



Audio-Video – Active Speaker Localization using Deep Learning

Elnatan Kleinman

Roy Dovrat

B.Sc. Graduation project- Computer Engineering

Professional Advisor: Amit Eliav

Academic Advisor: Prof. Sharon Gannot

Abstract	3
Acknowledgment	4
1. Introduction:	
1.1 Deep Learning	5
1.2 Neural network	6
1.3 convolutional neural networks.....	7
Audio-Video – Active Speaker Localization:	
1.4 The Problem	9
1.5 Our solving approach.....	10
1.6 Short-Time Fourier Transform (STFT)	10
1.7 Alternative Solutions	12
2. The Dataset:	
2.1 EasyCom Dataset	13
2.2 Our Dataset	14
2.3 Audio Preprocessing	14
2.4 Video Preprocessing	15
2.5 Data Splitting	
2.5.1 Train	17
2.5.2 Validation	17
2.5.3 Test	18
3. The model:	
3.1 Architecture	
3.1.1 U-Net Network	19
3.1.2 ResNet (Residual Network)	20
3.1.3 Our Combination	20
3.2 metrics	23
3.3 Additional ways to handle the unbalanced data.....	30
Model Block Diagram	32
4. The Results:	
4.1 Wearer voice activity detection task	34
4.2 Voice activity in frame segmentation task	35
5. Conclusions and Summary	40
6. possible future projects	43
7. Bibliography	44

Abstract:

Augmented reality (AR) devices have the potential to improve human perception and assist in complex social interactions [1]. These devices can offer various helpful features, but one key challenge lies in effectively understanding and processing the audio and visual context of these interactions. The challenge is detecting and localizing the voices of the people around them.

Augmented reality devices are technologies that overlay digital information or virtual elements onto the real world. These devices have the capability to enhance how humans perceive and engage with their surroundings. In particular, they can provide assistance in intricate conversational scenarios, making interactions more meaningful and informative.

To make the most of AR in social situations, it's important to accurately capture the audio and visual context. This context involves recognizing who is speaking and where they are in the environment. However, this task is complicated due to several factors:

- **Egocentric Nature:** AR devices are typically worn on a person's head, resulting in movements that might cause blurriness in the captured video.
- **Viewing Angle Issues:** People around the wearer might be captured from challenging angles, making it difficult to identify them accurately.
- **Obstructions and Clutter:** Visual obstructions, such as objects or other people, can hinder clear views of individuals in the environment.
- **Audio Complexity:** Background noise and other audio disturbances can complicate the task of isolating and understanding voices.
- **Lighting Conditions:** Poor lighting can further impact the quality of captured video and make it harder to distinguish people and their actions.

In order to address these challenges, we will propose a deep learning approach, resistant to noise, that uses video and a multi-channel microphone array to identify the active speakers in the video obtained by running on audio and video. There are methods for locating a speaker using neural networks on video. And there are those who work on audio, we are looking at different methods to combine in order to get better results.

Acknowledgment:

We are profoundly grateful to extend our heartfelt appreciation to the individuals who have played pivotal roles in the success of this project. Their unwavering support, guidance, and expertise have been instrumental in shaping this endeavor into what it is today.

Foremost, we express our deepest gratitude to our professional academic advisor, Mr. Amit Eliav. His presence throughout the project offers us encouragement and insights. His timely advice, constant guidance, and willingness to extend a helping hand in overcoming any problems have been instrumental in steering us towards success.

In the same breath, we extend our sincere thanks to Professor Sharon Ganot, our academic supervisor. His guidance has been instrumental in providing us with the necessary framework to execute our project with precision.

Additionally, our gratitude extends to Mr. Pinchas Tandaytnik for his sage advice and technical expertise that proved to be invaluable during the project's course.

1. Introduction:

1.1 Deep Learning:

To understand the background of deep learning, it's important to first gain a comprehensive perspective on artificial intelligence (AI), machine learning (ML), and then explore deep learning (DL).

