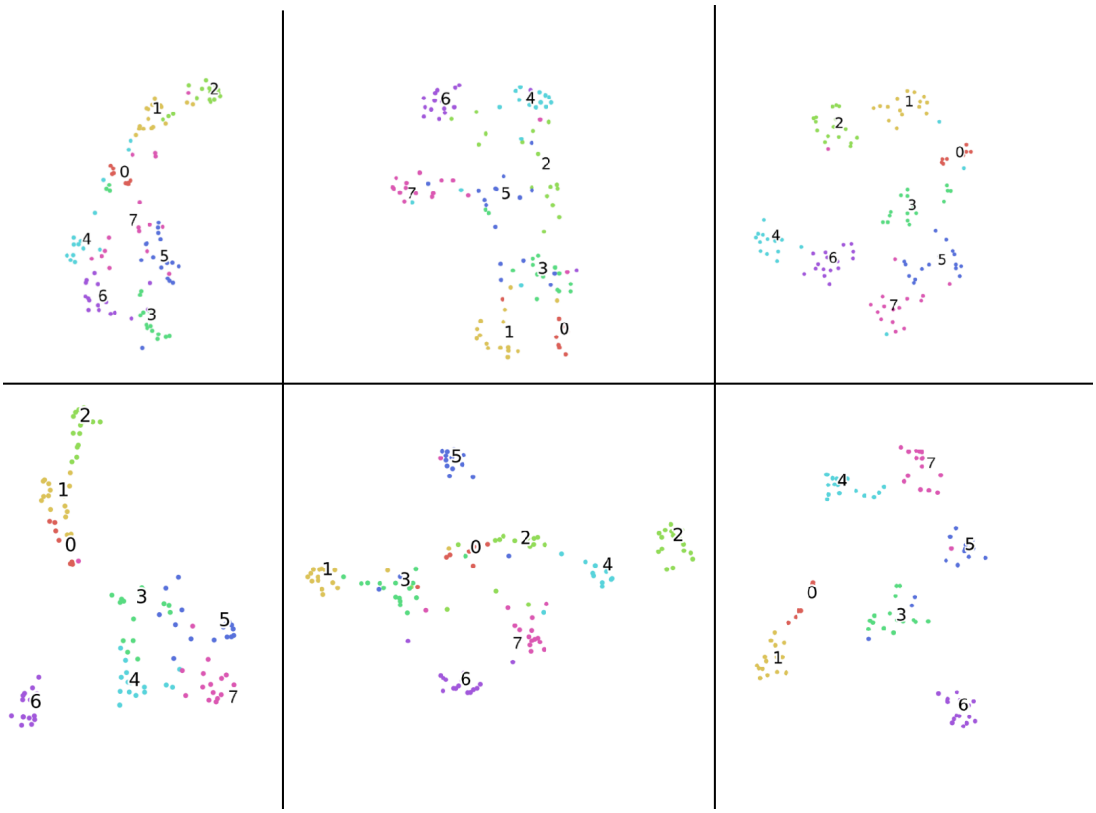


Figure 5.1. Visualization (t-SNE algorithm) of deep features clustering from two different setups where the train/validation sets are shuffled. The clustering with respect to emotion classes are listed. 0: *neutral*, 1: *calm*, 2: *happy*, 3: *sad*, 4: *angry*, 5: *fearful*, 6: *disgust*, 7: *surprised*. **Top part**: clustering results from one setup of uni-modalities and multi-modality. *Left part*: only image modality. *Middle part*: only audio modality. *Right part*: multi-modality. **Down part**: clustering results from the other setup where train/validation sets are shuffled.

to missing information for the same mode solutions, whether they uni-modal or multi-modal. The situations seem to be exactly the same for the fuzzy information overlap in emotional categories. This gives credence to the fact that patterns within uni-modality are difficult to generalize, a finding that aligns with the human voting findings.

When that happens, we aren't sure which training sample is fuzzy in which modality, meaning it is unclear not just whether it will cause within-modal learning to be fuzzy, but also if it will cause inter-modal learning to be fuzzy during end-to-end training. For example, learning is crystal clear in modality B and fuzzy in modality A. The distribution of incorrect information about directions is unclear.

The blue box in Figure 5.2 shows that a fusion component lacks in the design during gradient backpropagation. The concatenation unit of features from various modalities may iteratively change the weights of individual modalities to backpropagate gradients. The result is that certain modalities can experience distortion.

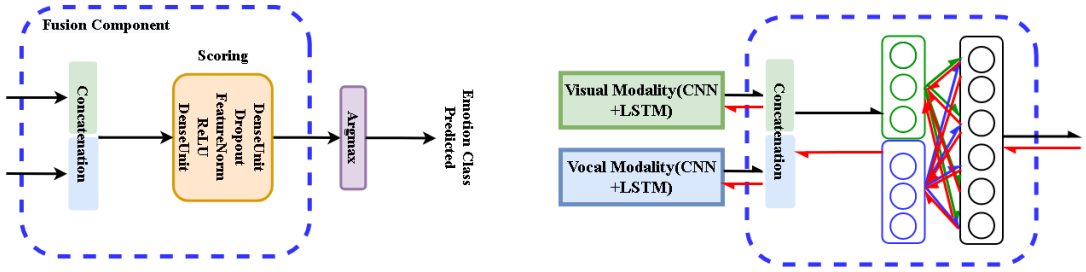


Figure 5.2. Distorted gradients backpropagation in some modality since the gradients from fused layer makes impact on gradients flow into neural weights of both modalities.

5.1.4. Late fusion modeling for multi-modal data can be insufficient

Although late fusion appears to prevent the system from learning intermodal information from the joint gradient flow, if the samples contain clean information in all modalities, the frozen parameters of the shadow layers are unable to make the necessary adjustments to learn intermodal information from the joint gradient flow.

5.2. Proposed methods

To address these problems, we developed a new MRPN coupled with a multi-term loss function for improved network parameterization that takes benefit of both late fusion and end-to-end methods while avoiding their drawbacks. MRPN can resolve these issues regardless of whether the data is noisy or clean.

5.2.1. Functional description of analyzed networks

The functional descriptions of the analyzed deep networks are presented for their training mode (see Figure 5.4). They are based on the selected functionalities of neural units and components. We use index m for inputs of any modality. In our experiments $m = v$ or $m = a$.

1. F_m: feature extractor for input temporal sequence x_m of modality m, e.g. F_v for video frames x_v, F_a for audio segments x_a.
2. A_m: aggregation component SAC for temporal feature sequence leading to temporal feature vector f_m, eg. A_v, A_a for video and audio features, respectively.

$$f_m \doteq A_m(F_m(x_m)) \longrightarrow f_v \doteq A_v(F_v(x_v)),\ f_a \doteq A_a(F_a(x_a)) \tag{24}$$

3. Standard computing units: `DenseUnit` – affine (a.k.a. dense, full connection), `Dropout` – random elements dropping for model regularizing, `FeatureNorm` – normalization for learned data regularizing (batch norm is adopted in the current implementation), and `Concatenate` – joining feature maps, `ReLU`, `Sigmoid` – activation units.
4. `Scoring` – component mapping feature vectors to class scores vector, usually composing the following operations:

