Multimodal Emotion Recognition with Deep Learning and Fusion Mechanisms

Ali Kanafin, Ayan Myrzakhmet, Zhaksylyk Kuanysh

Project Advisor/Co-Advisors: Adnan Yazici, Enver Ever, Minho Lee

1. Executive Summary

This report outlines an innovative research endeavor aimed at enhancing emotion recognition accuracy through a multimodal approach, integrating video, audio, and electroencephalography (EEG) data. The goal was to construct a sophisticated model that leverages both physiological and non-physiological inputs for more precise emotion detection.

Key Research Components:

- Literature Review: Our initial investigation into existing emotion recognition techniques underscored the necessity for an integrated approach. Insights drawn from established research informed our selection of technologies and strategies, particularly the incorporation of advanced deep-learning techniques such as CNNs, attention mechanisms, and different fusion techniques.
- Data Collection and Preprocessing: The project utilized prominent datasets, including RAVDESS for audiovisual signals and FER-2013 for facial expressions. A significant challenge was the absence of a comprehensive dataset that encapsulated all desired

modalities, which hindered the ability to create a unified model. Eventually, we could get Professor Minho's dataset, which had all 3 modalities.

- Model Development and Fusion Methods: We developed distinct models for each data type and explored several methods to combine these into a cohesive system. Techniques such as late fusions and intermediate attention proved crucial in enhancing the overall accuracy of our emotion recognition system.
- Implementation and Evaluation: The multimodal system was embedded into a user-friendly web application to allow for interactive emotion recognition. Evaluation of various datasets confirmed that our integrated approach was superior to single-modality models in terms of accuracy and reliability.
- Challenges and Prospects: The complex nature of EEG data and the scarcity of an all-encompassing multimodal dataset posed considerable challenges. Future initiatives will aim to refine these integration techniques and broaden the emotional and demographic scope of the datasets.

This study significantly advances the field of emotion recognition, demonstrating the effectiveness of a multimodal strategy in improving the accuracy of detected emotions, which is essential for applications across healthcare, marketing, and automated systems.

2. Introduction

Emotion recognition is a subject that has received a lot of investigation recently. While the majority have concentrated on a single modality, some have focused on multimodal recognition. However, research that takes into account multimodal emotion detection typically only uses physiological (EEG, EMG, EOG) or non-physiological (facial expressions, speech,

etc.) inputs. This study aims to create a model that identifies human emotion based on three different modalities: video, audio, and electroencephalography (EEG). Physiological and non-physiological signals from participants were mixed in this study so that the models from various signals could complement one another and increase the accuracy of the results. Given the publicly available datasets and datasets that were accessed, the way to create a model that recognizes emotion from audio, video, and EEG is a hybrid fusion of modalities. We trained a few different models, tried different approaches for fusion mechanisms. Eventually, we trained a decent multimodal model and got high accuracy for Professor Minho's dataset that includes all 3 modalities.This study will include the background and related work, approaches, execution, and evaluation of the project.

3. Background and Related Work

Last semester, we investigated different models for each modality along with fusion methods. Upon reviewing Venkataramanan and Rajamohan (2019), we found their work on emotion recognition from speech provides several useful insights that informed our approach. The paper presents a helpful survey of the current state of the field, highlighting popular datasets, feature extraction methods, and classification architectures commonly utilized. This helped us identify best practices to leverage when building our model. In particular, their discussion of MFCC features and recurrent neural networks reinforced the techniques we decided to employ for our task.

Research done by Chen et al. (2022) is one of the most crucial works for video emotion recognition. Their VGG16 + LSTM model was the most interesting in terms of the logical

architecture. They provided a comparison of this model to other models like MLP and PCA. Their model outperformed all others and reached 67.20%.

One of our ideas was to use one image/frame emotion recognition. For this approach we decided to work on the FER-2013 dataset, thus related works that we read for one image/frame emotion recognition were based on this dataset. Models and test accuracies of the best researchers in terms of the test accuracies are shown in the table below.

Related Works	Proposed Method	Test Accuracy
Talegaonkar et al. (2019)	CNN + Batch Normalization	60.12%
Agrawal & Mittal (2020)	CNN + Batch Normalization + Varying number of filters	65%
Arriaga et al. (2019)	CNN + Batch Normalization + GAP	66%
Mollahosseini et al. (2016)	New architecture + polynomial learning rate	66.4%
Quinn et al. (2017)	Custom CNN	66.67%
Negara et al. (2020)	VGG-16 + GAP	69.40%
Minaee & Abdolrashidi (2019)	New architecture based on attentional CNN	70.02%
Tanget (2013)	Multi-Level CNN (MLCNN)	73.03%

Table 1. Related works on image/frame emotion recognition with their models and accuracies

Among these researches average test accuracy is 67.08%. However, these researches are the best compared to others. We decided to show you the best and work according to their models. However, some of these papers, not all of them, do not provide codes or all relevant information for models, and some of them are too advanced and out of our knowledge. So, we took the best according to our knowledge for image recognition. The third paper from the top done by Arriaga et al. (2019) best fits our condition. We took their model as a base for one of our own custom CNN models. According to their research, this model got around 66% test accuracy

on the FER-2013 dataset. Unfortunately, they did not provide implemented code and specific values (batch sizes, stride sizes, kernels, etc.) except for the picture of its architecture (Figure 1).

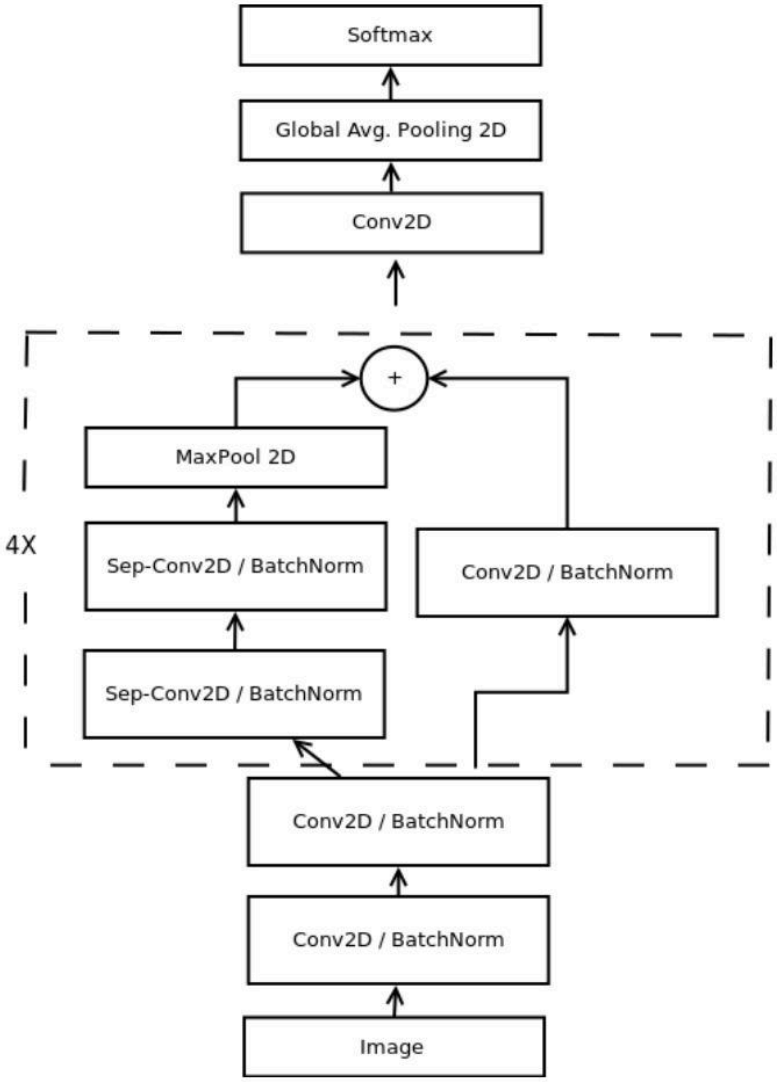


Figure 1. Custom CNN architecture

While researching, we came across a paper by Chumachenko, Iosifidis, and Gabbouj (2022). In addition to the article, they shared a GitHub repository with a code for image detection and code for training fusion mechanisms which trains models for recognizing emotions using

audio and video on the RAVDESS dataset. The RAVDESS dataset, short for the Ryerson Audio-Visual Database of Emotional Speech and Song, is a widely used database in emotion recognition research. It was developed by the Ryerson University in Toronto, Canada. The dataset contains audiovisual recordings of actors portraying different emotions through speech and song. The dataset covers 8 types of emotions, including neutral, calm, happy, sad, angry, fearful, surprised, and disgusted. Each actor portrays these emotions while speaking short, semantically neutral sentences and singing two full octaves of a neutral note. There are a total of 24 professional actors (12 female, 12 male) contributing to the dataset. These actors come from various ethnic backgrounds and ages, providing diversity in expression. The dataset includes both audio recordings and video recordings of the actors' performances. This dual modality allows researchers to explore emotion recognition using either or both modalities. Each audiovisual recording in the dataset is annotated with metadata including the gender, emotion portrayed, and other relevant information about the performance. The audio recordings are typically provided in WAV format, while video recordings are in MP4 format. Researchers and developers often use the RAVDESS dataset for tasks such as emotion recognition, speech analysis, and affective computing. It serves as a benchmark for evaluating algorithms and models in these domains. Overall, the RAVDESS dataset is valuable for advancing research in emotion recognition and related fields due to its diverse set of emotions, actors, and audiovisual content.

Chumachenko, Iosifidis, and Gabbouj (2022) have provided 3 options for the merging mechanism: late transformer, intermediate transformer, and intermediate attention.

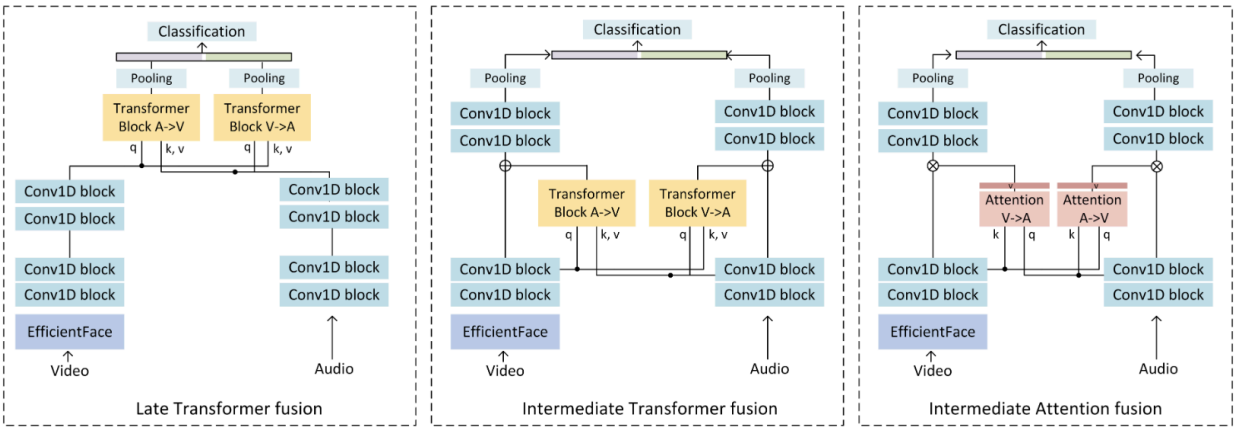


Figure 2. Three multimodal emotion recognition models for audio and video modalities

According to Chumachenko, Iosifidis, and Gabbouj (2022), intermediate attention was the best approach, because as a result, the model trained with intermediate attention showed the highest accuracy on two different datasets RAVDESS and MOSEI.

	RAVDESS			MOSEI		
	AV	A	V	AV	A	V
LT1	79.08	**59.16**	72.66	67.11	63.62	62.9
LT4	79.25	53.00	70.92	64.47	53.71	64.91
IT1	77.33	48.41	73.75	62.80	62.85	63.09
IT4	78.91	44.33	**74.92**	67.01	64.30	63.12
IA1	**81.58**	58.08	72.83	**67.19**	**64.52**	**64.91**
IA4	79.58	57.16	71.83	63.48	62.74	63.18

Table 1. Performance of different fusion methods on RAVDESS and MOSEI.

"AV", "A", and "V" in the table mean the input type of the data. In other words "AV" denotes Audio-Video input, "A" denotes Audio input only, and "V" denotes Video input only. "LT1"

denotes a late transformer with one number of heads, “LT4” denotes a late transformer with four numbers of heads. The approach goes for “IT” (intermediate transformer) and “IA” (intermediate attention).

We also searched for some fusion algorithms and techniques. The paper that was written by Wang Q., Wang M., Yang Y., and Zhang X. in 2022 provides four methods for the fusion mechanism. Among four of them, two methods got very good results:

1. AVER method: calculates the average sum of the probabilities, and predicts the final label.

$$p_i = \frac{1}{2}(e_i + s_i), i = 1, \ldots, 4$$

$$c = \arg\max_i (p_i)$$

Figure 3. AVER method function

p_i represents the average probability and c represents the label.

2. WAVER method: similar to the AVER but uses weighted sum instead.

$$p_i = a_1 * e_i + a_2 * s_i, i = 1, \ldots, 4$$

$$c = \arg\max_i (p_i)$$

Figure 4. WAVER method function

We read related research papers and found a popular multimodal dataset that includes EEG data, video, and video data (Soleymani et al., 2011). We requested the dataset, however they didn't reply. Then there was a recent multimodal dataset that contained all the modalities we need, however, they only provided extracted features and not the data itself and they got quite a low accuracy rate (Chen et al., 2022).

4. Project Approach

Requirements and Functionalities

Functional requirements:

- Emotion recognition model: Deep learning architecture for each type of modality should be chosen and it should be developed.
- Fusion mechanism: All methods must be interconnected to make correct predictions.
- Predicting emotions using uploaded data: The forecasting process should be carried out in two Audio-Video or Audio-Video-EEG modes.
- Downloading the results: After forecasting, the user should be able to download the results in the form of two files:
    - 1) An identical video file with the result written in the upper left corner
- User interface: A User-friendly interface for user interaction with the system should be developed.
- Data Logging and Storage: Data and predictions should be saved for future use by users, etc.