Artificial Intelligence (AI): is a field in computer science that tries to make machines do things that usually need human thinking.

AI uses many ways to work, like rules, expert advice, and math. Its main goal is to make smart machines that can think, learn, understand their surroundings, and make choices.

Machine Learning (ML): is a subset of AI that focuses on developing algorithms and models that enable computers to improve their performance on a specific task through learning from data. In ML, models are trained on data to recognize patterns, make predictions, or optimize a specific objective without being explicitly programmed.

ML techniques include supervised learning, unsupervised learning and reinforcement learning.

Deep Learning (DL):^[4] is a type of machine learning, specifically a neural network with three or more layers. It tries to imitate how the human brain works, learning from lots of data. More layers make it better at making accurate predictions. Deep learning powers AI applications that automate tasks and improve services, from digital assistants to self-driving cars.

Deep learning and machine learning differ in how they handle data and learn. In machine learning, structured, labeled data is used for predictions, and features are organized into tables. It can handle unstructured data but usually requires preprocessing to structure it.

Deep learning, on the other hand, can work with unstructured data like text and images without much preprocessing. It automates feature extraction, reducing the need for human intervention. Deep learning algorithms adjust themselves for

accuracy through gradient descent and backpropagation, making precise predictions for new data.

Both machine learning and deep learning can perform supervised, unsupervised, and reinforcement learning. Supervised learning uses labeled data, unsupervised learning detects patterns without labels, and reinforcement learning optimizes actions based on feedback.

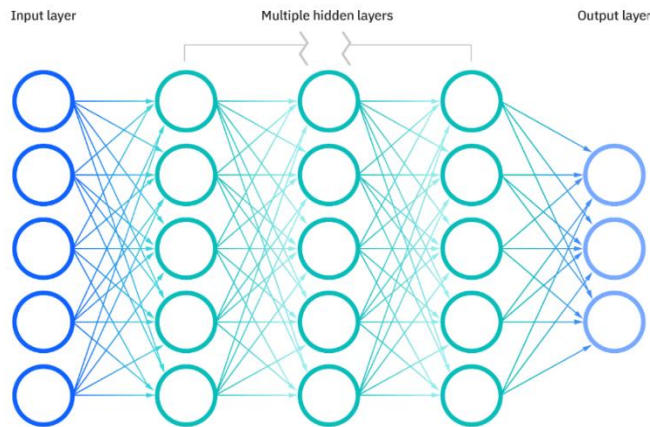
1.2 Neural network:

A neural network, or artificial neural network (ANN), is a core component of deep learning and a subset of machine learning. Inspired by the human brain, it consists of layers of interconnected nodes, each resembling an artificial neuron. These nodes process data, and if their output exceeds a set threshold, they activate and pass data to the next layer. Neural networks use training data to enhance their accuracy and are valuable tools in computer science and artificial intelligence. They excel in tasks like speech and image recognition, significantly speeding up processes compared to manual human efforts.

In a neural network, there are typically three types of layers:

1. Input Layer: This layer receives the initial data and passes it to the subsequent layers.
2. Hidden Layers: These layers perform complex computations on the input data. A neural network can have one or more hidden layers, depending on its architecture.
3. Output Layer: The final layer provides the network's output, which is often a prediction or classification based on the processed data.

These layers work together to process information and make predictions, with the network adjusting its internal parameters during training to improve its accuracy[5].



Neural networks can be classified into different types, which are used for different purposes.

In our project, we utilized Convolutional Neural Networks (CNNs).

1.3 convolutional neural networks:

Convolutional Neural Networks (CNNs) are tailored for tasks like image and audio analysis. They comprise three key layers: convolutional, pooling, and fully-connected layers. These layers collaborate to analyze data hierarchically, beginning with simple features like colors and edges in the convolutional layer and advancing to complex object recognition in subsequent layers. CNNs excel at identifying objects in images or audio by systematically analyzing and categorizing various aspects of the input data.