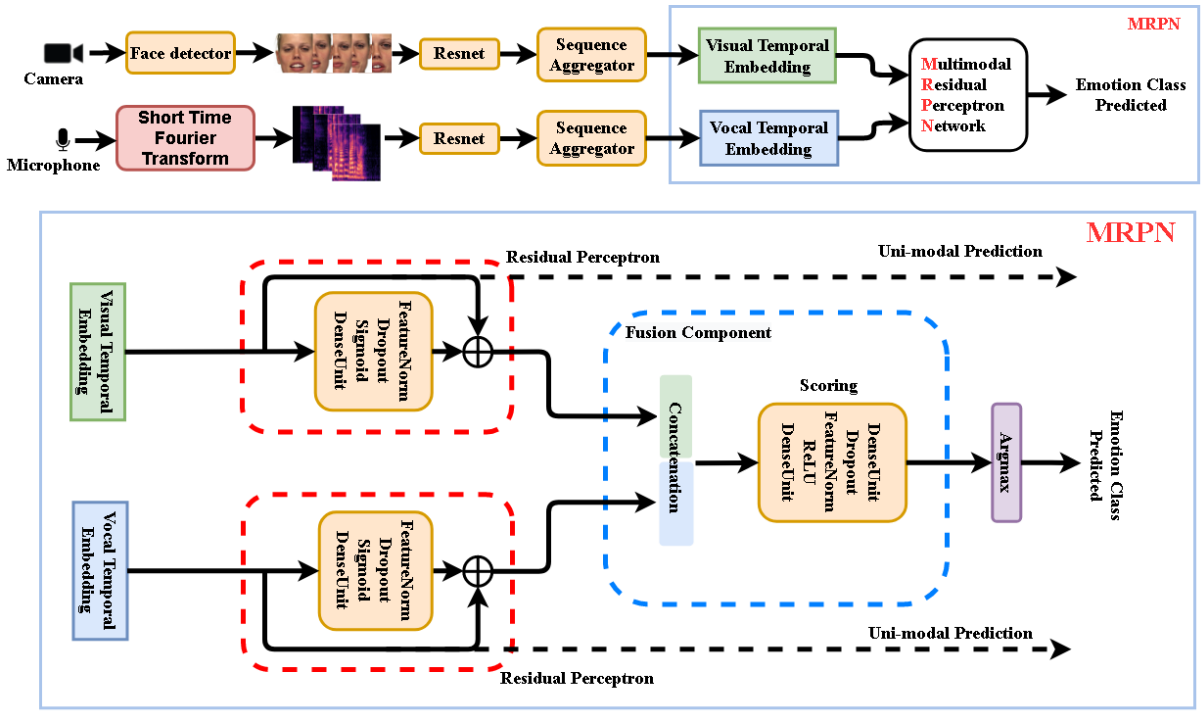


Figure 5.3. The proposed multi-modal emotion recognition system using DNN approach. **Upper part:** Video frames and audio spectral segments get independent temporal embeddings to be fused by our multi-modal Residual Perceptron Network (MRPN). **Lower part:** MRPN performs in each modality normalizations via the proposed Residual Perceptrons and then scores their concatenated outputs in the Fusion Component. The uni-modal prediction branches are only active in training mode.

$$\rightarrow DenseUnit \rightarrow ReLU \rightarrow FeatureNorm \rightarrow DenseUnit \tag{25}$$

$$\begin{aligned} &m \in \{v,a\},\ \hat{f}_m \doteq FeatureNorm(f_m) \longrightarrow s_m \doteq Scoring(\hat{f}_m) \\ &g_{va} \doteq FeatureNorm(Concatenate(g_v, g_a)) \longrightarrow s_{va} \doteq Scoring(g_{va}) \end{aligned} \tag{26}$$

5. `FusionComponent` – concatenates its inputs g_v, g_a, then makes the statistical normalization, and finally produces the vector of class scores:

$$\begin{aligned} &s_{va} \doteq FusionComponent(g_v, g_a) \longrightarrow \\ &g_v, g_a \rightarrow Concatenate \rightarrow Scoring \rightarrow s_{va} \end{aligned} \tag{27}$$

   In our networks g_v, g_a are statistically normalized multi-modal features $(\hat{f}_v, \hat{f}_a)$ or their residually updated form (f'_v, f'_a) – cf. those symbols in Figure 5.4.
6. `SoftMax` – computing unit for normalization of class scores to class probabilities:

$$\begin{aligned} &m \in \{v,a\} \longrightarrow p_m \doteq Softmax(s_m) \\ &p_{va} \doteq Softmax(s_{va}) \end{aligned}$$

7. `CrossEntropy` – a divergence of probability distributions used as loss function. Let p is the target probability distribution. Then the following loss functions are defined:

$$\begin{aligned} &m \in \{v,a\},\ p_m \doteq Softmax(s_m) \longrightarrow \mathscr{L}_m \doteq CrossEntropy(p,p_m) \\ &p_{va} \doteq Softmax(s_{va}) \longrightarrow \mathscr{L}_{va} = CrossEntropy(p,p_{va}) \\ &\mathscr{L} \doteq \mathscr{L}_v + \mathscr{L}_a + \mathscr{L}_{va} \end{aligned} \tag{28}$$

where $\mathscr{L}$ is *multi-term loss function* implying the gradient blending in the backpropagation stage.

8. `ResPerceptron` (Residual Perceptron) – component performing statistical normalization for the dense unit (perceptron) computing residuals for normalized data. In our solution it transforms a modal feature vector f_m into f'_m, as follows:

$$\begin{aligned} &\hat{f}_m \doteq FeatureNorm(f_m) \longrightarrow f'_m \doteq ResPerceptron(\hat{f}_m) \longrightarrow \\ &f'_m \doteq \hat{f}_m + FeatureNorm(Sigmoid(DenseUnit(\hat{f}_m))) \end{aligned} \tag{29}$$

Three networks $\mathscr{N}_0, \mathscr{N}_1, \mathscr{N}_2$ are defined for further analysis:

1. Network $\mathscr{N}_0(f_v, f_a; p)$ with fusion component and loss function $\mathscr{L}_{va}$:

$$\begin{aligned} &\hat{f}_v \doteq FeatureNorm(f_v),\ \hat{f}_a \doteq FeatureNorm(f_a) \\ &s_{va} \doteq FusionComponent(\hat{f}_v, \hat{f}_a) \\ &p_{va} \doteq SoftMax(s_{va}) \longrightarrow \mathscr{L}_{va} \doteq CrossEntropy(p,p_{va}) \end{aligned} \tag{30}$$

2. Network $\mathscr{N}_1(f_v, f_a; p)$ with fusion component and fused loss function $\mathscr{L} \doteq \mathscr{L}_v + \mathscr{L}_a + \mathscr{L}_{va}$:

$$\begin{aligned} &\hat{f}_v \doteq FeatureNorm(f_v),\ \hat{f}_a \doteq FeatureNorm(f_a) \\ &s_v \doteq DenseUnit(\hat{f}_v),\ s_a \doteq DenseUnit(\hat{f}_a),\ s_{va} \doteq FusionComponent(\hat{f}_v, \hat{f}_a) \\ &p_v \doteq SoftMax(s_v) \longrightarrow \mathscr{L}_v \doteq CrossEntropy(p,p_v) \\ &p_a \doteq SoftMax(s_a) \longrightarrow \mathscr{L}_a \doteq CrossEntropy(p,p_a) \\ &p_{va} \doteq SoftMax(s_{va}) \longrightarrow \mathscr{L}_{va} \doteq CrossEntropy(p,p_{va}) \end{aligned} \tag{31}$$

3. Network $\mathscr{N}_2(f_v, f_a; p)$ with normalized residual perceptron, fusion component and fused loss function $\mathscr{L} \doteq \mathscr{L}_v + \mathscr{L}_a + \mathscr{L}_{va}$:

$$\begin{aligned} &\hat{f}_v \doteq FeatureNorm(f_v),\ \hat{f}_a \doteq FeatureNorm(f_a) \\ &f'_v \doteq ResPerceptron(\hat{f}_v),\ f'_a \doteq ResPerceptron(\hat{f}_a) \\ &s_v \doteq DenseUnit(\hat{f}_v),\ s_a \doteq DenseUnit(\hat{f}_a),\ s_{va} \doteq FusionComponent(f'_v, f'_a) \\ &p_v \doteq SoftMax(s_v) \longrightarrow \mathscr{L}_v \doteq CrossEntropy(p,p_v) \\ &p_a \doteq SoftMax(s_a) \longrightarrow \mathscr{L}_a \doteq CrossEntropy(p,p_a) \\ &p_{va} \doteq SoftMax(s_{va}) \longrightarrow \mathscr{L}_{va} \doteq CrossEntropy(p,p_{va}) \end{aligned} \tag{32}$$

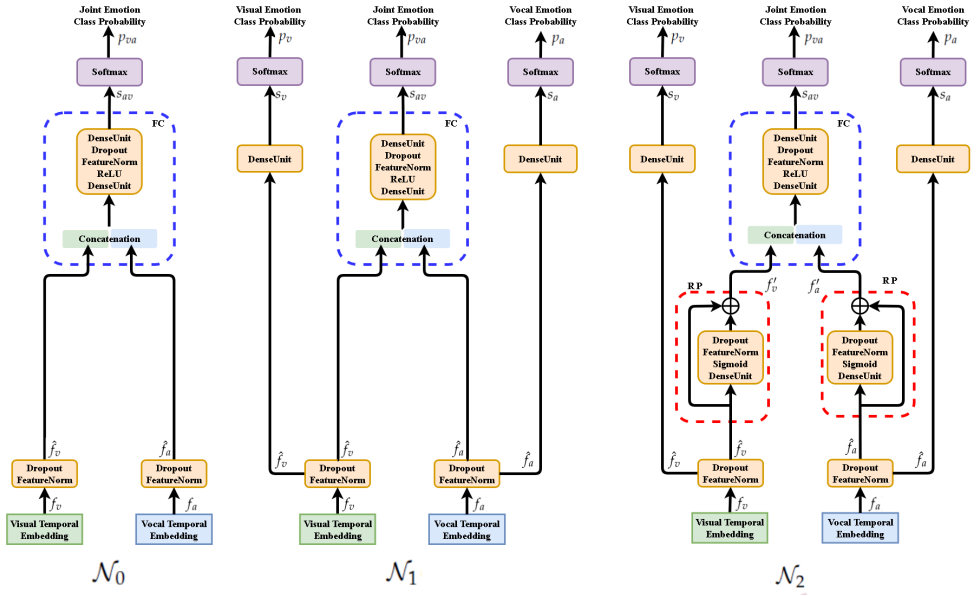


Figure 5.4. Evolution of network design for multi-modal fusion (presented for training mode). $\mathcal{N}_0$: Fusion component (FC) only. $\mathcal{N}_1$ ([97]): Beside FC, independent scoring of each modality is considered. $\mathcal{N}_2$: Extending $\mathcal{N}_1$ network by Residual Perceptrons (RP) in each modality branch.