Non-functional requirements:

- Accuracy and Performance: The system should produce a very high probability and low false rates
- Scalability: The system should be available to process a different number of users and input data
- Security/Privacy: User data must be protected, user authentication must be implemented
- Ethical Considerations: Ethical issues related to emotion recognition, such as bias and privacy should be considered

Domain requirements:

- Emotion Datasets: Publicly available emotion datasets should be used that include all modalities and cover a wide range of emotions and demographic groups
- Emotion labels: A clearly defined emotional set should be used

The user hardware and software requirements:

- Users should have internet access and a device (Phone, PC, Laptop, etc.) that can use a Web Browser (Google, Yandex, Safari, etc.). Users should have files with collected data for Audio, Video, and EEG.

Hardware and software requirements for the project:

- Hardware:
    - PC or Laptop with sufficient GPU/CPU power to train deep learning models and run a server for web application
- Software:
    - IDE: Visual Studio Code
    - Git: Git/Gitlab
    - Frameworks: Django
    - Libraries: PyTorch, TensorFlow, pandas, numpy, sklearn, matplotlib, keras and etc
    - Backend:
        - Python
        - PostgreSQL
    - Frontend:
        - HTML/CSS
        - Bootstrap
    - Local server: Apachectl on Mac

Design

Stakeholders:

- Healthcare:
    - Doctors, therapists, counselors, and psychologists can use emotional recognition tools to assess the emotional states of patients, especially those with mental health conditions.
    - Nurses and caregivers can use these applications to monitor patients' emotional well-being in healthcare facilities.
- Marketing and Advertising:

- Marketing professionals can use emotional recognition to analyze customer reactions to advertisements and marketing campaigns.
- Advertisers can test the emotional impact of different ad creatives to optimize their messaging

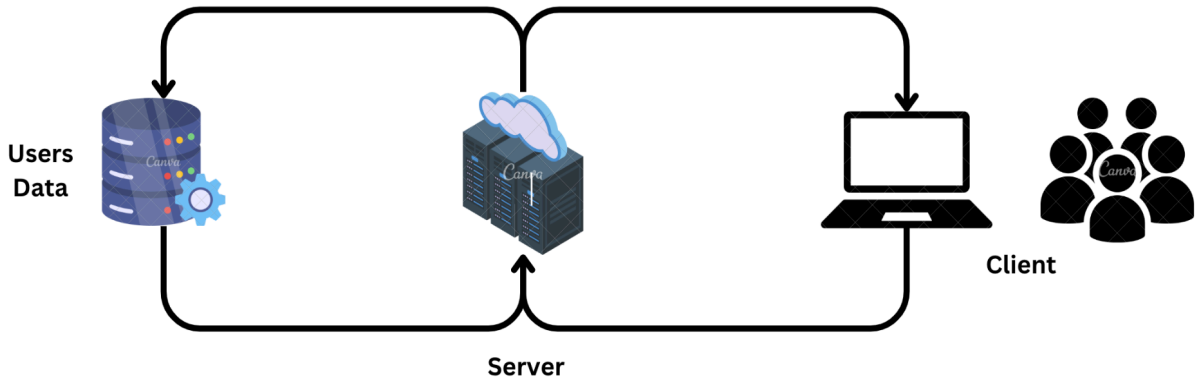


Figure 5. Design of the Server Architecture

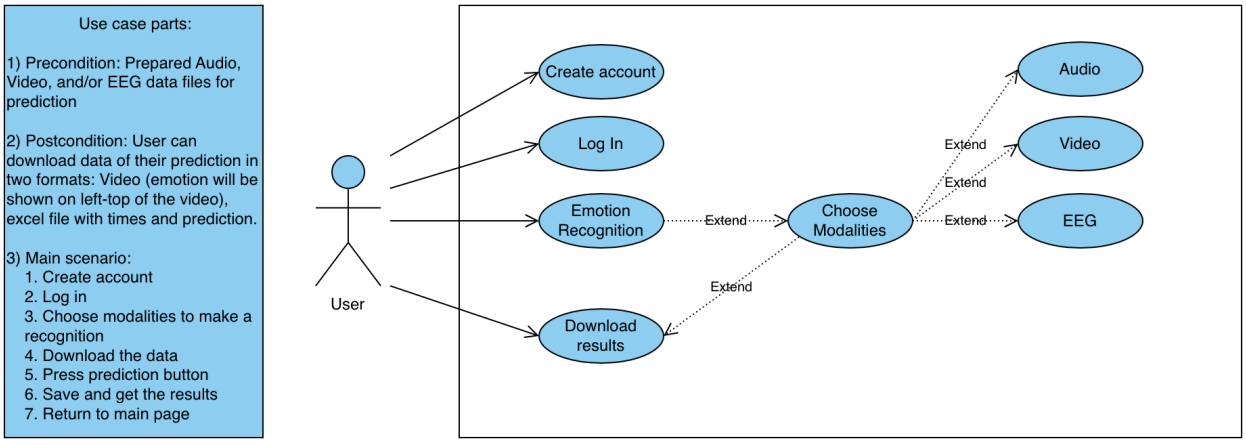
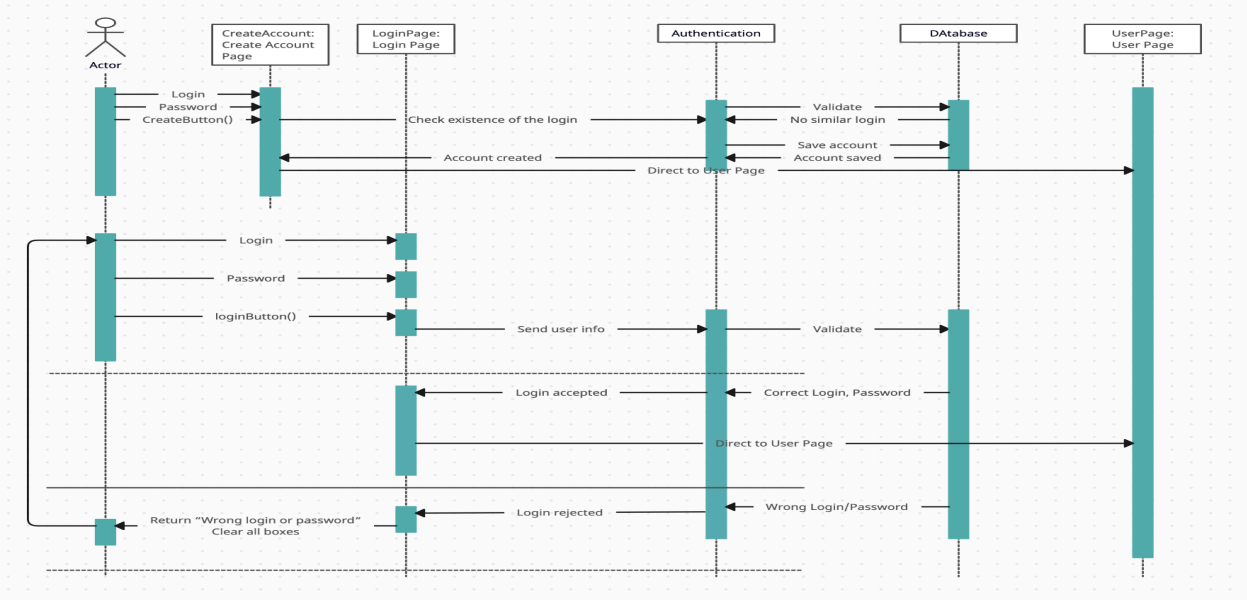
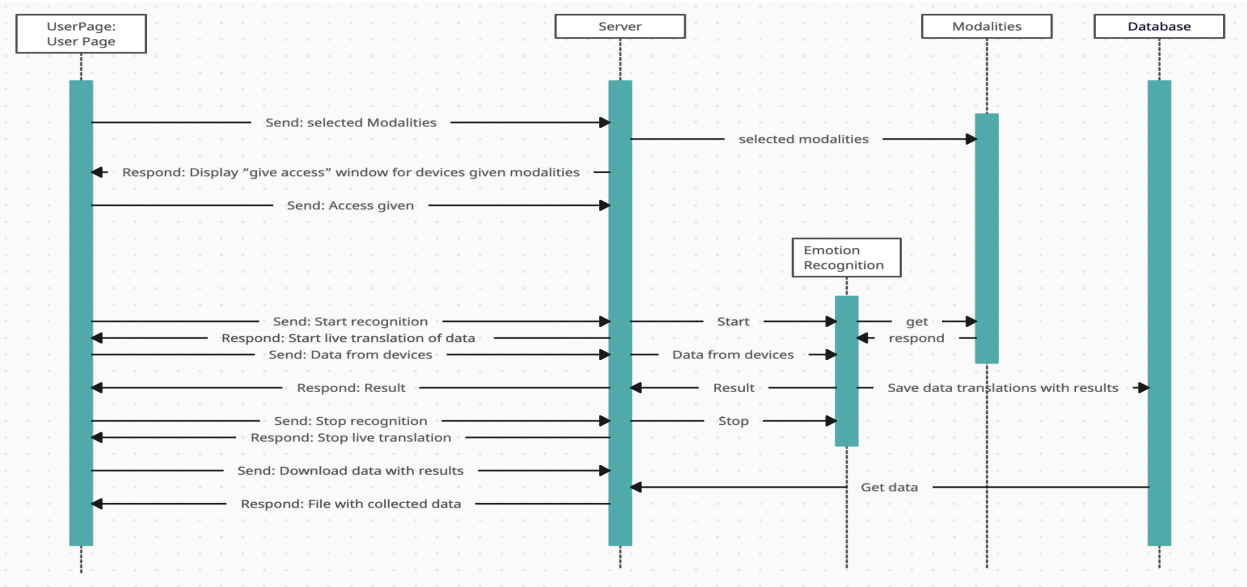


Figure 6. Use Case


Actor
CreateAccount: Create Account Page
LoginPage: Login Page
Authentication
DAtabase
UserPage: User Page
Login
Password
CreateButton()
Check existence of the login
Validate
No similar login
Save account
Account saved
Account created
Direct to User Page
Login
Password
loginButton()
Send user info
Validate
Login accepted
Correct Login, Password
Direct to User Page
Login rejected
Wrong Login/Password
Return "Wrong login or password"
Clear all boxes


UserPage: User Page
Server
Modalities
Database
Send: selected Modalities
selected modalities
Respond: Display "give access" window for devices given modalities
Send: Access given
Emotion Recognition
Send: Start recognition
Start
get
Respond: Start live translation of data
respond
Send: Data from devices
Data from devices
Respond: Result
Result
Save data translations with results
Send: Stop recognition
Stop
Respond: Stop live translation
Send: Download data with results
Get data
Respond: File with collected data

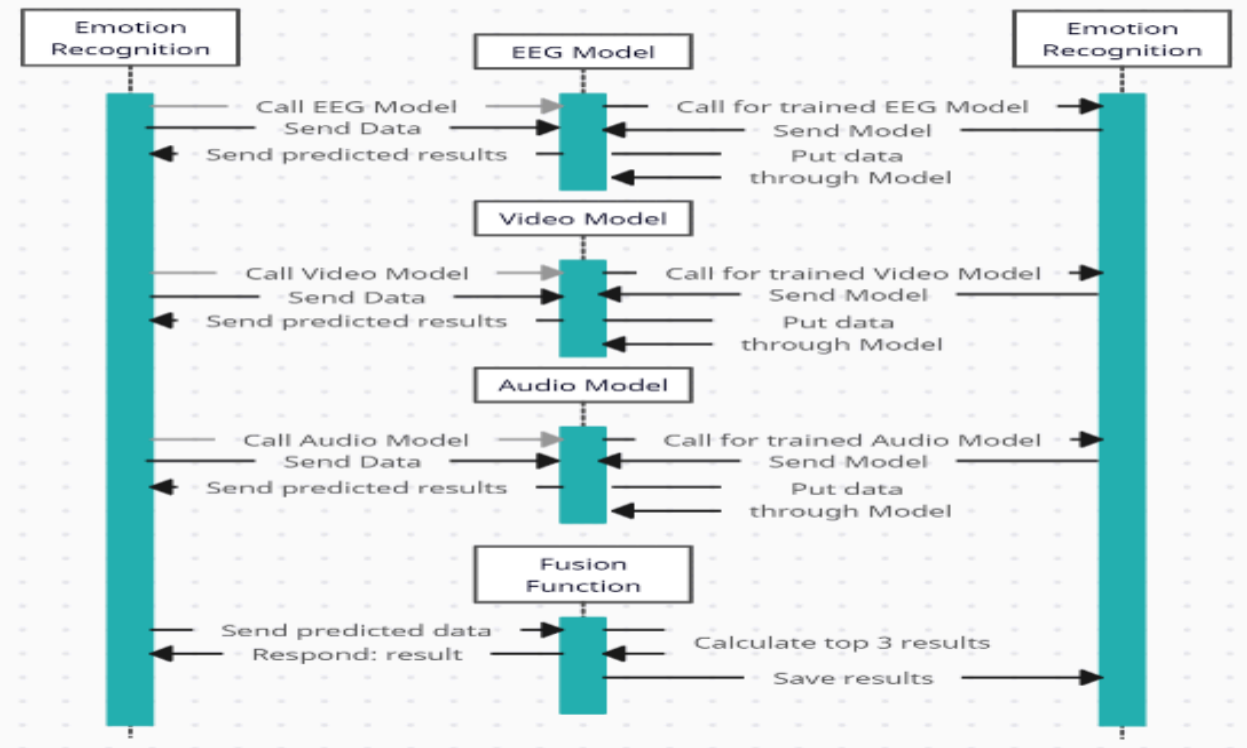


Figure 7. Sequence diagrams

Solutions and ideas for our requirements

As for the hardware, there were quite a few options. In the fall semester, we had options such as hosted Jupyter Notebook services (Collab, Kaggle) and public computers that are located in the university's computer lab. However, in the spring semester, Professor Minho Lee provided us with a super-powerful computer that can handle the training complex and complete datasets.