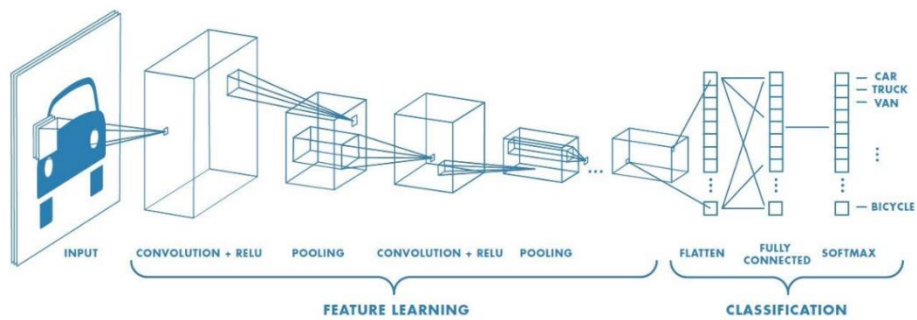
CNNs are a crucial part of deep learning, particularly for tasks involving images, audio, or speech[6]. The layers:

1. Convolutional Layer: This is the core of a CNN and where most of the computation occurs. It operates on the input data, which could be a color image with height, width, and depth (representing RGB channels). A feature detector (kernel or filter) moves across the image, calculating dot products between the filter and the input pixels. This produces a feature map, representing detected features. Hyperparameters like the number of filters, stride, and padding affect the output size.

Additional Convolutional Layers: More convolutional layers can follow the initial one, creating a hierarchical structure. Each layer identifies progressively complex patterns, and the combination of parts forms higher-

level patterns, allowing the network to interpret and extract features effectively.

2. Pooling Layer: Pooling layers reduce input dimensionality and the number of parameters. Filters, without weights, move across the input, applying aggregation functions like max or average pooling within their receptive fields. This simplifies the data, improves efficiency, and reduces overfitting risk.
3. Fully-Connected (FC) Layer: In this layer, each node in the output layer connects directly to a node in the previous layer. It performs classification based on extracted features from earlier layers.



Audio-Video – Active Speaker Localization:

1.4 The Problem:

In today's digital age, where communication is increasingly interconnected, multimedia data has become an integral part of our daily lives. From virtual meetings to content creation, we are surrounded by a vast array of audio and video content. One significant challenge in this landscape is the precise identification of active speakers within these audio and video. This project aims to tackle this challenge by introducing a deep learning approach.

To achieve this goal, we rely on a comprehensive dataset obtained through specialized glasses equipped with cameras and microphone arrays. This dataset captures the nuances of human communication, including speech patterns, visual cues, and the surrounding environment.

In this project we will deal with two problems related to this matter:

- The first and central problem we will face is the problem of locating an active speaker throughout the image in the video. We will solve this problem as a segmentation problem in which we determine for each pixel in the frame for each frame in the video whether there is an active speaker or not.
- The second and secondary problem is determining for each frame whether the wearer of the device is an active speaker. This question can also help the device understand the environment and the social interactions in it. This problem will be solved as a binary classification problem - the wearer of the device is an active speaker or not.

1.5 Our solving approach:

Our approach involves extracting relevant audio features using the Short-Time Fourier Transform (STFT) to capture temporal and frequency characteristics. After that, we employ convolutional neural networks (CNNs) to process the combination of the processed audio segments and the video frames, identifying spatial patterns and visual cues.

At the core of our project is the architecture of our model, consisting of two critical components. Firstly, a ResNet based audio network adapts the ResNet18 architecture to the audio domain, allowing us to extract crucial features from audio signals. Secondly, a U-Net-based video network processes video frames, segmenting them at the pixel level to pinpoint regions where speakers are active.

Our approach is the fusion of audio and video data at multiple stages within the network architecture. This deliberate integration capitalizes on the complementary nature of audio and video cues, further enhancing the precision of speaker localization.