For the networks $\mathcal{N}_0, \mathcal{N}_1, \mathcal{N}_2$ detailed in Figure 5.4, we can observe:

1. All instances of *FeatureNorm* unit are implemented as batch normalization units.
2. In testing mode only the central branch of networks $\mathcal{N}_1, \mathcal{N}_2$ are active while the side branches are inactive as they are used only to compute the extra terms of the extended loss function.
3. The above facts make network architectures $\mathcal{N}_0, \mathcal{N}_1$ equivalent in the testing mode. However, the models trained for those architectures are not the same, as weights are optimized for different loss functions.
4. In the testing mode all *Dropout* units are not active, as well.
5. The architecture of *FusionComponent* is identical for all three networks. The difference between models of $\mathcal{N}_0$ and $\mathcal{N}_1$ networks follows from the different loss functions while the difference between models of $\mathcal{N}_1$ and $\mathcal{N}_2$ networks is implied by using *ResPerceptron (RP)* components in $\mathcal{N}_2$ network.
6. To control the range of affine combinations computed by *Residual Perceptron (RP)* component, we use *Sigmoid* activations instead of the *ReLU* activations exploited in other components. The experiments confirm the advantage of this design decision.
7. The *Residual Perceptron (RP)* was introduced in the network $\mathcal{N}_2$ to implement better parameterization of within-modal features before their fusion.

5.2.2. MRPN components' role in multi-term optimization

1. Since stated in the hypothesis section, the late fusion approach has the benefit of retaining the best information in each uni-modality, as each uni-modality obtains generic deep features that are mostly unaffected by their own modality's outliers. i.e., in uni-modal solutions, a tiny quantity of incorrectly labeled input does not contribute to the general-

ized feature patterns since it is filtered out by the uni-modal neural network. Thus, the extra component in the loss functions denotes the blended gradient in each uni-shallow modality's layers, which aided in better parameterization of the features prior to fusion, while also retaining knowledge while uni-modalities are trained separately. As a result of the above, the end-to-end approach suffers from less intermodal information than late fusion.

2. However, the multi-term optimization can result in extracting inferior uni-modal features as the input to the fusion component. This problem was mentioned in the literature [98, 99, 100]. RP is introduced to make modified uni-modal features, instead of storing all information for uni-modal and multi-modal purposes in a single unit, resulting in a collision of loss converging from two directions, RP is used to create modified uni-modal features and modified multi-modal features, thus establishing a new route for the gradient flow. RP can retain the greatest characteristics of the uni-modal solution while yet allowing for the integration of additional multi-modal capabilities through the changed features from the short-cut.

The mentioned two novel properties make MRPN free from side-effects of late fusion and end-to-end strategy while preserving their own advantages.

5.2.3. MRPN in general multi-modal applications

As shown in Figure 5.5, MRPN may be used in any multi-modal application involving a large number of multi-modal inputs and a single target function, or a large number of multi-modal inputs and a large number of multi-modal target functions. In both instances, MRPN benefits from a large number of terms of loss functions proportional to the number of uni-modalities, updating the whole system simultaneously and avoiding learning from inter-modal fuzzy information. MRPN is sufficiently broad to be compatible with any other method suggested.

5.3. Computational Experiments and their Discussion

This subsection discusses the benefits of our proposed framework and time-dependent augmentation. For this aim, two datasets, RAVDESS and Crema-d, are used. The time augmentation method is improved using the naive fusion model, which produced SOTA results even without the use of MRPNs. The comparison identifies and discusses the inferior instances in such typical neural multi-modal solutions. The improvement of MRPN is then shown, not only in the inferior sub-datasets identified, but also in general data samples.

5.3.1. Datasets

RAVDESS and Crema-d vary in terms of the amount of expression categories, the total number of files, the number of identifiers, and also in terms of video quality.

1. The RAVDESS collection contains both speech and music files. We utilize just the voice files from the dataset for the speech recognition proposal. It includes 2880 files and

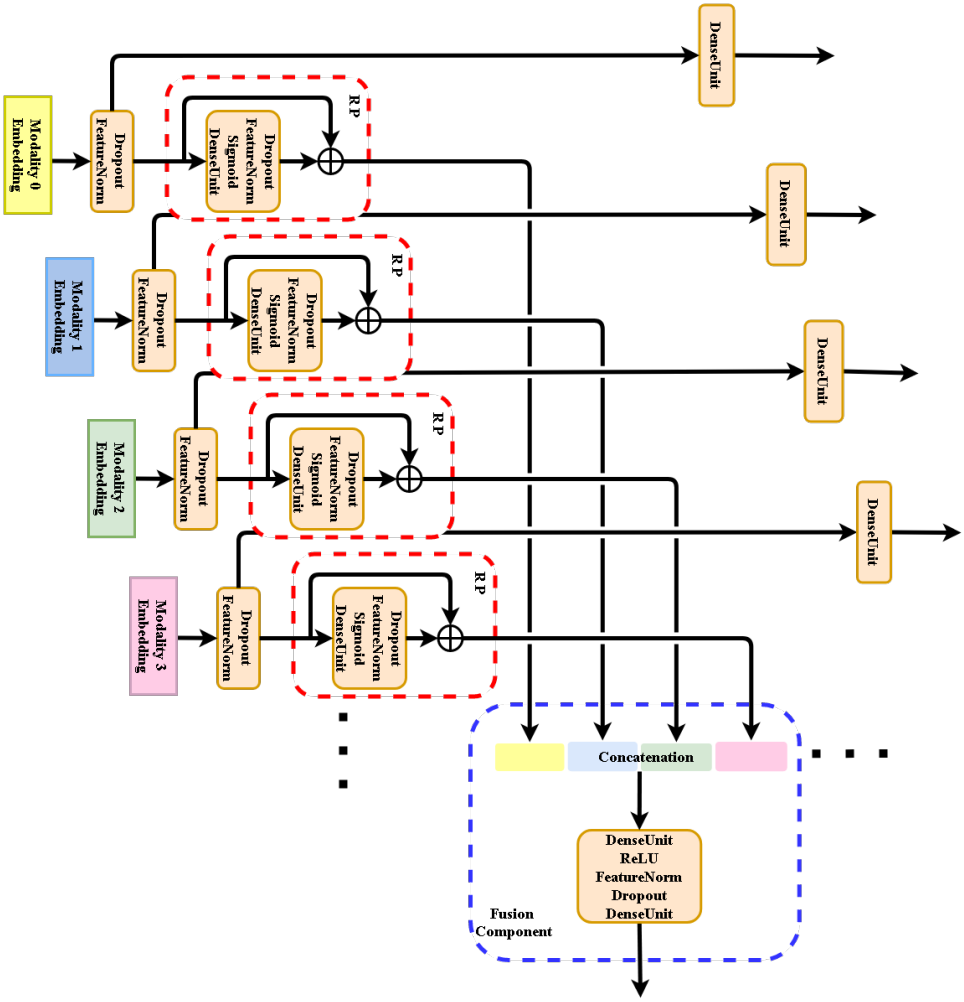


Figure 5.5. Generalization of our MRPN fusion approach to many modalities. It could be used for either regression or classification applications.

24 actors (12 females and 12 males), each of whom makes two lexically similar utterances. The expressions used in speech include calm, pleased, sad, furious, frightened, surprised, and disgusted. Each expression is generated in two emotional intensity levels (normal and strong), as well as a neutral expression, for a total of eight categories. To our knowledge, this is the most current video-audio emotional dataset with the highest video quality in this study field.

2. Crema-d is a collection of visual and audio emotional speech files expressing a variety of fundamental emotional states (happy, sad, anger, fear, disgust, and neutral). Multiple raters evaluated 7,442 footage of 91 performers from different ethnic origins in three modalities: audio, visual, and audio-visual.

The training and testing sets for both datasets are split using similar ideas to 10-fold inter-validation. Additionally, the identities of the actors are segregated in the train and val sets to avoid findings being influenced by the actors. Around 10% of the actors are utilized

for validation, while the other 90% are used for training; each set has an equal number of male and female actors. To get various findings over the whole dataset, we rotate the divided train/validation sub-datasets.

Although the crema-d dataset has fewer categories for classification tasks, according to the authors' study, Crema-d has a human recognition accuracy of 63.6 percent for six categories, which is less than RAVDESS's 72.3 percent for eight categories. The resolution of the video source is confirmed to be unrelated to the performance degradation. The RAVDESS dataset's superior findings, in our view, are due to the inclusion of more crystal clear and genuine emotion information.

5.3.2. Model organization and computational setup

The naive fusion model $\mathcal{N}_0$, advanced fusion network $\mathcal{N}_1$, which is equivalent to the Facebook [97] solution and the $\mathcal{N}_2$ (MRPN) have the same CNN extractors at the initial stage of the training. To compare the impact of strategy from features fusion only, CNN extractor architecture is fixed to Resnet-18 [109].