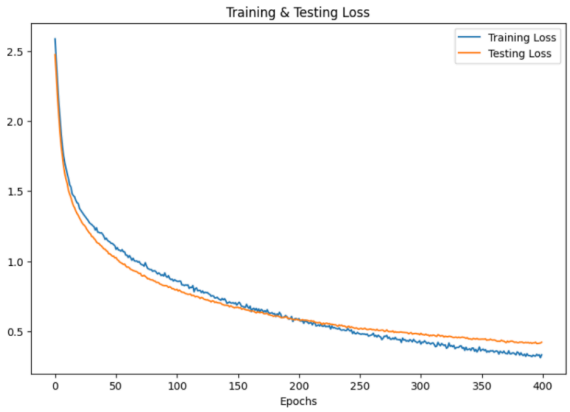
We had several options for solutions regarding the emotion recognition model. In the fall semester, we trained different models for each modality and could apply the late fusion mechanism. On the one hand, there was a public GitHub repository that was created by Chumachenko, Iosifidis, and Gabbouj (2022), which had different types of fusion mechanisms and excellent models using different techniques like dropout, regularization, data augmentation,

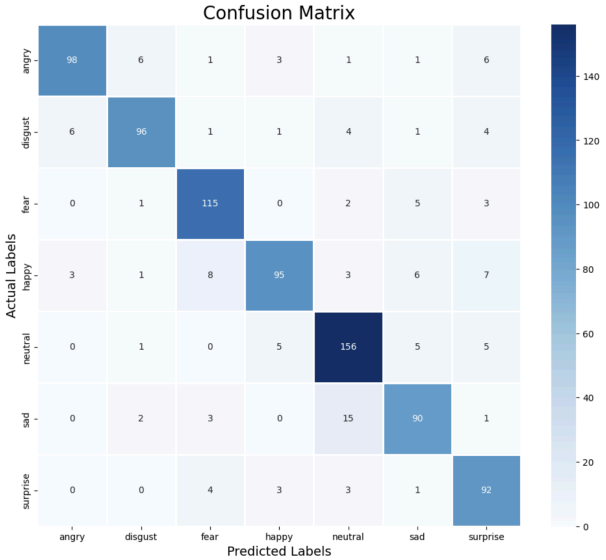
etc. So we came up with two approaches. The first one is to create three models for each of the modalities and in the end use late fusion mechanisms to get the prediction. The second one is to use one model for Audio-Video modalities and one model for EEG modality.

The first approach for creating an emotion recognition model

Audio Modal Recognition

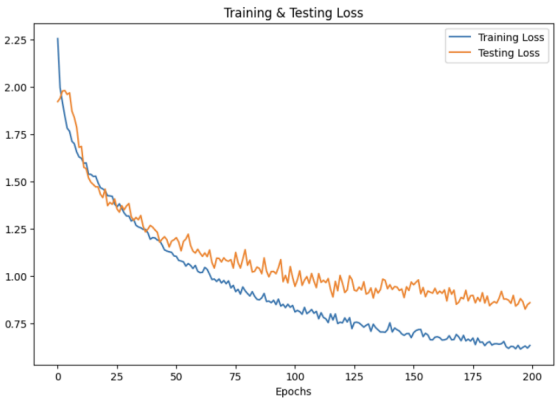
The RAVDESS dataset needed to create a model that recognizes emotions using sound contains 1440 audio files consisting of 60 files for each of the 24 actors (12 women, 12 men). The actors vocalized two statements in a neutral North American accent, conveying a range of emotions - happy, sad, angry, fearful, surprised, and disgusted. We wrote code that augments a speech emotion dataset with various audio perturbations to improve the model performance in real-world conditions. Four main augmentation techniques are applied including adding background noise, altering the speed, introducing subtle time offsets, and shifting the pitch. These aim to simulate noisy environments, different speaking rates, and emotional intonation that may be present in real speech. Features are then extracted from the original and augmented audio clips using Mel-frequency cepstral coefficients (MFCCs), an efficient spectral representation commonly used in speech analysis. All these processes are performed using the "librosa" library. Then we started developing models. After research on this topic, we decided to develop two different models using MLP and CNN. In the beginning, there were problems with overfitting, but we were able to avoid this by removing some layers, adding the dropout, and adjusting the size of neurons on the layer. The MLP model showed 84.72% accuracy on test data, which is much better than the CNN model, which showed 68% accuracy on test data.

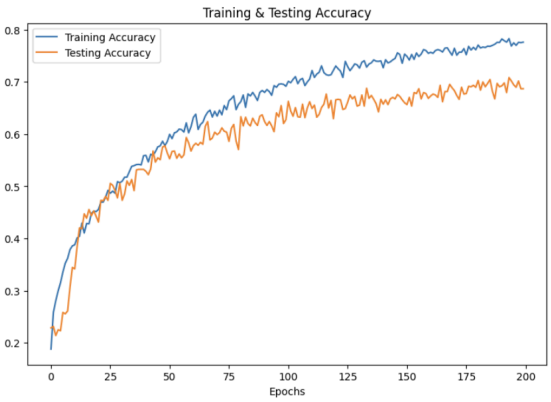


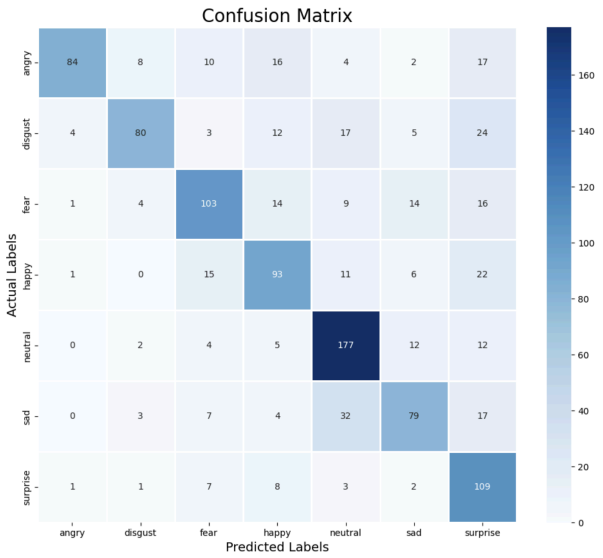


	precision	recall	f1-score	support
angry	0.92	0.84	0.88	116
disgust	0.90	0.85	0.87	113
fear	0.87	0.91	0.89	126
happy	0.89	0.77	0.83	123
neutral	0.85	0.91	0.88	172
sad	0.83	0.81	0.82	111
surprise	0.78	0.89	0.83	103
accuracy			0.86	864
macro avg	0.86	0.86	0.86	864
weighted avg	0.86	0.86	0.86	864

Figure 8. Results of the MLP model







	precision	recall	f1-score	support
angry	0.92	0.60	0.72	141
disgust	0.82	0.55	0.66	145
fear	0.69	0.64	0.66	161
happy	0.61	0.63	0.62	148
neutral	0.70	0.83	0.76	212
sad	0.66	0.56	0.60	142
surprise	0.50	0.83	0.63	131
accuracy			0.67	1080
macro avg	0.70	0.66	0.67	1080
weighted avg	0.70	0.67	0.67	1080

Figure 9. Results of the CNN model

Video Modal Recognition

We have prepared two methodologies to make face emotion recognition. The first methodology uses a sequence of frames from video, while the second recognizes emotion in one frame.

For the first methodology, we used the CNN + LSTM combined model that was provided in the research done by Chen et al. (2022). CNN extracts features from 30 images and puts these features as sequence inputs into the LSTM model. For CNN they used VGG16 and VGG19 pretrained models. We decided to use the VGG16 pre-trained model from the Keras applications library. We took output features as they did from the last max pooling layer of the VGG16 model. Extracted features of 30 frames were equal to 30 by 7 by 7 by 512 array, where each frame's extracted output shape was equal to 7 by 7 by 512. After that, we reshaped the output of each frame into one long features list with the shape of 25088 (7 * 7 * 512 = 25088). So our

input shape for the LSTM model was 30 by 25088, where 30 was the sequence length. To train this model we used s08 and s06 subjects from the PME4 dataset. They provide a numpy array of extracted features from 30 frames for each actor for different 7 classes (fear, anger, sad, happy, surprise, neutral, disgust). Combined subjects contain 697 of such numpy arrays. Then these 607 numbers of 30 by 25088 extracted features were used as inputs for the LSTM layer of the Keras layers library with 128 units and 30 by 25088 input shape. At the end of this model Dense layer with 7 class numbers and a softmax activation function was added. After 200 epochs with batch size 100, we got 60% accuracy on validation.

For the second methodology for one image/frame processing, we used two custom CNN models one ViT model from Timm's models, and two datasets. The main dataset was FER-2013 which contains 35527 grayscale face images (48 by 48 pixels) for 7 different emotions (fear, anger, sad, happy, surprise, neutral, disgust). The second dataset was a small part of big AffectNet which contains 29042 RGB face images (96 by 96 pixels) for 8 different emotions (fear, anger, sad, happy, surprise, neutral, disgust, contempt).

Label	Number of images	Emotion
0	4593	Angry
1	547	Disgust
2	5121	Fear
3	8989	Happy
4	6077	Sad
5	4002	Surprise
6	6198	Neutral

Label	Number of images	Emotion
0	3218	Angry
1	2477	Disgust
2	3176	Fear
3	5044	Happy
4	3091	Sad
5	4039	Surprise
6	5126	Neutral
7	2871	Contemp

Table 3. Number of images for each emotion (FER-2013 on the left and AffectNet on the right)

Our first custom CNN model was implemented by (Arriaga et al., 2017). According to their research, this model got around 66% test accuracy on the FER-2013 dataset. Unfortunately, they did not provide implemented code and specific values (batch sizes, stride sizes, kernels, etc.) except the picture of its architecture (Figure 10).

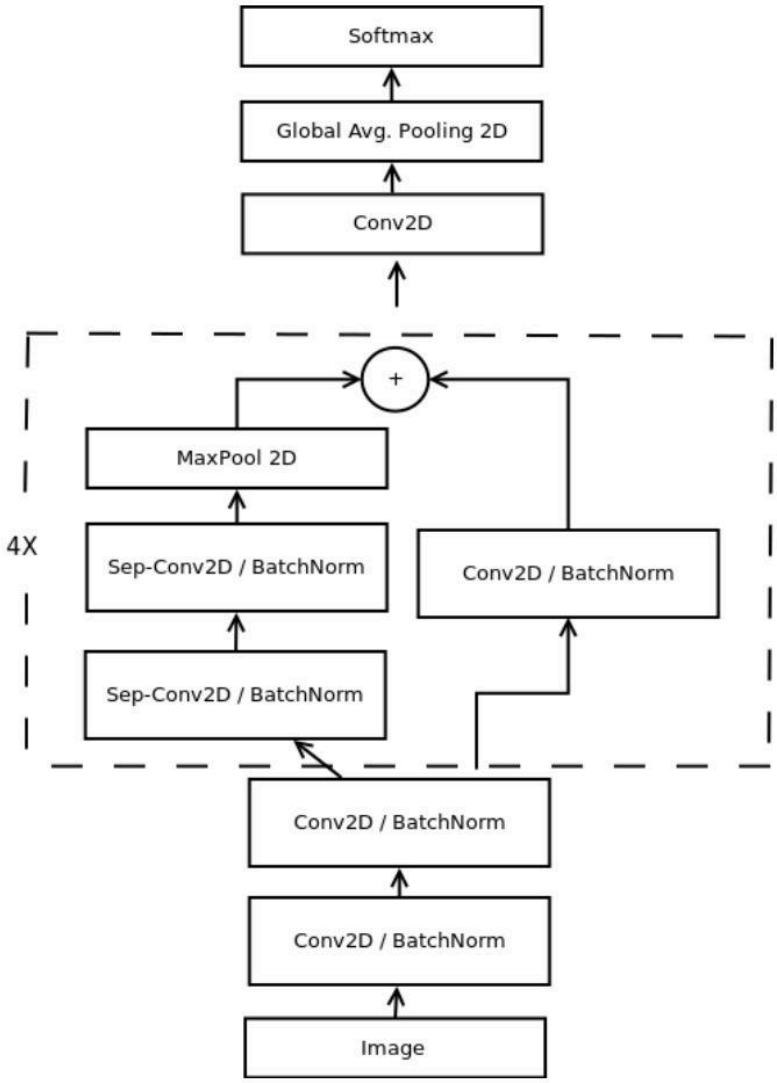
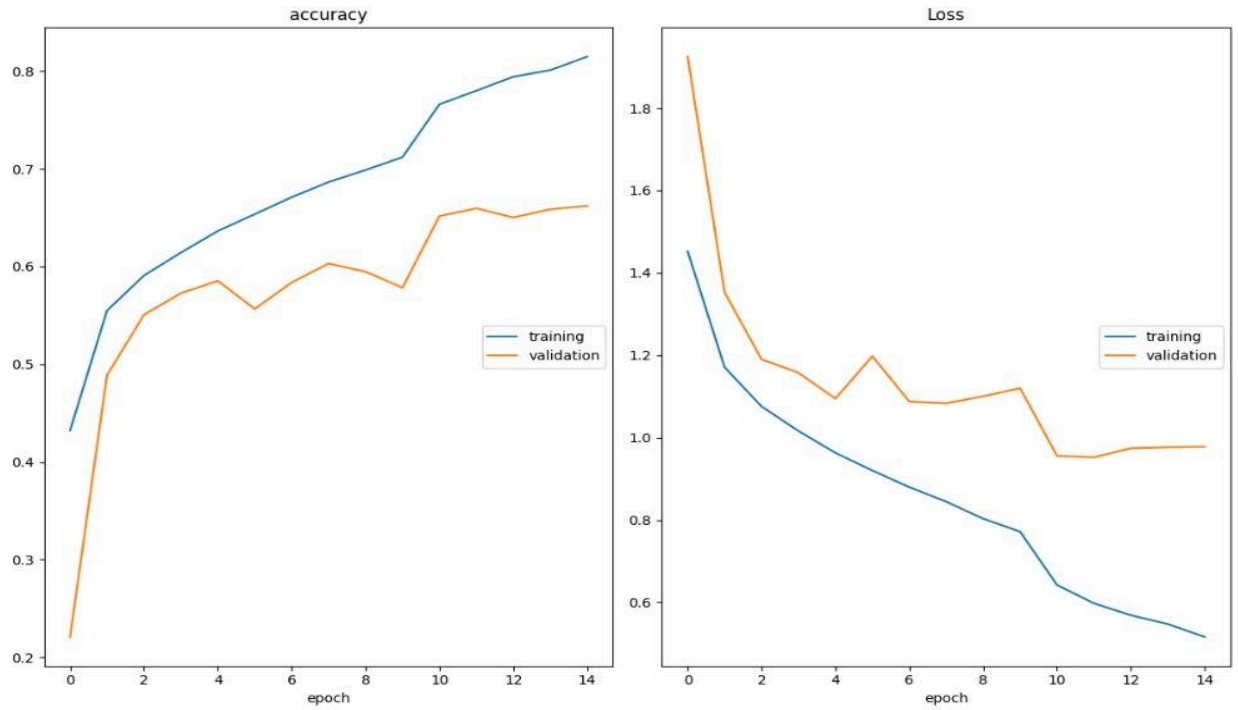