1.6 Signal Representation - Short-Time Fourier Transform (STFT):

The Short-Time Fourier Transform (STFT) is a powerful technique used in signal processing to analyze signals that vary in frequency over time. It is particularly useful when standard Fourier transforms are not adequate because they provide frequency information averaged over the entire signal time interval. STFT operates by applying a window function to the signal, segmenting it into smaller sections, and then calculating the Fourier transform for each segment. This results in a time-localized representation of the signal's frequency content[7]. The STFT pair is expressed as follows:

In continuous domain:

$$X(t, f) = \int x(t) * g(t - \tau) * e^{(-j2\pi f\tau)} d\tau$$

In discrete domain:

$$X(n, \omega) = \sum x(n) * g(n - m) * e^{(-j\omega m)}$$

Here, $x[k]$ represents the signal, and $g[k]$ represents the window function.

STFT allows for a trade-off between time and frequency resolution. Narrow-width windows offer better time resolution but poorer frequency resolution, while wider windows provide better frequency resolution but poorer time resolution.

Spectrograms, which are intensity plots of STFT magnitude over time, are commonly used to visualize STFT results.

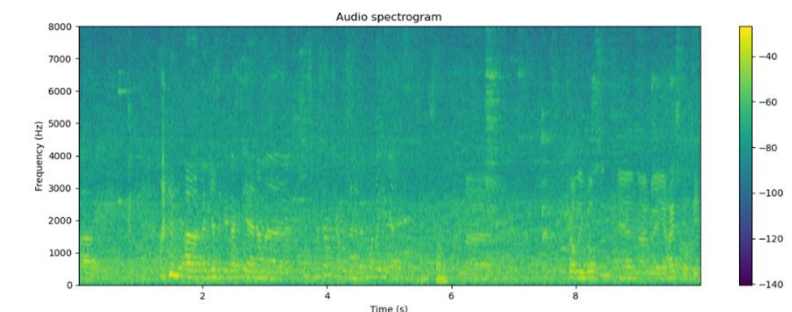
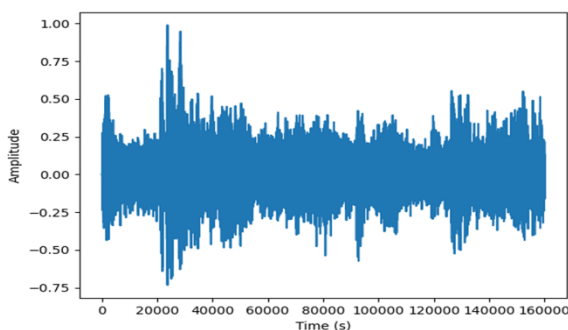
Signal Measurements:

- The bandwidth of the window function $g(t)$ is given by $\Delta\omega^2 = \int |G(\omega)|^2 d\omega / \int |g(t)|^2 dt$, where $G(\omega)$ is the Fourier transform of $g(t)$.
- The spread in time Δt^2 can be expressed as $\Delta t^2 = \int t^2 |g(t)|^2 dt / \int |g(t)|^2 dt$.
- The uncertainty principle (Heisenberg inequality) states that $\Delta t * \Delta\omega \geq \frac{1}{4\pi}$, implying that time and frequency resolution cannot both be arbitrarily small. Gaussian windows meet this bound with equality.

Spectrograms are obtained by taking the square magnitude of STFT results. They provide a two-dimensional representation of time and frequency distribution in a signal. Spectrograms are valuable tools for analyzing signals with non-stationary frequency components.

STFT has found extensive applications in audio signal processing, including music genre classification and speech signal synthesis. However, it is best suited for unimodal, univariate signals with low noise and few component complexities.

Wavelet Transform is considered for improving time-frequency representation and resolution, particularly for signals with transient structures.