The CNN in visual modality is initialized from a facial image expression recognition task, the challenge FER2013 [85]. As for vocal modality, The CNN is pretrained on the voice recognition task from VoxCeleb dataset [110]. The initialization of the CNN extractors made the whole system much easier to be optimized.

AdaMW optimizer is adopted for the model optimization, with the initial learning rate at $5 \cdot 10^{-5}$, decreased two times if validation loss is not dropping over ten epochs.

5.3.3. Data augmentation cannot generalize multi-modal feature patterns

This subsection illustrates the improvement of time-dependent augmentation. The improvement also proves that the inferior case of multi-modal solution doesn't depend on the with-modal patterns. The single modality solutions in our experiments (shown in Table 5.1) take pretrained Resnet-18 as extractors and LSTM cells as SACs. The naive multi-modal solution takes twice of the components with an additional fusion layer as Figure 5.4 illustrates on the left panel. Adopting time-dependent augmentation shows overall performance improvements on either single or multi-modal solutions.

The Table notations are presented in the follows:

In the variational train/val sub-datasets in Table 5.1, Ax,y stands for the validation files that came from actor x and y, odd number notes for a male actor, and even number for a female actor.

5.3.4. Discussion on inferior multi-modal cases

While time augmentation improves the overall performance of either the uni- or multi-modal method, the inferior situation in which the uni-modal solution is superior than the multi-modal solution persists, indicating that data augmentation cannot generalize multi-modal features. In Table 5.1 of the instance A9,10, only one inferior case is found.

Table 5.1. Comparison of single modalities models with $\mathcal{N}_0$ model (RAVDESS cases): VM – Visual Modality only, AM – Audio Modality only, JM – Joint Modalities ($\mathcal{N}_0$ model), T – having time augmentation by signal random slicing, NT – not having time augmentation.

RAVDESS	A1,2	A3,4	A5,6	A7,8	A9,10	A11,12
AM (NT)	70.8%	55.0%	57.5%	74.1%	43.5%	65.8%
AM (T)	71.6%	77.5%	71.6%	90.0%	55.8%	69.1%
VM (NT)	82.5%	70.0%	66.7%	74.1%	**80.3**%	63.3%
VM (T)	86.6%	75.0%	70.6%	76.6%	**87.3%**	69.1%
JM (NT)	90.8%	89.1%	85.2%	89.3%	**78.5**%	85.5%
JM (T)	97.5%	90.3%	87.5%	97.5%	**86.5**%	87.5%
RAVDESS	A13,14	A15,16	A17,18	A19,20	A21,22	A23,24
AM (NT)	59.8%	57.5%	51.6%	55.5%	55.8%	63.3%
AM (T)	70.0%	69.1%	57.5%	63.3%	68.3%	68.3%
VM (NT)	71.3%	60.0%	63.3%	70.8%	65.8%	70.8%
VM (T)	73.3%	65.0%	64.1%	78.3%	66.6%	74.1%
JM (NT)	77.5%	75.5%	76.3%	85.2%	82.8%	80.0%
JM (T)	82.4%	79.6%	83.2%	89.0%	85.5%	84.2%

However, we suggest that this shortcoming is widespread in fuzzy multi-modal data. Both modalities have an adequate capacity for pattern learning; both solutions perform better than 85 percent in situations such as A7,8 and A1,2. However, the ratio of mismatched learnt and target patterns varies when the sub-datasets are shuffled.

The performance degradation became apparent only when the proportion of pattern mismatched samples exceeded a predefined threshold in the training set. If this is the case, then removing or decreasing such side effects should result in overall improvements for any train and testing sub-dataset.

5.3.5. Improvement of MRPN

This subsection addresses the improvement of MRPN preventing the side-effects in the existing late fusion and end-to-end strategies we hypothesized as Table 5.2 and Table 5.3 illustrate.

The end-to-end strategy of $\mathcal{N}_1$, which takes multi-term loss function helped the better parameterization shows improved average performance over naive end-to-end and late fusion training strategies, yet it can still fail in some cases. Our proposed MRPN on the contrary demonstrates the same performance or most improvement in any circumstance.

It can been seen from the confusion matrices in Figure 5.6 and 5.7 the averaged improvements of $\mathcal{N}_2$ (MRPN) over the late fusion and end-to-end $\mathcal{N}_0$ models. Performance on some specific categories shows a slight decrease for MRPN, especially for the categories of calm and neutral expressions because they are naturally close to each other in the RAVDESS dataset. $\mathcal{N}_1$ doesn't always perform better than the existing solutions, the almost 6% improvements of

Table 5.2. Comparison for RAVDESS of MRPN approach (network $\mathcal{N}_2$) with late fusion strategy ($\mathcal{N}_0$), end-to-end strategy ($\mathcal{N}_0$), and advanced end-to-end fusion strategy ($\mathcal{N}_1$).

RAVDESS	A1,2	A3,4	A5,6	A7,8	A9,10	A11,12
$\mathcal{N}_0$(late fusion)	61.6%	92.1%	87.5%	96.6%	66.6%	87.5%
$\mathcal{N}_0$(end-to-end)	**97.5%**	90.3%	87.5%	**97.5%**	86.5%	87.5%
$\mathcal{N}_1$(end-to-end)	**97.5%**	89.1%	88.3%	**97.5%**	90.0%	**90.0%**
$\mathcal{N}_2$(end-to-end)	**97.5%**	**92.1%**	**90.8%**	**97.5%**	**91.4%**	**90.0%**
RAVDESS	A13,14	A15,16	A17,18	A19,20	A21,22	A23,24
$\mathcal{N}_0$(late fusion)	80.8%	85.0%	81.6%	87.5%	86.6%	65.8%
$\mathcal{N}_0$(end-to-end)	82.4%	79.6%	83.2%	89.0%	85.5%	84.2%
$\mathcal{N}_1$(end-to-end)	77.5%	89.1%	86.6%	92.5%	89.1%	**90.6%**
$\mathcal{N}_2$(end-to-end)	**84.3%**	**89.7%**	**89.8%**	**93.3%**	**90.6%**	**90.6%**

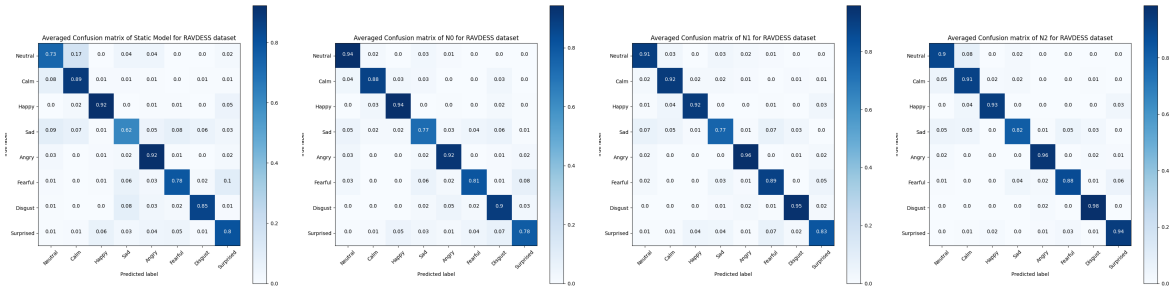


Figure 5.6. Averaged confusion matrices of tested models for RAVDESS dataset.

$\mathcal{N}_2$ (MRPN) over $\mathcal{N}_1$ suggests the level of data fuzziness can make the end-to-end multi-term optimization even harder without proposed RP components. The overall improvements suggest that multi-modal patterns are more generalized from the solution of $\mathcal{N}_2$ (MRPN).

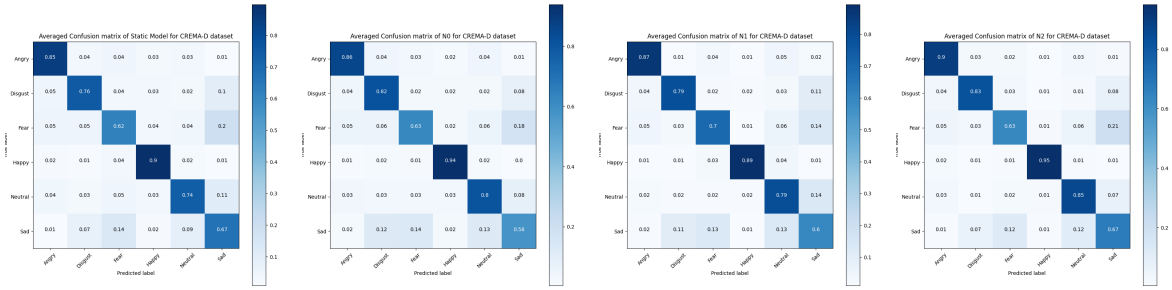


Figure 5.7. Averaged confusion matrices of tested models for Crema-d dataset.