Figure 10. Custom CNN architecture

After some experiments and modifications, we ended up with our first custom CNN Model with 3796743 total parameters. One of the main modifications was adding dropout layers to overcome dead neurons. After the training on the FER-2013 dataset, we were able to achieve

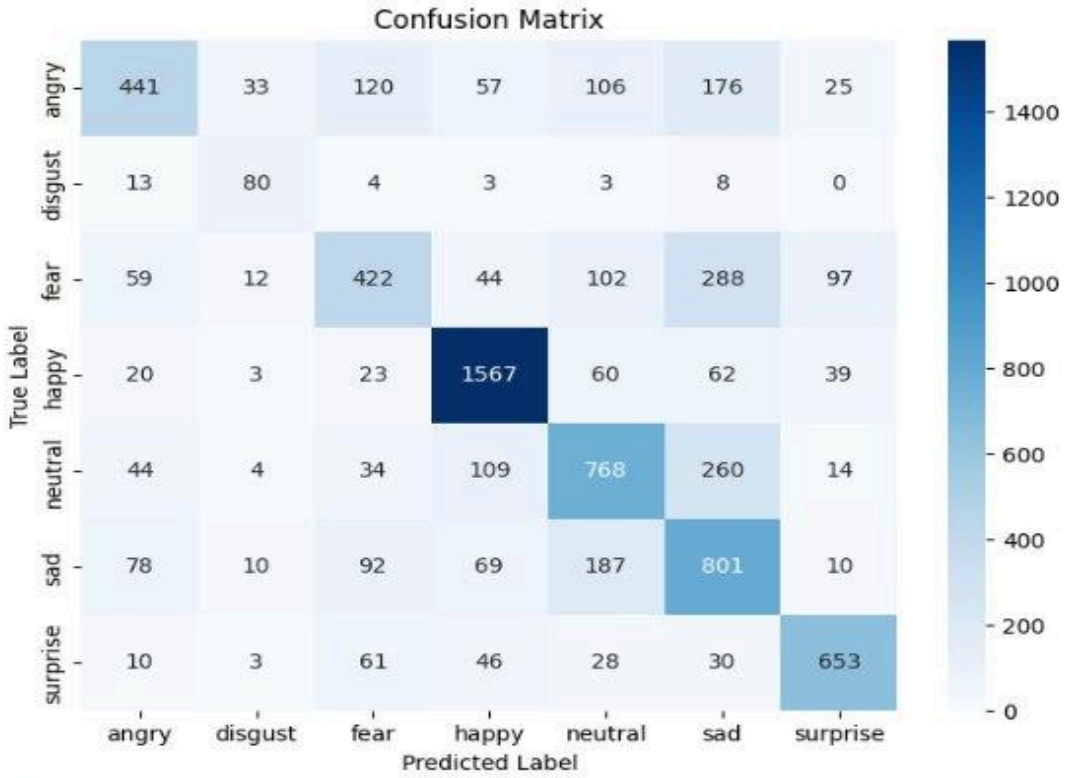
64.7% accuracy on test data after 10 epochs. To increase accuracy we tried to increase the number of parameters by increasing batch sizes of convolutional layers. The modified custom CNN model had 4763143 total parameters. This model got 66.21% accuracy on test data. However, it had a small overfitting. We thought that the reason was a vanishing problem. To overcome this problem we have changed activation functions from ReLu to Leaky ReLu. This third custom model got 66.09%, which was a little less than the previous one. In the last attempt, we decided to increase the number of epochs for this model from 10 to 25. After this modification, our accuracy went down to 65.47% on test data. So, among all of these attempts, the second attempt was the best in terms of test accuracy. It also got 66.25% test accuracy on the AffectNet dataset. You can see this model's training and test accuracies with the confusion matrix on the FER-2013 dataset in Figures #, # respectively.




```
accuracy
        training                (min:    0.432, max:    0.815, cur:    0.815)
        validation              (min:    0.221, max:    0.662, cur:    0.662)
Loss
        training                (min:    0.516, max:    1.452, cur:    0.516)
        validation              (min:    0.952, max:    1.925, cur:    0.978)

Epoch 15: saving model to model_weights_1.h5
448/448 [==============================] - 83s 185ms/step - loss: 0.5158 - accuracy: 0.8148 - val_loss: 0.9777 - val_accuracy: 0.6621 - lr: 1.0000e-05
```

Figure 11. Train and test accuracy with the loss for the best custom CNN architecture on the FER-2013 dataset (second attempt)



Classification Report:

	precision	recall	f1-score	support
0	0.66	0.46	0.54	958
1	0.55	0.72	0.62	111
2	0.56	0.41	0.47	1024
3	0.83	0.88	0.85	1774
4	0.61	0.62	0.62	1233
5	0.49	0.64	0.56	1247
6	0.78	0.79	0.78	831
accuracy			0.66	7178
macro avg	0.64	0.65	0.64	7178
weighted avg	0.66	0.66	0.65	7178

Accuracy on Test Data: 0.66

Figure 12. Confusion matrix for the best custom CNN architecture on the FER-2013 dataset (second attempt)

Our second custom CNN model is simpler, it is just a convolutional layer, batch normalization followed by max pooling and dropout. We repeat it 4 times and add a fully connected layer. The

results of it can be seen in the following figure.

Classification Report:

	precision	recall	f1-score	support
0	0.59	0.52	0.55	958
1	0.70	0.47	0.56	111
2	0.52	0.34	0.41	1024
3	0.80	0.87	0.83	1774
4	0.53	0.70	0.60	1233
5	0.52	0.47	0.50	1247
6	0.74	0.81	0.77	831
accuracy			0.64	7178
macro avg	0.63	0.60	0.60	7178
weighted avg	0.63	0.64	0.63	7178

Accuracy on Test Data: 0.64

Figure 13. Confusion matrix and test accuracy of second custom CNN model

We also trained the ViT model for image emotion recognition. We chose a pre-trained "convit_tiny.fb_in1k" model with 5710512 parameters from Timm's models. After fine-tuning this model on the FER-2013 dataset we got only 37% accuracy after 10 epochs. This result is the

same that we have tried simple pre-trained CNN models from Keras applications. That is why we created custom models for this task.

```
[Step 1] loss=2.46e+02 train accu=20.55% validation accu=22.43%
[Step 2] loss=2.15e+02 train accu=32.56% validation accu=36.00%
[Step 3] loss=2.05e+02 train accu=38.61% validation accu=26.61%
[Step 4] loss=2.01e+02 train accu=40.45% validation accu=31.96%
[Step 5] loss=1.99e+02 train accu=41.72% validation accu=37.09%
[Step 6] loss=1.98e+02 train accu=42.43% validation accu=36.34%
[Step 7] loss=1.98e+02 train accu=42.56% validation accu=34.88%
[Step 8] loss=1.97e+02 train accu=42.88% validation accu=33.42%
[Step 9] loss=1.97e+02 train accu=43.08% validation accu=30.17%
[Step 10] loss=1.98e+02 train accu=42.59% validation accu=37.83%
```

Figure 14. Loss, training accuracies, and test accuracies for each epoch after fine-tuning for the ViT model

EEG Modal Recognition

In EEG modality we only could get two datasets from the internet. The first dataset is the SEED dataset containing preprocessed EEG data in 62 frequency channels and with 3 emotion types (Ruo-Nan et al., 2013) (Zheng & Lu, 2015). The dataset weighs 7GB and has 645 data samples. The second dataset is EEG Brainwave Dataset: Feeling Emotions (https://www.kaggle.com/datasets/birdy654/eeg-brainwave-dataset-feeling-emotions) downloaded from kaggle (Bird et al., 2019). This dataset also has 3 types of emotions, however, the data is much simpler and is in CSV format. It has some extracted features and data only from 2 frequency channels, so it weighs only 12MB and has 2132 data samples. The sample codes for the second code were available and most of them used the GRU (gated recurrent unit) model and got 94.5 -98% accuracy on test data. We experimented with different models and got 98.4% accuracy on test data using the LSTM model. However, with the first dataset, we had problems, as the dataset contained huge arrays without any baseline code. Also, there was not much

available code using this dataset, the only code that we found and that actually works is this https://github.com/shivam-199/Python-Emotion-using-EEG-Signal, we used their code for feature extraction and got slightly more accuracy than they using MLP while the given code used SVM (support vector machines). The disadvantage of the given code is that it loses some time series information while extracting features, so the maximum accuracy on test data that we got was 74% while most research papers got more than 90% accuracy using advanced models for time series data. Overall, we got decent results, however, the models we trained are very unlikely to be used for all in one dataset. The reason is that EEG data is complex, and collecting and preprocessing are also complex. It is very unlikely that the data on which we trained our model would be similar to the data that we will have all in one dataset. Nevertheless, we got some experience working with the above-mentioned datasets. Below you can see the results as figures.

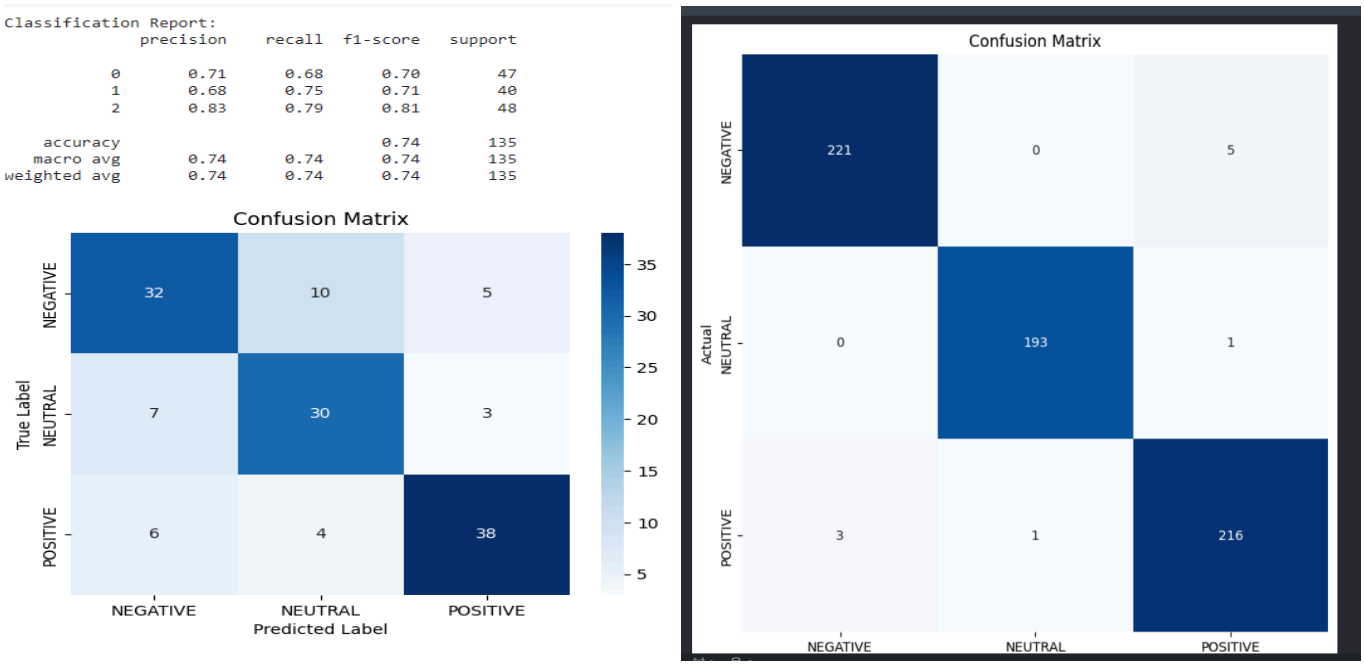
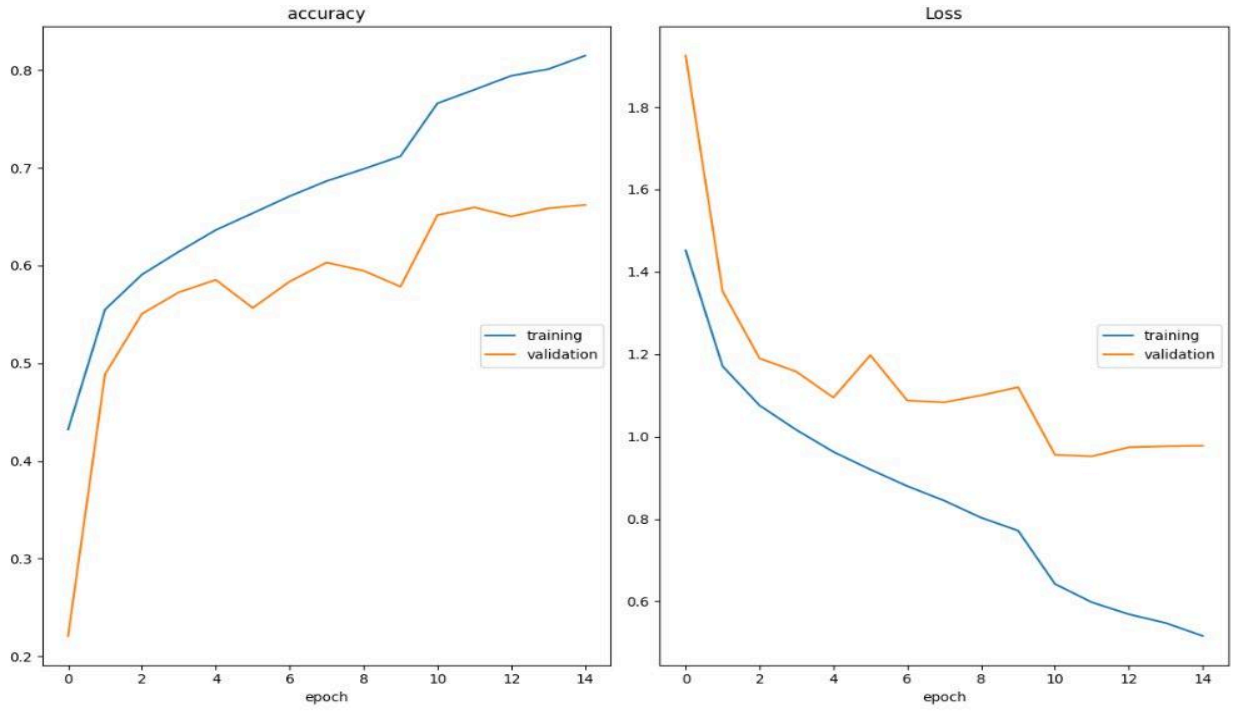