1.7 Alternative Solutions:

The solutions to the problem of locating an active speaker throughout the video:

1. **Audio-based solutions:** We can use audio signals to localize the active speaker. One approach is the use of a microphone array. Microphone arrays consist of multiple microphones strategically placed to capture audio from different directions. By analyzing the differences in arrival times and amplitudes of sound at each microphone, it's possible to estimate the direction from which the sound is coming. This can help in localizing the active speaker. Also, deep learning models can be trained to extract meaningful features from audio signals. These features can then be used to identify and localize the active speaker. Deep learning can capture complex patterns in the audio data that may not be evident through traditional audio processing techniques.
2. **Vision-based solutions:** Vision-based solutions rely on visual cues to locate the active speaker within the video frames. One common approach is face detection and tracking. This involves using computer vision algorithms to detect and track the faces of individuals in the video. By identifying and tracking individuals, it becomes possible to determine who is the active speaker at any given time. Also, we can use deep learning for vision-based speaker localization. Deep learning models can analyze video frames to extract relevant features, such as facial expressions, mouth movements, and body pose. These features can then be used to identify and track the active speaker.

2. The Dataset

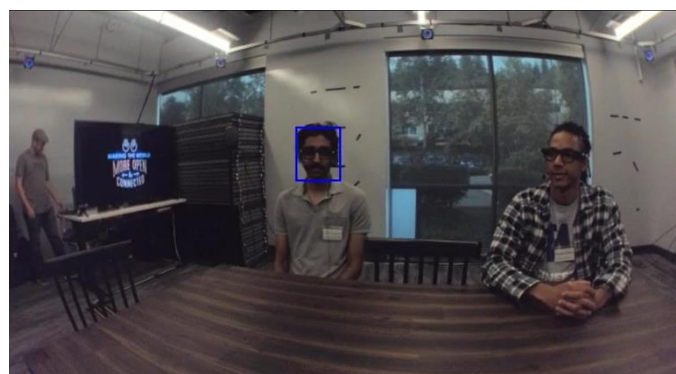
2.1 EasyCom Dataset:

We created our dataset based on Facebook's dataset called "EasyCom"[8]. This dataset is an AR dataset to support algorithms for **Easy Communication** in noisy environments, so it suited our project needs.

This dataset is set around natural conversations in a noisy environment. Participants were equipped with microphones, a camera and tracking markers. They were asked to engage in conversations during several tasks. The participants were exposed to a noisy environment while engaging in conversations. The recordings contain the conversation of all participants and an egocentric video viewpoint of all participants except one. The pose of every participant was also recorded. The dataset was additionally labeled by annotators with voice activity, speech transcriptions, target of speech assignments and head and face bounding boxes with matching participant IDs. The collected data comes from two main sensor modalities, namely audio and vision, and includes spatial and temporal information in both modalities to facilitate research in single channel speech enhancement, beamforming, audio-visual speech enhancement, conversational dynamics and more. The provided pose data, headset microphone data and annotations, including the generated head and face bounding boxes, all complement the main audio and video data for tasks such as model design and analysis.

The parts of the dataset we used are:

- A 6-channel head-mounted microphone array recording.
- Egocentric video (Full HD, Wide FOV).
- Annotated voice activity.
- Generated head and face bounding boxes with annotated IDs.



2.2 Our Dataset:

We took the audio from the 6-channel head-mounted microphone array recording, and processed it as detailed below. We extracted the labeling for the purpose of determining whether the wearer of the device is an active speaker for each video frame from the Annotated Voice Activity.

We took the video from the Egocentric video, and processed it as detailed below. For labeling each pixel in each frame whether it has an active speaker, we created labels using face bounding boxes.

2.3 Audio Preprocessing:

In preparation for our project, we implemented a comprehensive audio data processing workflow. This workflow is essential as it ensures that our audio data is in the optimal format for subsequent deep learning tasks, such as speaker localization.