5.3.6. Comparing baseline with SOTA

Our suggested MRPN achieves state-of-the-art performance on both datasets. It is compatible with any possible benefits derived from another new method. Experiments with pretraining the CNN extractors and time augmentation have strengthened the network's

Table 5.3. Comparison for Crema-d of MRPN approach (network $\mathcal{N}_2$) with simple fusion strategy ($\mathcal{N}_0$), and advanced fusion strategy ($\mathcal{N}_1$).

Crema-d	S1	S2	S3	S4	S5
$\mathcal{N}_0$ (late fusion)	76.5%	79.9%	76.6%	62.3%	78.2%
$\mathcal{N}_0$ (end-to-end)	77.3%	81.3%	79.2%	74.8%	78.6%
$\mathcal{N}_1$ (end-to-end)	72.6%	82.3%	77.3%	74.8%	74.2%
$\mathcal{N}_2$ (end-to-end)	**79.5**%	**83.0**%	**83.0**%	**76.8**%	**81.9**%
Crema-d	S6	S7	S8	S9	
$\mathcal{N}_0$ (late fusion)	81.8%	78.8%	80.0%	77.5%	
$\mathcal{N}_0$ (end-to-end)	**82.0**%	75.1%	79.5%	77.5%	
$\mathcal{N}_1$ (end-to-end)	**82.0**%	74.8%	79.3%	75.8%	
$\mathcal{N}_2$ (end-to-end)	**82.0**%	**80.0**%	**80.5**%	**78.6**%	

robustness in order to overcome overfitting problems associated with the limited quantity of training and testing data.

Additionally, we replaced LSTMs with Bidirectional LSTMs and Transformers as aggregators, but found no discernible change in their performance as sequence aggregators. Transformer derives its average feature from the decoded outputs; the idea is derived from ViT.

Table 5.4. Comparison of our fusion models with others recent solutions. Options used: IA – image augmentation, WO – without audio overlapping, VA – video frames augmentation, and AO – audio overlapping. X symbol – there is no report from authors for the given dataset.

Model (our)	**RAVDESS**	**Crema-d**
$\mathcal{N}_0$ (end-to-end),Resnet18+LSTM, IA	83.20%	77.25%
$\mathcal{N}_0$ (end-to-end),Resnet18+LSTM, VA+WO	85.20%	79.25%
$\mathcal{N}_0$ (late fusion),Resnet18+LSTM, VA+AO	81.6%	76.84%
$\mathcal{N}_0$ (end-to-end),Resnet18+LSTM, VA+AO	87.55%	81.30%
$\mathcal{N}_1$ (end-to-end),Resnet18+LSTM, VA+AO [97]	89.8%	77.0%
MRPN (end-to-end),Resnet18+LSTM, VA+AO	**90.8%**	**83.00%**
MRPN (end-to-end),Resnet18+Transformer(avg), VA+AO	**91.4%**	**83.15%**
Model (others)	**RAVDESS**	**Crema-d**
(OpenFace/COVAREP features + LSTM) + Attention [64]	58.33%	65.00%
Dual Attention + LSTM [65]	67.7%	74.00%
Resnet101 + BiLSTM [111]	77.02%	X
custom CNN [69]	X	69.42%
Early Cross-modal + MFCC + MEL spectrogram [67]	83.6%	X
CNN + Fisher vector + Metric learning [72]	X	66.5%
custom CNN+Spectrogram [105]	79.5% (Audio)	X

6. Conclusion

We built a multi-modal emotion detection system in our thesis project to solve the AVER issue and obtained SOTA results. Throughout the development process, the most useful components serving as the basis for the final product were chosen, tested, and refined.

Together with the multi-term loss function, the suggested MPRN architecture produces better fused features from multi-modal inputs. We notice that inferior instances of multi-modal solutions are eliminated in comparison to uni-modal solutions.

Our findings reach an average accuracy of 91.4 percent on the RAVDESS dataset and 83.15 percent on the Crema-d dataset, although every practical technique other than MRPN, such as data pre-processing, spatial-temporal data augmentation, and transfer learning, all contributed to this accomplishment. By removing inferior instances, the MRPN approach improves the average recognition rate by about 2%. We found that the greatest improvement of MRPN for a subset is approximately 90

The suggested data pre-processing method of temporal augmentation improves the overall rate for both uni- and multi-modal data. Additionally, it demonstrates how data augmentation cannot generalize multi-modal characteristics owing to the shortcomings of current multi-modal solutions' BP.

Additionally, the MRPN approach demonstrates its use for multi-modal classifiers that deal with signal sources other than optical and auditory.

We saw early on in the creation of the thesis topic that ER research had tremendous promise in a variety of ways. Improvements to unimodal solutions may help multimodal systems as well. While AVER is not the final answer to ER, other kinds of human expressions, like as language, posture, and so on, may nevertheless help fill in the gaps in our knowledge of human emotions. While fusing the deep characteristics in the end does not result in adequate integration at the utterance level, there are still opportunities to improve spatial-temporal features via neural network architecture.

List of Symbols and Abbreviations

ANN – Artificial Neural Networks
AU – Action Unit
AVER – Audio-Video Emotion Recognition
BP – Backpropagation
BPTT – Backpropagation Through Time
CFCCs – mel-frequency cepstral coefficients
CK+ – Extended Cohn-Kanade Dataset
CNN – Convolution Neural Network
DNN – Deep Neural Network
ER – Emotion recognition
EEG – Electroencephalography
F0 – fundamental frequency
FC – Fusion component
FER – Facial Emotion Recognition
fp68 – facial salient points
FACS – Facial Action Coding System
GMM – Gaussian mixture model
GPU – Graphics Processing Unit
GRU – Gated Recurrent Units
HCI – Human-Computer Actition
HMM – hidden Markov model
HOG – Historgram Oriented Gradients
IEMOCAP – Interactive Emotional Dyadic Motion Capture dataset
LDA – Linear Discriminant Analysis
LMM – Levenberg Marquardt Method
LPCC – linear prediction cepstral coefficients
LSTM – Long Short-term Memory
MRPN – Multimodal Residual Perceptron Network
NLP – Nartual Language Processing
NN – Neural Network
PCA – Principal Component Analysis
RaFD – The Radboud Faces Database
RAVDESS – Ryerson Audio-Visual Database of Emotional Speech and Song
RNN – Recurrent Neural Networks
ROI – Region of Interest
ViT – Vision Transformer
SAC – Sequence Aggregation Component
SER – Speech Emotion Recognition

SOTA – state-or-the-art
SU – Shape Unit
SVM – Support Vector Machine

List of Figures

1.1 Video frames of visual facial expressions selected from RAVDESS (Ryerson Audio-Visual Database of Emotional Speech and Song) dataset. 10
1.2 Mel Spectrograms of vocal timbres selected from RAVDESS (Ryerson Audio-Visual Database of Emotional Speech and Song) dataset. 11
2.1 Neuron element diagram from a biological perspective. (Image source: Alan Woodruff / QBI) 13
2.2 Artificial neural element versus biological neural element (Image source: https://cs231n.github.io/neural-networks-1/) 14
2.3 Block box system as system modeling concept. 15
2.4 The general idea of error gradient backpropagation through the neural component. 15
2.5 Examples of commonly used activation functions in neural networks. 17
2.6 Examples of gradients for commonly used activation functions. 18
2.7 Graphical illustration of model fitting to data. (Image source: geeksforgeeks . . 19
2.8 Typical model over-fitting observed for errors while the model is being trained. . 19
2.9 Typical model under-fitting observed for errors while the model is being trained. 20
2.10 Misfitting in the training. left: training curve of the same training data.middle: validation curve on one validation dataset. right: validation curve on the other validation dataset. 20
2.11 Data augmentation in image pixel domain and image color space.(Image source: https://github.com/aleju/imgaug) 21
2.12 An example of trained CNN and the middle features extracted by the trained kernels.(Image source: http://cs231n.stanford.edu/) 22
2.13 Recurrent neural network. (Image source: https://medium.datadriveninvestor.com/) 23
2.14 LSTM. (Image source: https://medium.datadriveninvestor.com/) 24
3.1 Examples of eigenface (top) and fisherface (bottom). 27
3.2 HOG features. [32] 28
3.3 Illustration of SVM working as the classifier maximizing the margin between two categories. 29
3.4 Face detection: deep features vs HOG features. Image source: towardsdatascience 29
3.5 Random erasing in the data augmentation and its benefited results. 30
3.6 Facial muscles important in forming facial actions relevant to visual emotions.(Image source: https://openstax.org/books/anatomy-and-physiology/pages/preface) 31
3.7 Facial Action Units extacted from Cohn and Kanade dataset. 32
3.8 Candide-3 model. 33
3.9 General concept of transfer learning. 33
4.1 Facial feature points indexed in FP68 categorization. 36
4.4 example 41

4.6 Samples from the training dataset (top: Cohn and Kanade dataset.) and testing dataset (bottom: RaFD). 43

4.7 Image augmentation results from Cohn and Kanade dataset, final augmentation consists of all operations applied in random order. 45