Figure 15. Results on the first and second dataset respectively EEG dataset

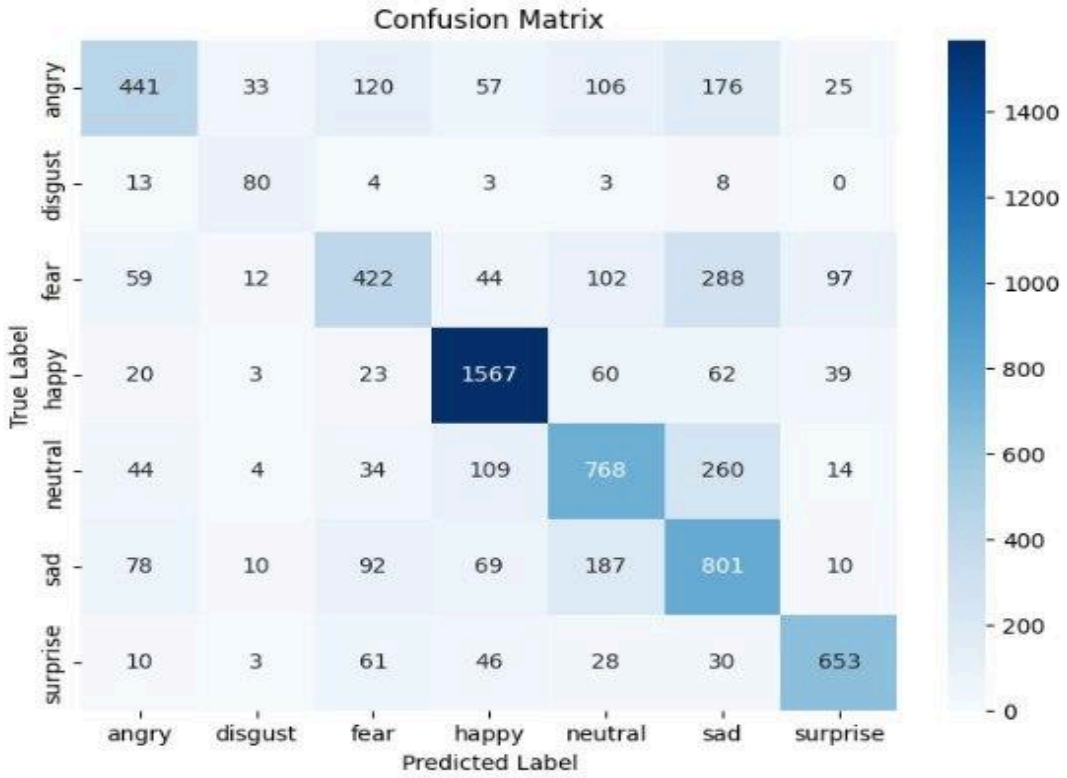
After some experimentation with batch sizes, activation functions, dropouts, and other hyperparameters, we ended up with a model that got 66.21% accuracy on test data. You can see this model's training and test accuracies with the confusion matrix on the FER-2013 dataset in Figures #, # respectively.




```
accuracy
        training                (min:    0.432, max:    0.815, cur:    0.815)
        validation              (min:    0.221, max:    0.662, cur:    0.662)
Loss
        training                (min:    0.516, max:    1.452, cur:    0.516)
        validation              (min:    0.952, max:    1.925, cur:    0.978)

Epoch 15: saving model to model_weights_1.h5
448/448 [==============================] - 83s 185ms/step - loss: 0.5158 - accuracy: 0.8148 - val_loss: 0.9777 - val_accuracy: 0.6621 - lr: 1.0000e-05
```

Figure 16. Train and test accuracy with the loss for the best custom CNN architecture on the FER-2013 dataset



Classification Report:

	precision	recall	f1-score	support
0	0.66	0.46	0.54	958
1	0.55	0.72	0.62	111
2	0.56	0.41	0.47	1024
3	0.83	0.88	0.85	1774
4	0.61	0.62	0.62	1233
5	0.49	0.64	0.56	1247
6	0.78	0.79	0.78	831
accuracy			0.66	7178
macro avg	0.64	0.65	0.64	7178
weighted avg	0.66	0.66	0.65	7178

Accuracy on Test Data: 0.66

Figure 17. Confusion matrix for the best custom CNN architecture on the FER-2013 dataset

The second approach for creating an emotion recognition model

For the time when we started working on the second approach, we got access to Professor Minho Lee's dataset. It contains 42 subjects collected by 42 actors. Each subject collects data from Audio, Video, and EEG modalities. It has five classes/emotions: Neutral, Calmness, Happiness, Sadness, and Anger. Each video, audio, and EEG data stores data with a duration of 21 seconds. We will use this data as a base for our multimodal emotion recognition model since it meets our requirements for this project.

Audio-Video Modal Recognition

For the Audio-Video emotion recognition model in the second approach, we used a model architecture created by Chumachenko, Iosifidis, and Gabbouj (2022) for the RAVDESS dataset. We chose the best model out of the three given which was the intermediate attention model with 1 number of heads. As mentioned in related work, this model has good accuracy for the RAVDESS dataset, also they shared code for training in GitHub. So, we took their code, changed it to fix bugs related to library versions, and trained an audio-video model fused by using an intermediate attention mechanism. We get the same results on test data as they got in the paper which is 81.58%. Initially, we trained the model for all 8 classes, however, Professor Minho's dataset has only 5 classes, so retrained it for 5 classes only, the accuracy of the model improved due to easier tasks. Our model had a test accuracy of 92% on 5 classes on the RAVDESS dataset. The input for the model is 3.6-second audio and video or just video with audio.

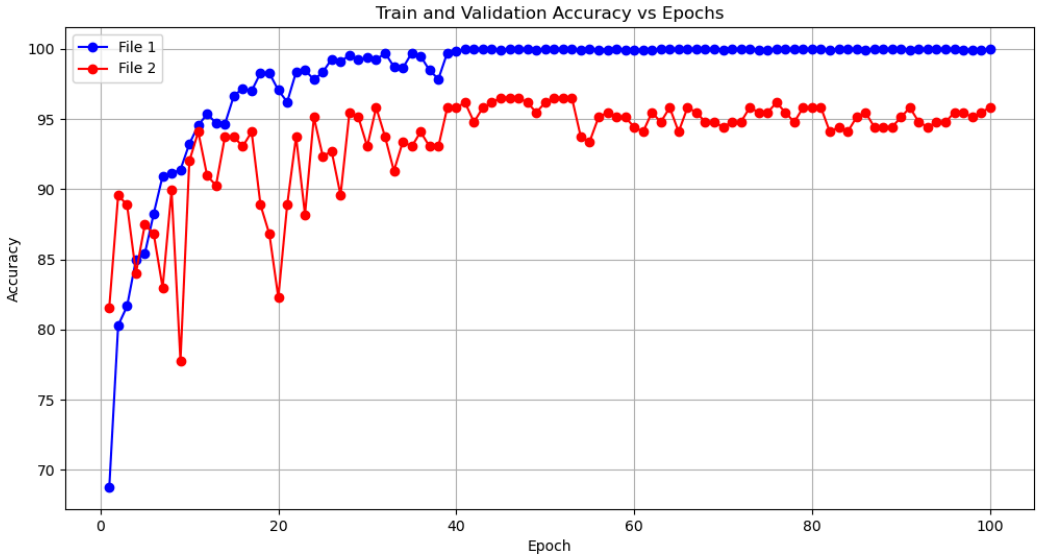


Figure 18. Train and Validation accuracies of audio-video model on RAVDESS with 5 classes

Then we gained access to Professor Minho Lee's dataset. Our task was to adapt this model to the dataset given to us by the professor, which met all our requirements, and train it. We have divided the Dataset into training (from 1 to 30 subjects), validation (from 31 to 36 subjects), and testing (from 37 to 42 subjects). Each video and audio lasted about 21 seconds. However, the model that we are going to adapt to this dataset accepts video and audio in 3.6 seconds. Moreover, 21 seconds is too long a period to identify emotions, because, in reality, a person can show several emotions within a couple of seconds. For these reasons, instead of adjusting the model to the dataset, we decided to adjust the dataset to the model. To accomplish this task, we divided each video and audio into six parts of 3.6 seconds each. After that, we trained two models. One from scratch and another from transfer learning and fine-tuning a pre-trained model on RAVDESS.

After training the model from scratch for 100 epochs we got a result of 61% on validation and 58% on the test. Below we have provided a graph of the accuracy of the model as the model evolves. Blue line for training accuracy, and red line for validation accuracy. The same model trained on RAVDESS and tested on RAVDESS has an accuracy of more than 90%. The reason for such difference is that in the RAVDESS dataset, each actor really tried to show emotion as much as possible. While in Professor Minho's dataset, actors didn't try to maximally express their emotions.

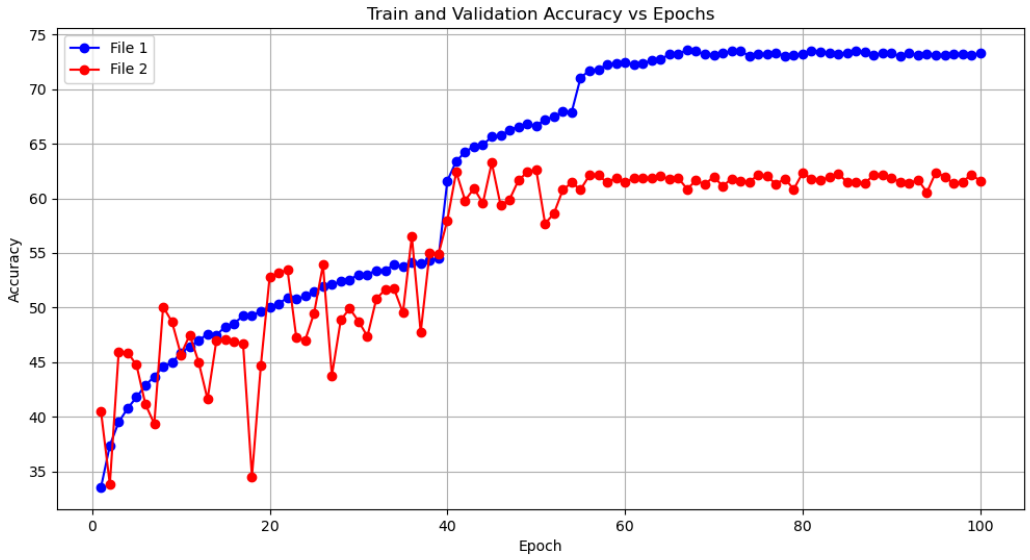


Figure 19. Training and validation accuracy on the full dataset

In the model configurations, we use different types of optimizations such as Dropout, Regularization, Cross Validation, Noise, and so on. However, given such a large number of different techniques that prevent overfitting and a small number of parameters (approximately 1850000), you can see some overfitting. The reason for this phenomenon is the cross-validation technique that we used. In short, we used different subjects for training, validation, and testing. Thus, our model was never familiar with the actors during the validation and testing. To prove

our argument, we will provide the results of training and tests of models trained on 5 subjects of the same dataset. In the first case, we divided the dataset by subjects and added 1,2,3 subjects for training, 4 subjects for validation, and 5 subjects for testing. In the second case, we divided each subject into 60% for training, 20% for validation and 20% for testing. Overall we have a similar amount of data for training, validation, and testing.