Let's break down the key steps involved in this audio data processing:

- **Loading the Audio Data:** We initiated the process by loading the audio and video data from our source, which is the Glasses Microphone Array Audio.
- **Down sampling the Audio:** To ensure uniformity and reduce computational complexity, we applied a down sampling technique to the audio signal. This step involved converting the audio signal to a tensor and using a Resample transform to adjust the sample rate from Sampling rate of 48kHz to Sample rate of 16kHz. This standardized the audio data for further processing.
- **Audio Normalization:** We then normalized the audio data. Normalization involves scaling the audio waveform to ensure that its amplitude falls within a specific range. In this case, we normalized the data to a range of -0.99 to 0.99. This step is crucial for ensuring that the audio data is in a consistent and manageable format.
- **Segmentation:** To work effectively with audio data in deep learning, we divided it into smaller, manageable segments. These segments are essentially smaller chunks of audio data that our model can process efficiently. We sliced the audio file into segments of equal duration of 50ms, making it easier to feed into the neural network for training and analysis.

2.4 Video Preprocessing:

To prepare the video data for our project, we implemented a video data processing procedure. This process ensured that our video data was in the optimal format for subsequent deep learning tasks, particularly for speaker localization. Here's a breakdown of the key steps involved in this video data preparation:

- **Loading the Video Data:** We began by loading the video from Easycom MP4 Egocentric video file. This file served as the primary data source for our project, containing the necessary video frames.
- **Down sampling:** The original resolution of the original video is 1080x1920 so we applied down sampling to reduce the frame size to 360x640 pixels. This step involved resizing the frames while preserving their aspect ratio.
- **Frame Saving:** After standardizing the resolution of the video frames, we proceeded to save individual video frames. These frames were saved as smaller pickle files, each containing a single video frame. This segmentation enabled efficient handling and feeding of the video data into our model. The frame rate of the video is 20 frames per second, so each frame represents 50ms and matches one audio segment.

Our dataset has several features:

- An audio, video or both segments can be loaded on demand depending on the nature of the data set task.
- The audio segment can be loaded with "context", that is, several previous segments and a number of subsequent segments. Same for the video. On the one hand, adding context can give more information and a deeper understanding of the model's audio-video context, which may improve performance. On the other hand, increasing the context increases the complexity of the network, the complexity of the required memory and the training time.

- The video can be loaded in "RGB" or "grayscale" format, here too the considerations are similar to the trade-off that exists in choosing the size of the context. "RGB" is richer in information than "grayscale" but working with it is slower and more complex.

2.5 Data Splitting:

At the beginning the dataset is divided into three main subsets: the training set, the validation set, and the test set. The training set is used to train the model, the test set is reserved for final evaluation, and the validation set is used during training to tune hyperparameters and monitor model performance.

2.5.1 Train:

The training process for the multi-Channel speaker localization model involves several key steps. At the beginning of the training process, various lists and dictionaries are initialized to store metrics such as loss and accuracy, and best-performance records are set up to monitor the model's progress. The training loop iterates over a predefined number of epochs, with each epoch representing a complete pass through the training dataset.

Within each epoch, the model undergoes a training step where it learns from the training data. This step involves forward and backward passes through the neural network, updating the model's weights to minimize the training loss, and recording the loss and accuracy for that epoch.

After the training step, the model's performance is evaluated on a validation dataset during the validation step. The goal here is to assess how well the model generalizes to data it has not seen during training.

2.5.2 Validation:

In our project, validation is performed after each training epoch in order to monitor how effectively the model is learning from the training data and to detect any signs of overfitting, where the model becomes too specialized in the training data and performs poorly on new, unseen data. After each training epoch, the model is evaluated using the validation dataset. This evaluation involves passing the validation data through the model and collecting various performance metrics, such as accuracy, loss, F1 score, IoU, recall, and precision.

The validation helps fine-tune hyperparameters by assessing the model's performance with different settings, optimizing results. It enables early stopping, halting training if overfitting and ensuring the model's generalization. Additionally,

the validation helps us to monitor training progress by tracking metrics like loss and accuracy over epochs, revealing trends and issues.