4.8 Cropped faces from Cohn and Kanade dataset used for training phase after augmentation procedure. 46

4.9 Image samples from VGG-Face dataset used for training Face Descriptor model. . 48

4.10 Samples from FER2013 database compared with samples from JAFFE and CK+ datasets. 50

4.11 Contaminated image samples in FER2013 dataset. 50

4.12 Histogram of class labels for FER2013 training dataset. 51

4.13 Augmented samples from FER2013 dataset for different emotions. 51

4.14 Impact of learning rate: top-left: training accuracy; top-right: training loss; bottom-left:validation accuracy; bottom-right: validation loss. 52

4.15 Impact of fine-tuning strategies: top-left: training accuracy; top-right: training loss; bottom-left:validation accuracy; bottom-right: validation loss. 53

4.16 Misclassification of single frame based CNN facial emotion classifier from some frame sequences, where yellow frame stands for recognition of angry emotion and blue frame stands for the recognition of neutral emotion. 54

4.17 Resnet-LSTM results for the streaming classification results, the pink frame stands for the recognition of angry expression. 56

4.18 Example of segments with a fixed duration from the original spectrogram. 57

4.19 Spectrogram comparison. top: our proposed windowing; middle: having too big hopping steps at 128. button: having too small hopping steps at 32. 58

4.20 Overlapped spectrogram segments generated from the original spectrogram sample. 59

5.1 Visualization (t-SNE algorithm) of deep features clustering from two different setups where the train/validation sets are shuffled. The clustering with respect to emotion classes are listed. 0: *neutral*, 1: *calm*, 2: *happy*, 3: *sad*, 4: *angry*, 5: *fearful*, 6: *disgust*, 7: *surprised*. **Top part**: clustering results from one setup of uni-modalities and multi-modality. *Left part*: only image modality. *Middle part*: only audio modality. *Right part*: multi-modality. **Down part**: clustering results from the other setup where train/validation sets are shuffled. 62

5.2 Distorted gradients backpropagation in some modality since the gradients from fused layer makes impact on gradients flow into neural weights of both modalities. 63

5.3 The proposed multi-modal emotion recognition system using DNN approach. **Upper part:** Video frames and audio spectral segments get independent temporal embeddings to be fused by our multi-modal Residual Perceptron Network (MRPN). **Lower part:** MRPN performs in each modality normalizations via the proposed Residual Perceptrons and then scores their concatenated outputs in the Fusion Component. The uni-modal prediction branches are only active in training mode. 64
5.4 Evolution of network design for multi-modal fusion (presented for training mode). $\mathcal{N}_0$: Fusion component (FC) only. $\mathcal{N}_1$ ([97]): Beside FC, independent scoring of each modality is considered. $\mathcal{N}_2$: Extending $\mathcal{N}_1$ network by Residual Perceptrons (RP) in each modality branch. 66
5.5 Generalization of our MRPN fusion approach to many modalities. It could be used for either regression or classification applications. 68
5.6 Averaged confusion matrices of tested models for RAVDESS dataset. 71
5.7 Averaged confusion matrices of tested models for Crema-d dataset. 71

List of Tables

4.1 Accuracy results for selected features 46
4.2 Standard deviation and mean of accuracy for SVM classifiers 47
4.3 Transfer learning versus other train from scratch methods. 53
5.1 Comparison of single modalities models with $\mathcal{N}_0$ model (RAVDESS cases): VM – Visual Modality only, AM – Audio Modality only, JM – Joint Modalities ($\mathcal{N}_0$ model), T – having time augmentation by signal random slicing, NT – not having time augmentation. 70
5.2 Comparison for RAVDESS of MRPN approach (network $\mathcal{N}_2$) with late fusion strategy ($\mathcal{N}_0$), end-to-end strategy ($\mathcal{N}_0$), and advanced end-to-end fusion strategy ($\mathcal{N}_1$). 71
5.3 Comparison for Crema-d of MRPN approach (network $\mathcal{N}_2$) with simple fusion strategy ($\mathcal{N}_0$), and advanced fusion strategy ($\mathcal{N}_1$). 72
5.4 Comparison of our fusion models with others recent solutions. Options used: IA – image augmentation, WO – without audio overlapping, VA – video frames augmentation, and AO – audio overlapping. X symbol – there is no report from authors for the given dataset. 72

References

[1] P. Ekman, E. T. Rolls, D. I. Perrett, and H. D. Ellis, "Facial expressions of emotion: An old controversy and new findings [and discussion]," *Philosophical Transactions: Biological Sciences*, vol. 335, no. 1273, pp. 63–69, 1992. [Online]. Available: http://www.jstor.org/stable/55476

[2] P. Ekman and W. Friesen, *Facial Action Coding System*. Consulting Psychologists Press, 1978, no. v. 1. [Online]. Available: https://books.google.pl/books?id=08l6wgEACAAJ

[3] P. N. Belhumeur, J. a. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. fisherfaces: Recognition using class specific linear projection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 7, p. 711–720, Jul. 1997. [Online]. Available: https://doi.org/10.1109/34.598228

[4] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," in *Proceedings of the IEEE*, 1998, pp. 2278–2324.

[5] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2017.

[6] Y. Zhai and M. Shah, "Video scene segmentation using markov chain monte carlo," *IEEE Transactions on Multimedia*, vol. 8, no. 4, pp. 686–697, 2006.

[7] A. Ullah, J. Ahmad, K. Muhammad, M. Sajjad, and S. W. Baik, "Action recognition in video sequences using deep bi-directional lstm with cnn features," *IEEE Access*, vol. 6, pp. 1155–1166, 2018.

[8] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *CoRR*, vol. abs/1502.03167, 2015. [Online]. Available: http://arxiv.org/abs/1502.03167

[9] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Instance normalization: The missing ingredient for fast stylization," 2017.

[10] Y. Wu and K. He, "Group normalization," 2018.

[11] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, p. 84–90, May 2017. [Online]. Available: https://doi.org/10.1145/3065386

[12] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2015.

[13] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015.

[14] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," 2014.

[15] A. Graves, A. rahman Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," 2013.

[16] H. Sak, A. Senior, K. Rao, and F. Beaufays, "Fast and accurate recurrent neural network acoustic models for speech recognition," 2015.

[17] A. Graves, M. Liwicki, S. Fernández, R. Bertolami, H. Bunke, and J. Schmidhuber, "A novel connectionist system for unconstrained handwriting recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 5, pp. 855–868, 2009.

[18] Y. Zhang, Q. Liu, and L. Song, "Sentence-state lstm for text representation," 2018.

[19] K. Pichotta and R. J. Mooney, "Using sentence-level lstm language models for script inference," 2016.

[20] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and tell: A neural image caption generator," 2015.

[21] Z. Yang, Y.-J. Zhang, S. ur Rehman, and Y. Huang, “Image captioning with object detection and localization,” 2017.

[22] J. Yue-Hei Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici, “Beyond short snippets: Deep networks for video classification,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[23] A. Ullah, J. Ahmad, K. Muhammad, M. Sajjad, and S. Baik, “Action recognition in video sequences using deep bi-directional lstm with cnn features,” *IEEE Access*, vol. PP, pp. 1–1, 11 2017.

[24] 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,” 2020.

[25] M. Turk and A. Pentland, “Face recognition using eigenfaces,” in *Proceedings. 1991 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1991, pp. 586–591.

[26] V. Kshirsagar, M. Baviskar, and M. Gaikwad, “Face recognition using eigenfaces,” in *2011 3rd International Conference on Computer Research and Development*, vol. 2, 2011, pp. 302–306.

[27] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*. New Yotk: John Willey & Sons, 1973.

[28] R. A. Fisher, “The use of multiple measurements in taxonomic problems,” *Annals of Eugenics*, vol. 7, no. 7, pp. 179–188, 1936.

[29] D. E. King, “Max-margin object detection,” *arXiv:1502.00046 [cs.CV]*, 2015. [Online]. Available: http://arxiv.org/abs/1502.00046

[30] ——, “Dlib-ml: A machine learning toolkit,” *Journal of Machine Learning Research*, vol. 10, pp. 1755–1758, 2009.

[31] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05) - Volume 1 - Volume 01*, ser. CVPR ’05. Washington, DC, USA: IEEE Computer Society, 2005, pp. 886–893. [Online]. Available: http://dx.doi.org/10.1109/CVPR.2005.177

[32] ——, “Histograms of oriented gradients for human detection,” in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05)*, vol. 1, 2005, pp. 886–893 vol. 1.

[33] F. Baumann, “Action recognition with hog-of features,” in *Pattern Recognition*, J. Weickert, M. Hein, and B. Schiele, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 243–248.

[34] B. Kwolek, “Face detection using convolutional neural networks and gabor filters,” in *Artificial Neural Networks: Biological Inspirations – ICANN 2005*, W. Duch, J. Kacprzyk, E. Oja, and S. Zadrożny, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 551–556.