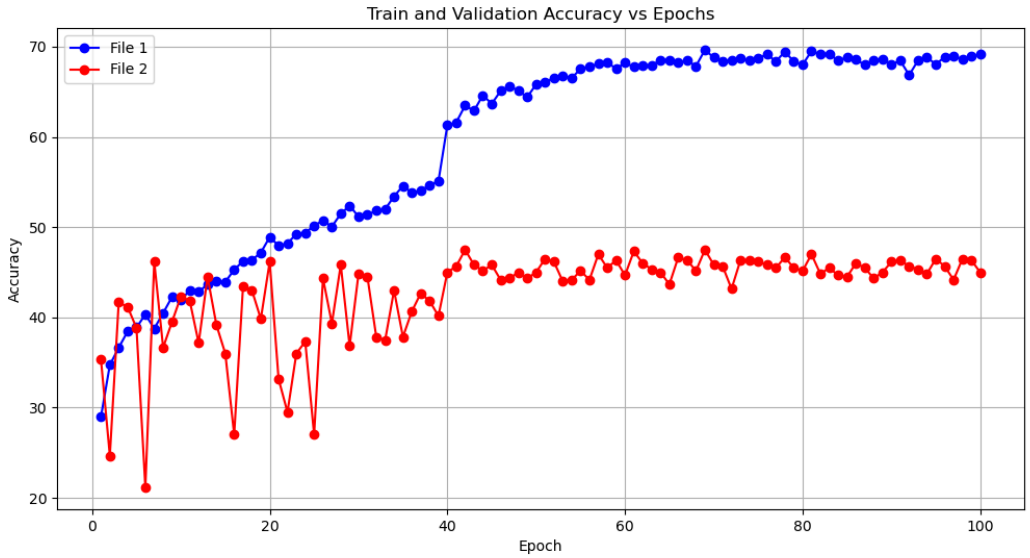


Figure 20. Training and validation accuracy on five subjects using the first Cross-Validation technique

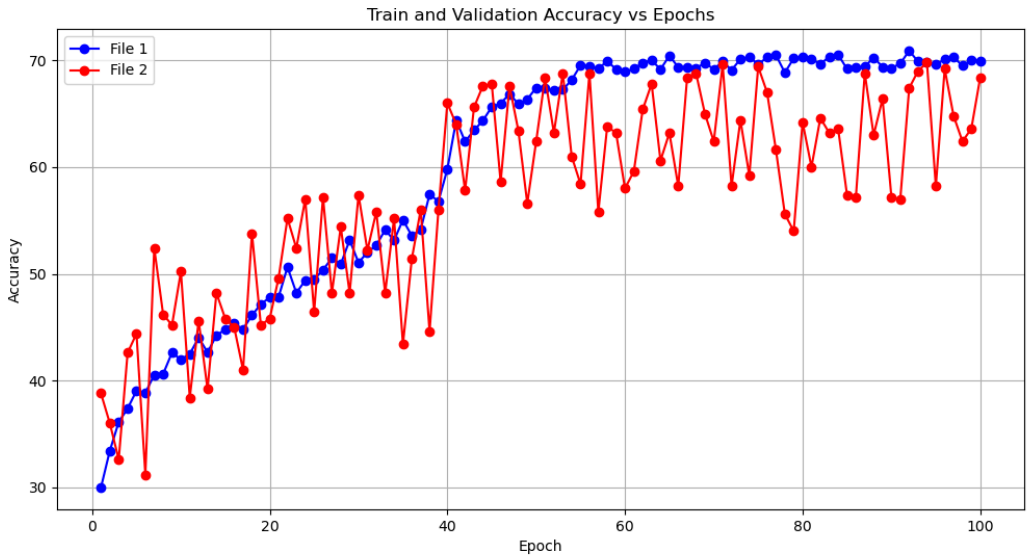


Figure 21. Training and validation accuracy on five subjects using the second Cross-Validation technique

As you can see on the graphs, in the first case, the model shows overfitting, when the second one does not have it. We decided to use the first method for cross-validation, since in reality, our model will not be familiar to the users of our site, so we can more accurately determine the accuracy of our model.

After training the second model, which uses a pre-trained model trained on 5 classes of the RAVDESS dataset by transfer learning and fine-tuning it, we got unsatisfactory results. So, we stuck with the model that was trained only on Professor Minho's dataset.

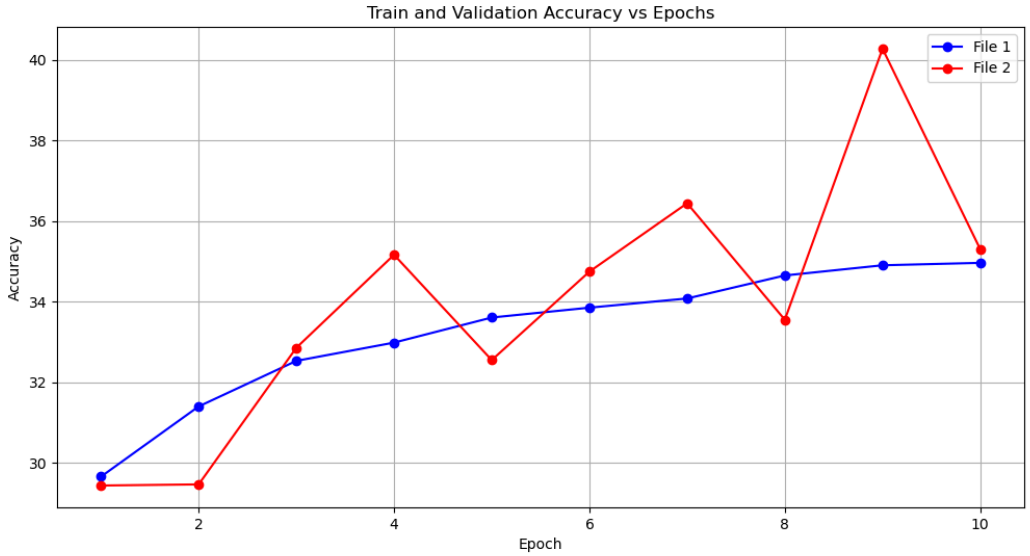


Figure 22. Training and validation accuracy on full dataset after transfer learning

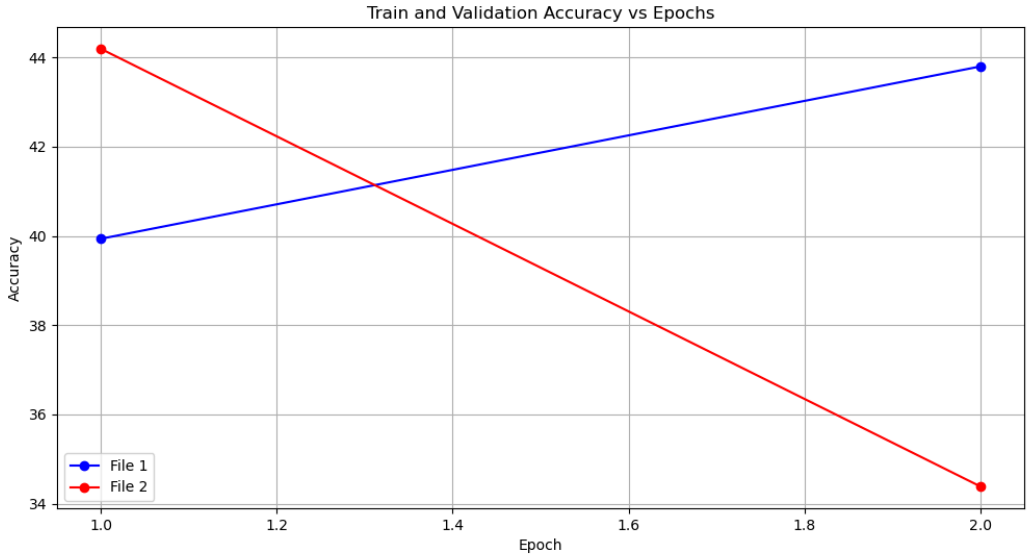


Figure 23. Training and validation accuracy on full dataset after fine-tuning

EEG Modal Recognition

Professor Minho provided us with not only the dataset itself but also the preprocessing code for EEG, audio, and video separately. We only used his code for EEG, as our EEG models trained in the last semester were trained on different EEG data which was preprocessed differently, so we couldn't reuse our older models and codes. So, we decided to use Professor Minho's EEG model. We slightly changed the code, as in the original code, the model was training only for 1 subject. So, we trained the EEG model on 30 subjects, and used 6 subjects as validation and the last 6 subjects as test data. So, the training accuracy was 59% while validation was around 39%. The reason for the big difference is that EEG data is subject-dependent. Meaning that each subject has its patterns of brain activity for each emotion. As we used totally different subjects for training and validation, we have this difference.

One thing to note is that the EEG model makes 4 predictions during the 21 seconds of data recording. So, the EEG model makes 4 predictions for 1 data sample. In most of the cases, we added the logits of all 4 predictions and divided them by 4, taking the average.

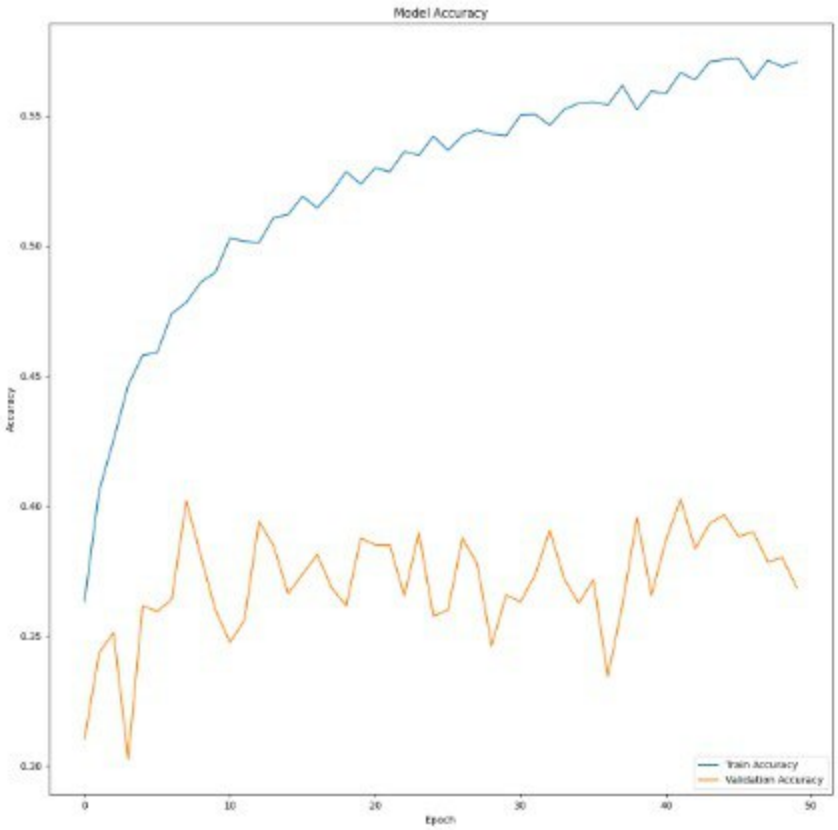


Figure 24. Training and Validation accuracies of EEG model

Fusion Mechanism

Our first fusion method is a WAVER late fusion technique that we read about in the paper written by Wang Q., Wang M., Yang Y., and Zhang X. in 2022. As we said before each video-audio was divided into 6 parts and EEG data was divided into 4 parts. That is why before applying this method to our models we need to sum the outputs from the Dense layer and normalize them. After this process, we get a 1 by 5 array from Audio-Video and 1 by 5 from EEG. Then we take the validation accuracies from both models and multiply these arrays to their respective validation accuracy. In the end, we sum up arrays and apply argmax() function to take the index of the greatest value. This index denotes the class of the prediction.

	precision	recall	f1-score	support
Neutral	0.80	0.74	0.77	120
Sadness	0.62	0.87	0.72	120
Anger	0.89	0.72	0.79	120
Happiness	0.64	0.74	0.69	120
Calmness	0.74	0.52	0.61	120
accuracy			0.72	600
macro avg	0.74	0.72	0.72	600
weighted avg	0.74	0.72	0.72	600

Figure 25. Results of weighted fusion on validation data.

	precision	recall	f1-score	support
Neutral	0.90	0.46	0.61	120
Sadness	0.70	0.75	0.73	120
Anger	0.60	0.90	0.72	120
Happiness	0.49	0.68	0.57	120
Calmness	0.79	0.45	0.57	120
accuracy			0.65	600
macro avg	0.70	0.65	0.64	600
weighted avg	0.70	0.65	0.64	600

Figure 26. Results of weighted fusion on test data.

The second fusion method that we used is similar to the previous one, however, we took the probabilities of the audio-video and EEG models and fed them to the boosting tree. The reason why we chose this approach is that our first approach only accounts for final probabilities. What if the probability for some class in one model is the highest but not higher enough than others, so summing the outputs with another model results in a different answer that is incorrect. In this case, our first fusion method can do nothing to retrain. In our second approach, the boosting tree could learn some patterns and improve results. We trained our boosting tree xgboost on validation data and tested it on the test data. So, our boosting tree found some patterns for validation subjects, but we tested the results on the test subjects that our models

didn’t see. The reason why we chose the boosting tree is that they are very popular right now, and they are highly used in state-of-the-art production models.

	precision	recall	f1-score	support
Neutral	0.93	0.68	0.79	120
Sadness	0.74	0.70	0.72	120
Anger	0.51	1.00	0.68	120
Happiness	0.67	0.41	0.51	120
Calmness	0.77	0.57	0.66	120
accuracy			0.67	600
macro avg	0.72	0.67	0.67	600
weighted avg	0.72	0.67	0.67	600

Figure 27. Results of xgb fusion on test data

In the third fusion method, we tried to make some earlier fusion. However, our fused audio-video model’s last layers were fully connected layers and attention layers. We couldn’t use the attention layer as it gives weights to the audio and video block of the model, so without them, the attention layers couldn’t be considered as valid features. Also, we couldn’t add an EEG model to the attention block as it has different lengths of data, while audio video consists of 3.6 seconds, the EEG data is around 5.2 seconds. So, we decided to take the layers that are after the attention block. After the attention block, we only have a fully connected layer and a dense layer. We didn’t choose fully connected layers because they are flat, and we could train only MLP on this flat data, but MLP doesn’t account for time and spatial data. So, we decided to choose dense layers from both the audio-video model and EEG model and train a new CNN model as a fusion. For each 21-second video, we have 6 predictions from the audio-video model and 4 predictions from the EEG model, so in total we have 10 predictions each with 5 logits for each class. So, the input for our CNN model is a 10x5 numpy array. We trained this CNN model on validation data

only and tested it on test data. The model summary and the training and testing accuracies could be seen in the figures below.

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 9, 4, 32)	160
max_pooling2d (MaxPooling2D)	(None, 4, 4, 32)	0
dropout (Dropout)	(None, 4, 4, 32)	0
conv2d_1 (Conv2D)	(None, 2, 2, 64)	18,496
max_pooling2d_1 (MaxPooling2D)	(None, 1, 2, 64)	0
dropout_1 (Dropout)	(None, 1, 2, 64)	0
flatten (Flatten)	(None, 128)	0
dense (Dense)	(None, 64)	8,256
dropout_2 (Dropout)	(None, 64)	0
dense_1 (Dense)	(None, 5)	325

Total params: 27,237 (106.39 KB)

Figure 28. Third fusion CNN model summary

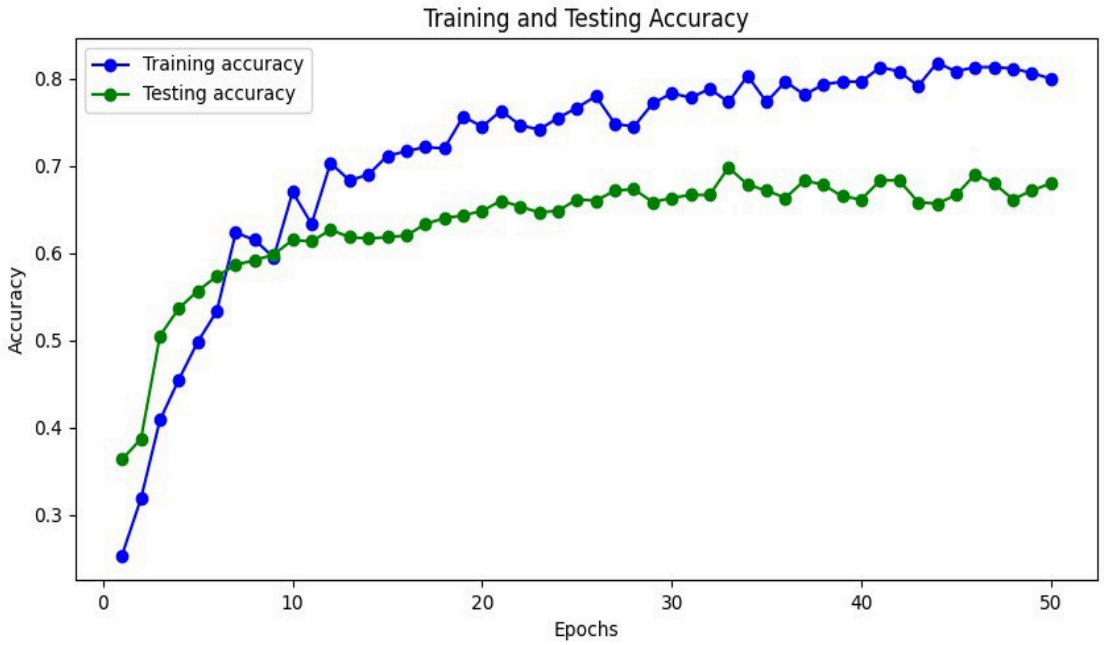


Figure 29. Training and testing accuracies of CNN fusion model.

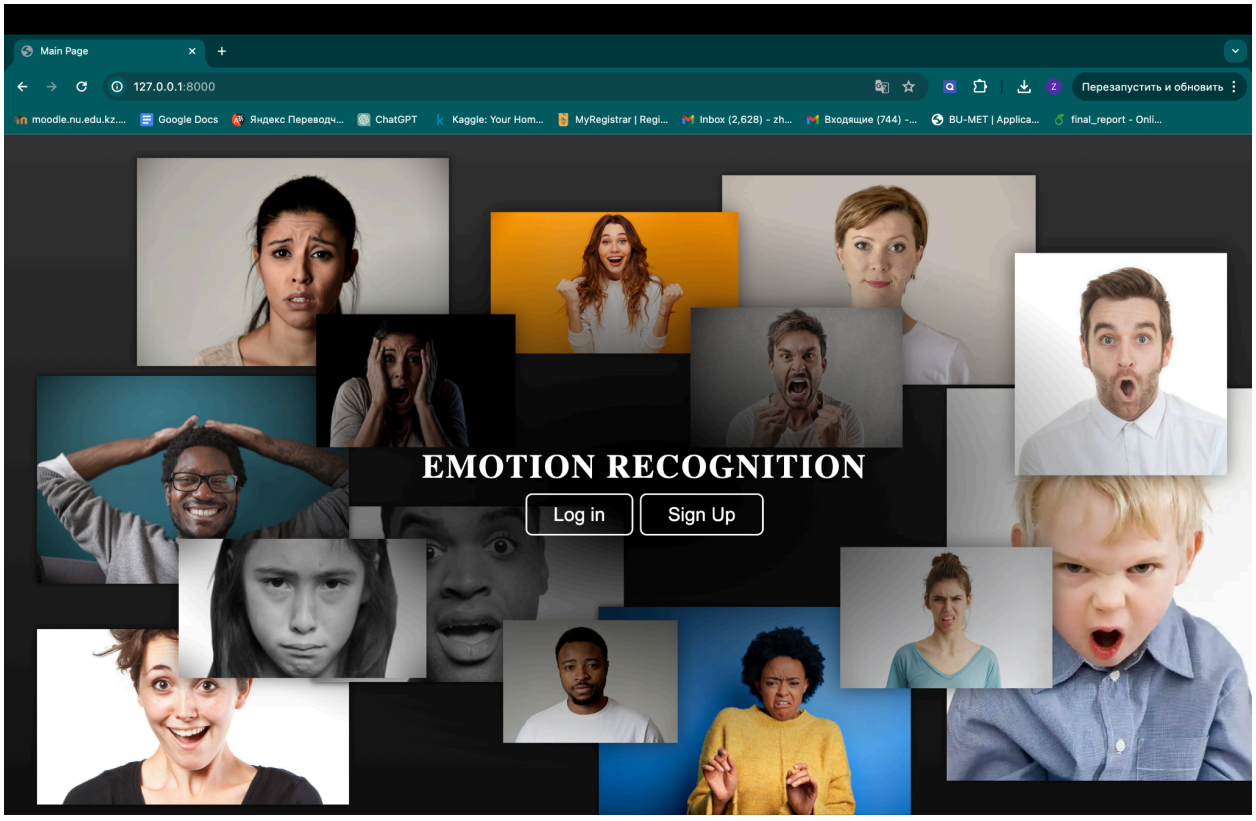
	precision	recall	f1-score	support
Neutral	0.98	0.71	0.82	120
Sadness	0.71	0.76	0.73	120
Anger	0.52	0.94	0.67	120
Happiness	0.62	0.53	0.57	120
Calmness	0.90	0.47	0.62	120
accuracy			0.68	600
macro avg	0.74	0.68	0.68	600
weighted avg	0.74	0.68	0.68	600

Figure 30. Results of CNN fusion on test data

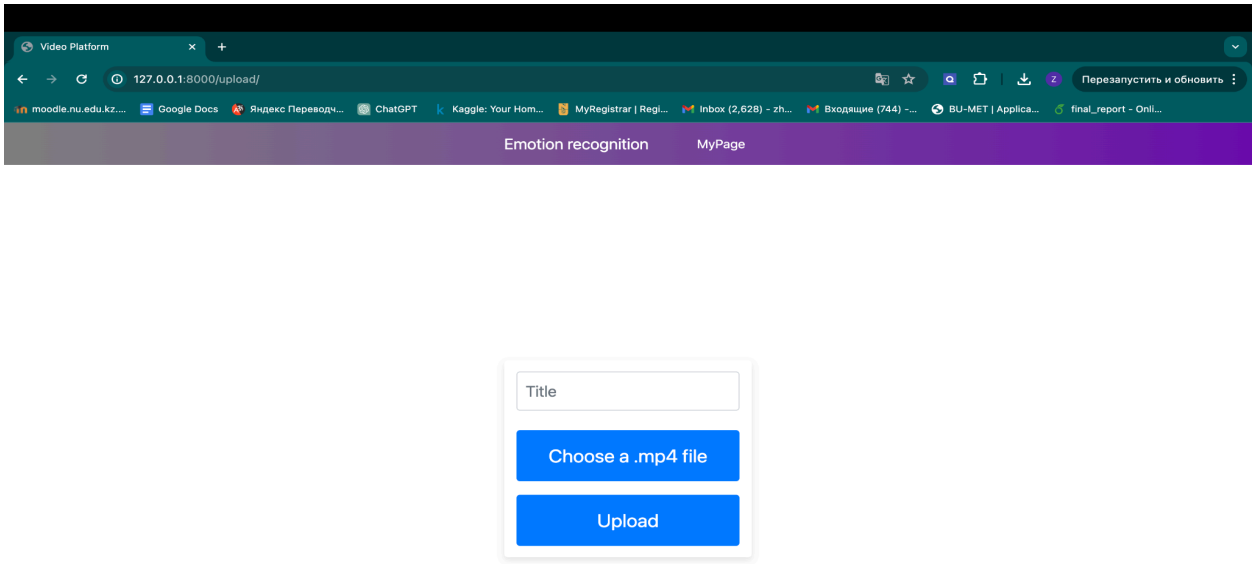
Web Application

The website was created using the Django framework. Currently, there are 5 pages available on the site: a login page, 2 upload pages, depending on whether there is only audio-visual input or audio-visual data, EEG data included, and 2 pages with a video player with

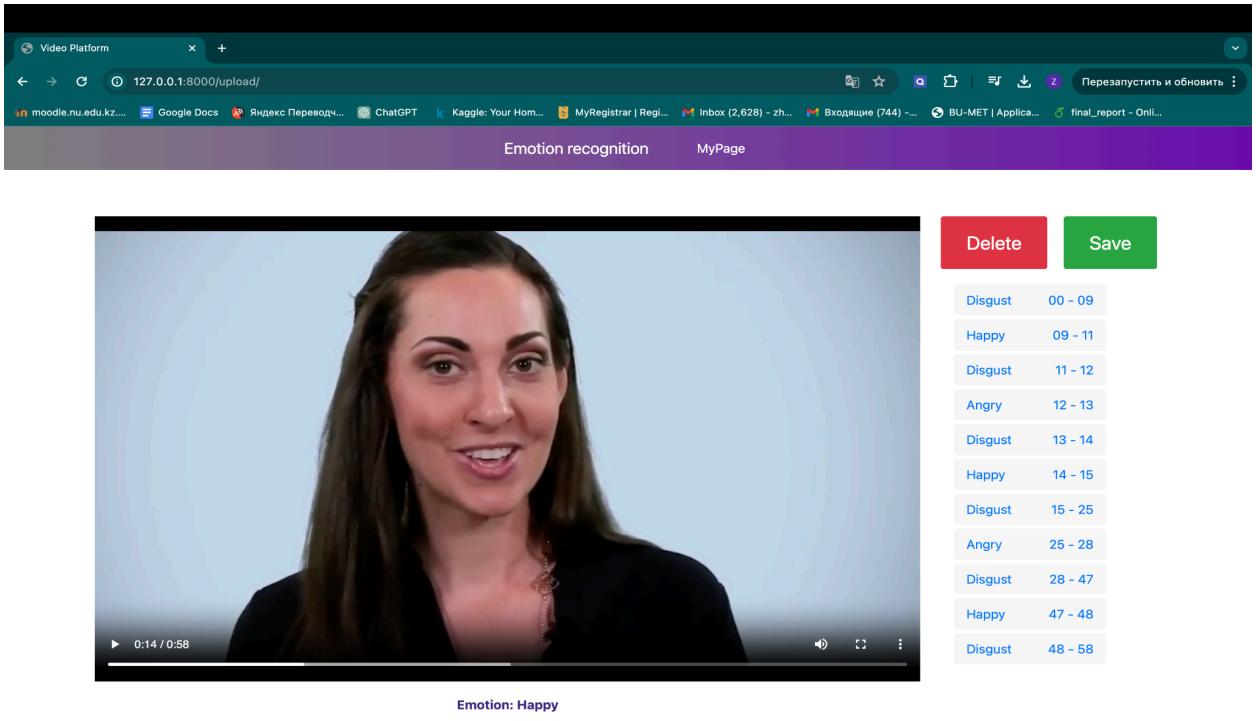
emotion labels. The reason we created 2 upload pages and 2 video player pages is because datasets that include data from 3 modalities are unavailable. On one of the pages with the player, when data is entered only from audio and video, a list of emotions that were expressed in the video and the corresponding temporary links will be available, which you can rewind by simply clicking on this label. On the second upload page, you will need 3 files to download: .mp4, .wav and .mat. We created this page specifically because we can test our model on Professor Minho Lee's dataset only. Accordingly, the second page with the player will show only one emotion, since for testing we upload only one video, audio and EEG data from the dataset of professor Minho Lee.