[35] H. Li, Z. Lin, X. Shen, J. Brandt, and G. Hua, “A convolutional neural network cascade for face detection,” in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 5325–5334.

[36] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, “Joint face detection and alignment using multitask cascaded convolutional networks,” *IEEE Signal Processing Letters*, vol. 23, no. 10, pp. 1499–1503, Oct 2016.

[37] Z. Zhong, L. Zheng, G. Kang, S. Li, and Y. Yang, “Random erasing data augmentation,” 2017.

[38] A. J. Calder, A. Burton, P. Miller, A. W. Young, and S. Akamatsu, “A principal component analysis of facial expressions,” *Vision Research*, vol. 41, no. 9, pp. 1179–1208, 2001. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0042698901000025

[39] A. Hassan Mansour, G. Z. Alabdeen Salh, and A. S. Alhalemi, “Facial expressions

recognition based on principal component analysis (pca)," *International Journal of Computer Trends and Technology*, vol. 18, no. 5, p. 188–194, Dec 2014. [Online]. Available: http://dx.doi.org/10.14445/22312803/IJCTT-V18P143

[40] N. Bajaj and S. L. Happy, "Dynamic model of facial expression recognition based on eigen-face approach," 11 2013.

[41] M. Anggo and L. Arapu, "Face recognition using fisherface method," *Journal of Physics: Conference Series*, vol. 1028, p. 012119, 06 2018.

[42] I. Gangopadhyay, A. Chatterjee, and I. Das, "Face detection and expression recognition using haar cascade classifier and fisherface algorithm," in *Recent Trends in Signal and Image Processing*, S. Bhattacharyya, S. K. Pal, I. Pan, and A. Das, Eds. Singapore: Springer Singapore, 2019, pp. 1–11.

[43] J. Ahlberg, "Candide-3 - an updated parameterised face," 2001.

[44] D. Hazarika, S. Poria, R. Zimmermann, and R. Mihalcea, "Conversational transfer learning for emotion recognition," 2020.

[45] J. Deng, Z. Zhang, E. Marchi, and B. Schuller, "Sparse autoencoder-based feature transfer learning for speech emotion recognition," in *2013 Humaine Association Conference on Affective Computing and Intelligent Interaction*, 2013, pp. 511–516.

[46] S. Kumari, U. Kowsalya, R. Preethi, R. Theepa, J. Paulraj, and S. JeyaAnusuya, "Audio-visual emotion recognition using 3dcnn and dbn techniques," *International Journal of Advance Research, Ideas and Innovations in Technology*, vol. 4, pp. 634–639, 2018.

[47] S. Ji, W. Xu, M. Yang, and K. Yu, "3d convolutional neural networks for human action recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, pp. 221–231, 2013.

[48] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. V. Gool, "Temporal segment networks for action recognition in videos," 2017.

[49] Y. Fan, X. Lu, D. Li, and Y. Liu, "Video-based emotion recognition using cnn-rnn and c3d hybrid networks," in *Proceedings of the 18th ACM International Conference on Multimodal Interaction*, ser. ICMI '16. New York, NY, USA: Association for Computing Machinery, 2016, p. 445–450. [Online]. Available: https://doi.org/10.1145/2993148.2997632

[50] S. R. Bandela and T. K. Kumar, "Stressed speech emotion recognition using feature fusion of teager energy operator and mfcc," in *2017 8th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, 2017, pp. 1–5.

[51] S. G. Koolagudi, Y. V. Murthy, and S. P. Bhaskar, "Choice of a classifier, based on properties of a dataset: Case study-speech emotion recognition," *Int. J. Speech Technol.*, vol. 21, no. 1, p. 167–183, Mar. 2018. [Online]. Available: https://doi.org/10.1007/s10772-018-9495-8

[52] R. Xia and Y. Liu, "A multi-task learning framework for emotion recognition using 2d continuous space," *IEEE Transactions on Affective Computing*, vol. 8, no. 1, pp. 3–14, 2017.

[53] S. Zhang, S. Zhang, T. Huang, and W. Gao, "Speech emotion recognition using deep convolutional neural network and discriminant temporal pyramid matching," *IEEE Transactions on Multimedia*, vol. 20, no. 6, pp. 1576–1590, 2018.

[54] T. L. Nwe, S. W. Foo, and L. C. De Silva, "Speech emotion recognition using hidden markov models," *Speech Communication*, vol. 41, no. 4, pp. 603–623, 2003. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167639303000992

[55] E. Tzinis and A. Potamianos, "Segment-based speech emotion recognition using recurrent neural networks," in *2017 Seventh International Conference on Affective Computing and Intelligent Interaction (ACII)*, 2017, pp. 190–195.

[56] S. Mirsamadi, E. Barsoum, and C. Zhang, "Automatic speech emotion recognition using recurrent neural networks with local attention," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 2227–2231.

[57] Y. Zhang, J. Du, Z. Wang, J. Zhang, and Y. Tu, "Attention based fully convolutional network for speech emotion recognition," *2018 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, Nov 2018. [Online]. Available: http://dx.doi.org/10.23919/APSIPA.2018.8659587

[58] R. Rana, "Gated recurrent unit (gru) for emotion classification from noisy speech," 2016.

[59] Y. Niu, D. Zou, Y. Niu, Z. He, and H. Tan, "A breakthrough in speech emotion recognition using deep retinal convolution neural networks," 2017.

[60] M. Chen and X. Zhao, "A multi-scale fusion framework for bimodal speech emotion recognition," in *INTERSPEECH*, 2020.

[61] H.-S. Choi, J.-H. Kim, J. Huh, A. Kim, J.-W. Ha, and K. Lee, "Phase-aware speech enhancement with deep complex u-net," 2019.

[62] N. Neverova, C. Wolf, G. W. Taylor, and F. Nebout, "Moddrop: adaptive multi-modal gesture recognition," 2015.

[63] V. Vielzeuf, S. Pateux, and F. Jurie, "Temporal multimodal fusion for video emotion classification in the wild," 2017.

[64] R. Beard, R. Das, R. W. M. Ng, P. G. K. Gopalakrishnan, L. Eerens, P. Swietojanski, and O. Miksik, "Multi-modal sequence fusion via recursive attention for emotion recognition," in *Proceedings of the 22nd Conference on Computational Natural Language Learning*. Brussels, Belgium: Association for Computational Linguistics, Oct. 2018, pp. 251–259. [Online]. Available: https://www.aclweb.org/anthology/K18-1025

[65] 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.

[66] A. Zadeh, P. P. Liang, N. Mazumder, S. Poria, E. Cambria, and L.-P. Morency, "Memory fusion network for multi-view sequential learning," 2018.

[67] E. Mansouri-Benssassi and J. Ye, "Speech emotion recognition with early visual cross-modal enhancement using spiking neural networks," in *2019 International Joint Conference on Neural Networks (IJCNN)*, 2019, pp. 1–8.

[68] S. Zhang, S. Zhang, T. Huang, W. Gao, and Q. Tian, "Learning affective features with a hybrid deep model for audio–visual emotion recognition," *IEEE Trans. Cir. and Sys. for Video Technol.*, vol. 28, no. 10, p. 3030–3043, Oct. 2018. [Online]. Available: https://doi.org/10.1109/TCSVT.2017.2719043

[69] N. Ristea, L. C. Duţu, and A. Radoi, "Emotion recognition system from speech and visual information based on convolutional neural networks," in *2019 International Conference on Speech Technology and Human-Computer Dialogue (SpeD)*, 2019, pp. 1–6.

[70] E. Tzinis, S. Wisdom, T. Remez, and J. R. Hershey, "Improving on-screen sound separation for open domain videos with audio-visual self-attention," 2021.

[71] Y. Wu, L. Zhu, Y. Yan, and Y. Yang, "Dual attention matching for audio-visual event localization," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[72] E. Ghaleb, M. Popa, and S. Asteriadis, "Metric learning-based multimodal audio-visual emotion recognition," *IEEE MultiMedia*, vol. 27, no. 1, pp. 37–48, 2020.

[73] X. Cao, Y. Wei, F. Wen, and J. Sun, "Face alignment by explicit shape regression," *Int. J. Comput. Vision*, vol. 107, no. 2, pp. 177–190, Apr. 2014. [Online]. Available: http://dx.doi.org/10.1007/s11263-013-0667-3

[74] V. Kazemi and J. Sullivan, "One millisecond face alignment with an ensemble of regression trees," in *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*, ser. CVPR '14. Washington, DC, USA: IEEE Computer Society, 2014, pp. 1867–1874.

[75] J. H. Friedman, "Stochastic gradient boosting," *Comput. Stat. Data Anal.*, vol. 38, no. 4, pp. 367–378, Feb. 2002. [Online]. Available: http://dx.doi.org/10.1016/S0167-9473(01)00065-2

[76] P. Ekman and W. V. Friesen, *Facial Action Coding System*. Consulting Psychologist Press, 1977.

[77] P. Lucey, J. F. Cohn, T. Kanade, J. Saragih, Z. Ambadar, and I. Matthews, "The extended cohn-kanade dataset (ck+): A complete dataset for action unit and emotion-specified expression," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops*, 2010, pp. 94–101.

[78] O. Langner, R. Dotsch, G. Bijlstra, D. H. J. Wigboldus, S. T. Hawk, and A. van Knippenberg, "Presentation and validation of the radboud faces database," *Cognition and Emotion*, vol. 24, no. 8, pp. 1377–1388, 2010. [Online]. Available: https://doi.org/10.1080/02699930903485076

[79] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," 2017.

[80] O. Parkhi, A. Vedaldi, and A. Zisserman, "Deep face recognition." British Machine Vision Association, 2015, pp. 1–12.

[81] W. Skarbek, "Symbolic tensor neural networks for digital media - from tensor processing via BNF graph rules to CREAMS applications," *CoRR*, vol. abs/1809.06582, 2018. [Online]. Available: http://arxiv.org/abs/1809.06582

[82] M. Lyons, M. Kamachi, and J. Gyoba, "The Japanese Female Facial Expression (JAFFE) Dataset," Apr. 1998, 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. [Online]. Available: https://doi.org/10.5281/zenodo.3451524

[83] D. Masko and P. Hensman, "The impact of imbalanced training data for convolutional neural networks," 2015.

[84] Y. Gan, "Facial expression recognition using convolutional neural network," in *Proceedings of the 2Nd International Conference on Vision, Image and Signal Processing*, ser. ICVISP 2018. New York, NY, USA: ACM, 2018, pp. 29:1–29:5. [Online]. Available: http://doi.acm.org/10.1145/3271553.3271584

[85] 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," *Neural Networks*, vol. 64, pp. 59 – 63, 2015, special Issue on "Deep Learning of Representations". [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0893608014002159

[86] C. Pramerdorfer and M. Kampel, "Facial expression recognition using convolutional neural networks: State of the art," *CoRR*, vol. abs/1612.02903, 2016. [Online]. Available: http://arxiv.org/abs/1612.02903

[87] C. Li, N. Ma, and Y. Deng, "Multi-network fusion based on cnn for facial expression recognition," in *2018 International Conference on Computer Science, Electronics and*

Communication Engineering (CSECE 2018). Atlantis Press, 2018/02. [Online]. Available: https://doi.org/10.2991/csece-18.2018.35

[88] Y. Tang, "Deep learning using support vector machines," *CoRR*, vol. abs/1306.0239, 2013. [Online]. Available: http://arxiv.org/abs/1306.0239

[89] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[90] E. S. Salama, R. A. El-Khoribi, M. E. Shoman, and M. A. Wahby Shalaby, "A 3d-convolutional neural network framework with ensemble learning techniques for multi-modal emotion recognition," *Egyptian Informatics Journal*, vol. 22, no. 2, pp. 167–176, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1110866520301389

[91] F. Noroozi, M. Marjanovic, A. Njegus, S. Escalera, and G. Anbarjafari, "Audio-visual emotion recognition in video clips," *IEEE Transactions on Affective Computing*, vol. 10, no. 01, pp. 60–75, jan 2019.

[92] M. S. Hossain and G. Muhammad, "Emotion recognition using deep learning approach from audio–visual emotional big data," *Information Fusion*, vol. 49, pp. 69–78, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1566253517307066

[93] F. Ma, W. Zhang, Y. Li, S.-L. Huang, and L. Zhang, "Learning better representations for audio-visual emotion recognition with common information," *Applied Sciences*, vol. 10, p. 7239, 10 2020.

[94] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. V. Gool, "Temporal segment networks: Towards good practices for deep action recognition," 2016.

[95] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," 2014.

[96] J. Carreira and A. Zisserman, "Quo vadis, action recognition? a new model and the kinetics dataset," 2018.

[97] W. Wang, D. Tran, and M. Feiszli, "What makes training multi-modal classification networks hard?" in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 12 692–12 702.

[98] T. Standley, A. R. Zamir, D. Chen, L. Guibas, J. Malik, and S. Savarese, "Which tasks should be learned together in multi-task learning?" 2020.

[99] R. Caruana, "Multitask learning," *Mach. Learn.*, vol. 28, no. 1, p. 41–75, Jul. 1997. [Online]. Available: https://doi.org/10.1023/A:1007379606734

[100] T. Yu, S. Kumar, A. Gupta, S. Levine, K. Hausman, and C. Finn, "Gradient surgery for multi-task learning," *ArXiv*, vol. abs/2001.06782, 2020.

[101] W. Wang, Y. Fu, Q. Sun, T. Chen, C. Cao, Z. Zheng, G. Xu, H. Qiu, Y.-G. Jiang, and X. Xue, "Learning to augment expressions for few-shot fine-grained facial expression recognition," 2020.

[102] H.-W. Ng, V. D. Nguyen, V. Vonikakis, and S. Winkler, "Deep learning for emotion recognition on small datasets using transfer learning," in *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction*, ser. ICMI '15. New York, NY, USA: Association for Computing Machinery, 2015, p. 443–449. [Online]. Available: https://doi.org/10.1145/2818346.2830593

[103] A. Dhall, A. Kaur, R. Goecke, and T. Gedeon, "Emotiw 2018: Audio-video, student engagement and group-level affect prediction," 2018.

[104] C. Busso, M. Bulut, C.-C. Lee, A. Kazemzadeh, E. Mower Provost, S. Kim, J. Chang, S. Lee, and

S. Narayanan, “Iemocap: Interactive emotional dyadic motion capture database,” *Language Resources and Evaluation*, vol. 42, pp. 335–359, 12 2008.

[105] Mustaqeem and S. Kwon, “A cnn-assisted enhanced audio signal processing for speech emotion recognition,” *Sensors*, vol. 20, no. 1, 2020. [Online]. Available: https://www.mdpi.com/1424-8220/20/1/183

[106] S. Latif, R. Rana, J. Qadir, and J. Epps, “Variational autoencoders for learning latent representations of speech emotion,” *CoRR*, vol. abs/1712.08708, 2017. [Online]. Available: http://arxiv.org/abs/1712.08708

[107] H. Cao, D. G. Cooper, M. K. Keutmann, R. C. Gur, A. Nenkova, and R. Verma, “Crema-d: Crowd-sourced emotional multimodal actors dataset,” *IEEE Transactions on Affective Computing*, vol. 5, no. 4, pp. 377–390, 2014.

[108] 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, pp. 1–35, 05 2018. [Online]. Available: https://doi.org/10.1371/journal.pone.0196391

[109] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” *CoRR*, vol. abs/1512.03385, 2015. [Online]. Available: http://arxiv.org/abs/1512.03385

[110] A. Nagrani, J. S. Chung, and A. Zisserman, “Voxceleb: A large-scale speaker identification dataset,” *Interspeech 2017*, Aug 2017. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2017-950

[111] 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.

List of Xin Chang publications

[112] K. Yuksel, X. Chang, and W. Skarbek, "Smile detectors correlation," in *Photonics Applications in Astronomy, Communications, Industry, and High Energy Physics Experiments 2017*, R. S. Romaniuk and M. Linczuk, Eds., vol. 10445, International Society for Optics and Photonics. SPIE, 2017, pp. 479 – 490. [Online]. Available: https://doi.org/10.1117/12.2280760

[113] X. Chang and W. Skarbek, "Facial expressions recognition by animated motion of Candide 3D model," in *Photonics Applications in Astronomy, Communications, Industry, and High-Energy Physics Experiments 2018*, R. S. Romaniuk and M. Linczuk, Eds., vol. 10808, International Society for Optics and Photonics. SPIE, 2018, pp. 41 – 50. [Online]. Available: https://doi.org/10.1117/12.2500175

[114] R. Pilarczyk, X. Chang, and W. Skarbek, "Human face expressions from images - 2d face geometry and 3d face local motion versus deep neural features," 2019.

[115] X. Chang and W. Skarbek, "From face identification to emotion recognition," in *Photonics Applications in Astronomy, Communications, Industry, and High-Energy Physics Experiments 2019*, R. S. Romaniuk and M. Linczuk, Eds., vol. 11176, International Society for Optics and Photonics. SPIE, 2019, pp. 141 – 149. [Online]. Available: https://doi.org/10.1117/12.2536735

[116] X. Chang and W. Skarbek, "Multimodal emotion classification by streaming fixed time segments for speaker movies," in *Photonics Applications in Astronomy, Communications, Industry, and High Energy Physics Experiments 2020*, R. S. Romaniuk and M. Linczuk, Eds., vol. 11581, International Society for Optics and Photonics. SPIE, 2020, pp. 54 – 65. [Online]. Available: https://doi.org/10.1117/12.2579932

[117] X. Chang and W. Skarbek, "Multi-modal residual perceptron network for audio–video emotion recognition," *Sensors*, vol. 21, no. 16, 2021. [Online]. Available: https://www.mdpi.com/1424-8220/21/16/5452