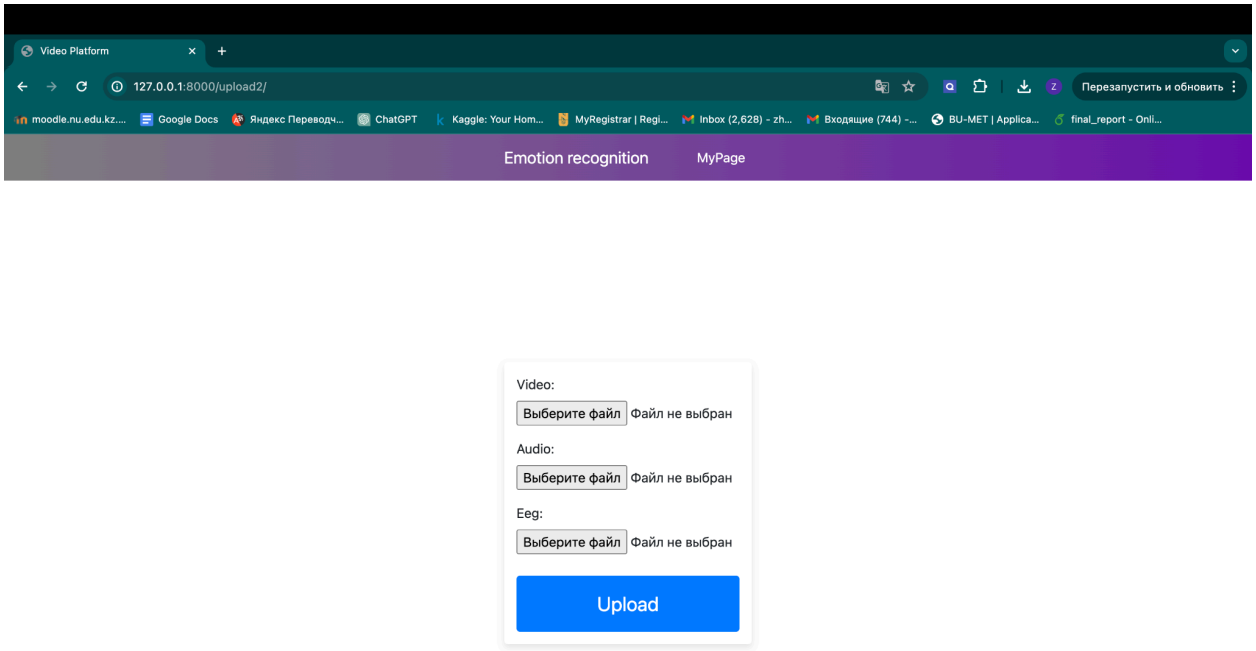
Screenshot 1. Login page



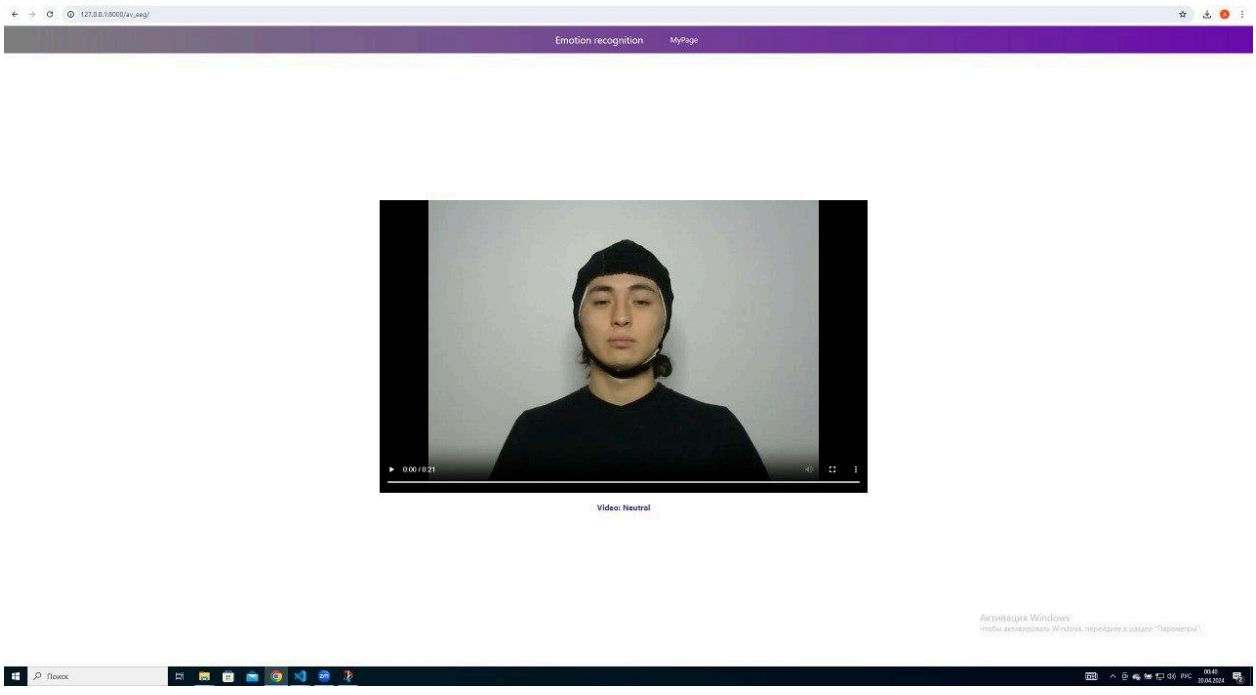
Screenshot 2. Upload page for audio-video model



Screenshot 3. Uploaded video with results and links.



Screenshot 4. Upload page for audio-video and EEG model



Screenshot 5. Uploaded video with results

5. Project Execution

Initially, it was planned to create a website that could predict emotions in real-time, however, due to the features of our trained model, especially prediction time that can lead to slow work of the website, we decided to change it to a website where the user can upload a video and the EEG data. At the moment, the user can upload input data to the site and after a while, get to a page where there is a video player, a list of emotions, and corresponding time links.

There were constant problems with training the model in terms of time since the devices we used every day were not suitable for training the model. In the fall semester, we used Jupyter Notebook-hosted services such as Colab and Kaggle. Since we didn't have a complete dataset in the fall semester, we could work on these platforms. However, there were still problems when using these platforms, such as time limits, quotas, and queues on these platforms. At the beginning of the spring semester, we got access to a new computer specially designed for data processing with powerful characteristics. Only after installing the operating system, drivers, IDE, and special libraries we were ready to cope with large, complete datasets.

Last semester, we trained various models specifically for late fusion, but our image prediction capabilities for further video prediction were limited because our image prediction model was trained based on parts of FER-2013 and AffectNet datasets. Since we couldn't access the full AffectNet and FER-2013 dataset, we started looking for a ready-made model that was already trained on the complete dataset. That is why we decided to use a third-party model created by Chumachenko, Iosifidis, and Gabbouj (2022).

Another problem is that we did not have a suitable dataset for our project. To be more precise, there was no dataset that includes data from all the modalities. Everything was resolved when Professor Minho Lee shared the dataset he had collected. After that, we started adapting

the dataset to our code. The dataset consisted of 42 subjects and included audio, video, and EEG data. We trained a model with intermediate attention to video and audio for fusion and then added an EEG in late fusion. We tried 3 different techniques for fusion. All the processes were described in the previous section.

6. Evaluation

Our main evaluation method was of course accuracy score and f1-score for unbalanced datasets. Whether we solved the problem mentioned in the introduction really depends on the metrics on different datasets as we tried different approaches for models and for users also. Below we will provide evaluation results on test splits of different datasets, including FER-2013, RAVDESS, and Professor Minho's dataset. The FER-2013 has a predefined split for test data, so we tested our model on this test split. While RAVDESS and Professor Minho's datasets don't have predefined splits. When training and testing on the RAVDESS dataset, we used the same splits as the Chumachenko (2022) paper. They trained on 16 subjects and used 4 subjects as validation and the other 4 subjects as test data. So, we used the same splits and evaluated our model on test subjects which our model didn't see in training and validation processes. As we mentioned before when training on Professor Minho's dataset, we used 30 subjects for training, 6 for validation, and 6 for testing. So, we evaluated our models on test subjects that our models didn't see.

The first is an evaluation of our custom CNN model trained on the FER-2013 dataset. We mentioned our results, where our best model got an accuracy rate of 66.21% along with results of other papers earlier. Overall, we made a decent model that outperformed some mentioned models, however, it is not the best model among all papers.

Our end model is a fusion model of our audio-video model and EEG model. Our audio-video model trained on the RAVDESS dataset got a test accuracy of 85.24% on 8 classes of the RAVESS dataset. Our audio-video model trained on the RAVDESS dataset got a test accuracy rate of 92% on 5 classes of the RAVDESS dataset. Our audio-video model fine-tuned on Professor Minho's dataset got a test accuracy of 40.5% on test data. This was not satisfactory, so we decided to train the model on Professor Minho's dataset only, to see whether we get better results or not. As mentioned above, this model has a test accuracy of 58.36% on Minho's dataset. Which is better than our fine-tuned model. In our opinion, this test result was satisfactory as subjects in Professor Minho's dataset didn't obviously express their emotions, so it was hard to classify emotions even for us humans. Also, as mentioned above, if we shuffled the data of all subjects for training, validation, and testing we could get much higher test results. However, we didn't see it as fair, as the user that is going to use our web application is an unseen subject. So, we stick to the plan of using different subjects for different splits, so the test subjects are unseen data for our model just like in real-life scenarios. So, we got the audio-video model. Then as mentioned above in the EEG modality section, we trained the EEG model on 30 subjects and got test accuracy of 39.42%. Which is fine, as we used professor code for training, and we mentioned the problem of EEG data earlier. So this result was fine for us. Then we made 3 fusion methods described in the above fusion section. The first fusion method got a validation accuracy of 72% and test accuracy of 65%. This is quite strange as our model didn't see both of the splits, but it got higher results for validation data. The possible reason for that is that subjects in the validation split could be more similar to the training subjects than subjects in the test split. Nevertheless, the test result of the fused model is better than the test results of each model separately. Our next fusion, the xgboost tree trained on validation data, got test accuracy of

67.33%. Which is better than the first fusion method. Our last fusion method, CNN model trained on validation data, got test accuracy of 68.33%. So, the third fusion method was the best in terms of evaluation. So, we chose this fusion method for our web application.

7. Conclusion and possible future work

This project has made significant strides in multimodal emotion recognition by integrating audio, video, and EEG data to enhance the accuracy of emotion detection systems. Throughout the research, the team successfully developed models that leverage the strengths of each individual modality and employed advanced fusion techniques to create a robust system capable of recognizing emotions with a high degree of accuracy. The results underscore the potential of using a hybrid fusion approach to improve emotion recognition systems, particularly in complex scenarios where traditional single-modality methods may fall short. In conclusion, we investigated and tried different approaches to solve the problem of multimodal emotion recognition using fusion mechanisms. We trained the model for each modality separately and tried late fusion. Also, we trained a fused audio-video model and fused EEG model to it using different late fusion techniques. During the process, we tried different deep learning architectures, different datasets. The notable models that we trained are complex CNN model trained on the FER-2013 dataset, which got accuracy of 66.21% on test data. The next is audio-video fused models with attention mechanisms trained on RAVDESS and professor Minho's datasets. Next, we made different fusion techniques for EEG and audio-video modalities. We made weighted fusion on probabilities of classes, we made a boosting tree classifier trained on logits of models and finally a CNN model trained on logits of models.

Eventually, the CNN fusion was the best in terms of test data accuracy. Additionally, we made a web application, with 2 different modes. In the first mode, users could upload a video, and then get a list of predicted emotions and their timings. Also, users could download a modified version of the video, where in each frame the predicted emotion label is written at the top left corner of the screen. The second mode is made specifically for professor Minho's dataset, where users could upload data for all 3 modalities and get the final prediction.

The possible future work is to try to improve the accuracy of the model by trying other fusion mechanisms for the EEG model. One could use an attention mechanism with all 3 modalities together. Additionally, one could make a web application where users could try to use different models for emotion recognition, for example, lightweight real-time recognition based only on 1 video frame or complex fused model which require some time but could provide better results. The other future improvements could be to collect more data and train better models on a higher number of data samples.

8. References:

Agrawal, A., & Mittal, N. (2020). Using CNN for facial expression recognition: a study of the effects of kernel size and number of filters on accuracy. The Visual Computer, 36(2), 405-412.

Arriaga, O., Valdenegro-Toro, M., & Plöger, P. (2017). Real-time convolutional neural networks for emotion and gender classification. arXiv preprint arXiv:1710.07557.

Bird, J. J., Faria, D. R., Manso, L. J., Ekárt, A., & Buckingham, C. D. (2019). A deep evolutionary approach to bioinspired classifier optimisation for brain-machine interaction. Complexity, 2019.

Chen, J., Ro, T., & Zhu, Z. (2022). Emotion recognition with audio, video, eeg, and emg: a dataset and baseline approaches. IEEE Access, 10, 13229-13242.

Duan, R. N., Zhu, J. Y., & Lu, B. L. (2013, November). Differential entropy feature for EEG-based emotion classification. In 2013 6th International IEEE/EMBS Conference on Neural Engineering (NER) (pp. 81-84). IEEE.

K. Chumachenko, A. Iosifidis and M. Gabbouj, "Self-attention fusion for audiovisual emotion recognition with incomplete data," in 2022 26th International Conference on Pattern Recognition (ICPR), Montreal, QC, Canada, 2022 pp. 2822-2828. doi: 10.1109/ICPR56361.2022.9956592

Kumar, C. A., Maharana, A. D., Krishnan, S. M., Hanuma, S. S. S., Lal, G. J., & Ravi, V. (2022, December). Speech Emotion Recognition Using CNN-LSTM and Vision Transformer. In International Conference on Innovations in Bio-Inspired Computing and Applications (pp. 86-97). Cham: Springer Nature Switzerland.

Kusuma, G. P., Jonathan, J., & Lim, A. P. (2020). Emotion recognition on fer-2013 face images using fine-tuned vgg-16. Advances in Science, Technology and Engineering Systems Journal, 5(6), 315-322.

Minaee, S., Minaei, M., & Abdolrashidi, A. (2021). Deep-emotion: Facial expression recognition using attentional convolutional network. Sensors, 21(9), 3046.

Mollahosseini, A., Chan, D., & Mahoor, M. H. (2016, March). Going deeper in facial expression recognition using deep neural networks. In 2016 IEEE Winter conference on applications of computer vision (WACV) (pp. 1-10). IEEE.

Qian Wang, Mou Wang, Yan Yang, Xiaolei Zhang, Multi-modal emotion recognition using EEG and speech signals, Computers in Biology and Medicine, Volume 149, 2022, 105907, ISSN 0010-4825, https://doi.org/10.1016/j.compbiomed.2022.105907.

Quinn, M., Sivesind, G., Reis, G., "Real-time Emotion Recognition From Facial Expressions," 2017.

Soleymani, M., Lichtenauer, J., Pun, T., & Pantic, M. (2011). A multimodal database for affect recognition and implicit tagging. IEEE transactions on affective computing, 3(1), 42-55.

Talegaonkar, I., Joshi, K., Valunj, S., Kohok, R., & Kulkarni, A. (2019, May). Real time facial expression recognition using deep learning. In Proceedings of international conference on communication and information processing (ICCIP).

Tang, Y. (2013). Deep learning using linear support vector machines. arXiv preprint arXiv:1306.0239.

Venkataramanan, K., & Rajamohan, H. R. (2019). Emotion recognition from speech. arXiv preprint arXiv:1912.10458.

Zheng, W. L., & Lu, B. L. (2015). Investigating critical frequency bands and channels for EEG-based emotion recognition with deep neural networks. IEEE Transactions on autonomous mental development, 7(3), 162-175.