2.5.3 Test:

After training and validation, the model is evaluated on a separate dataset called the test set. This dataset contains examples that the model has not seen during training or validation. It is crucial for assessing the model's ability to generalize to new, unseen data and verifying that it can accurately identify active speakers in both audio and video data.

The testing process begins with a loop that iterates over the test dataset. For each batch of test data, the model is applied to make predictions. During testing, various metrics are computed to assess the model's performance. The model's predictions are compared to the ground truth labels in the test set to calculate these metrics. This evaluation provides insights into how well the model can identify active speakers in both audio and video data.

3. The model

3.1 Architecture:

3.1.1 U-Net Network:

The U-Net is a specialized deep learning architecture primarily used for semantic segmentation tasks in computer vision. It excels at classifying and locating objects within images. Here's a concise overview of its key features and functioning:

Semantic Segmentation Tasks: U-Net is tailored for tasks like semantic segmentation, where the objective is to classify and locate objects in images accurately.

Challenges in Segmentation: One challenge in semantic segmentation is generating an output image with the same dimensions as the input while maintaining spatial information.

Encoder-Decoder Design: U-Net uses an encoder-decoder architecture. The encoder extracts relevant features from the input image, while the decoder reconstructs a segmentation mask with the original image's dimensions.

Fully Convolutional Network (FCN): U-Net is a type of FCN, which is well-suited for segmentation because it maintains spatial alignment throughout the network.

Addressing Feature Loss: U-Net combats the problem of losing detailed features in deep networks by employing skip connections. These connections allow the reintroduction of fine-grained details into the network.

Key Qualities: U-Net is characterized by its encoder-decoder structure, which captures both general and detailed features, making it suitable for tasks like medical image analysis, where precision is vital.

Implementation Details: U-Net implementations involve choices like loss functions (e.g., cross-entropy or dice loss), up-sampling methods (e.g., interpolation or deconvolution), and handling padding in convolution layers.

U-Net is a specialized neural network architecture designed for semantic segmentation tasks. Its unique structure and skip connections enable it to produce high-resolution segmentation masks that match the input image's dimensions, making it especially valuable in applications requiring accurate object localization and classification[9].

3.1.2 ResNet (Residual Network):

ResNet is a significant neural network architecture widely used in computer vision tasks like image classification, object detection, and image segmentation.

ResNet can contain many layers allowing it to capture intricate image features.

ResNet addresses the vanishing gradient issue by using "identity shortcut connections" or skip connections. These connections allow gradients to flow directly through shortcut paths, making it possible to train very deep networks effectively.

ResNet's training involves two stages. In the first stage, the network creates unused layers and skips them using shortcut connections. In the second stage, the network is fine-tuned, and the residual convolutional layers are expanded, enabling the network to recognize complex patterns.

3.1.3 Our Combination:

Our project involves identifying active speakers in video recordings by leveraging both audio and video data. To achieve this, we devised a clever combination of deep learning architectures.

Audio Processing with ResNet:

We started by processing the audio data from the special glasses' microphones using a ResNet architecture.

ResNet, with its capabilities in image classification tasks, is employed to capture meaningful audio features. The ResNet network transforms audio spectrograms (obtained through Short-Time Fourier Transform - STFT) into a structured representation that captures audio characteristics.

Video Processing with U-Net:

We process video frames captured by the glasses' cameras using a U-Net architecture.

U-Net is suited for image segmentation tasks and is used to identify regions of the video frames where someone is actively speaking.

It produces pixel-level segmentation masks for each frame, indicating which pixels correspond to active speakers.

Combining Audio and Video Information:

By combining the information extracted from audio and video, our model can potentially achieve more accurate speaker localization.

The fusion of audio features learned by ResNet and the spatial information captured by U-Net segmentation masks provides a comprehensive understanding of where active speakers are located in the video.

This combination takes advantage of the strengths of both ResNet and U-Net architectures.

ResNet effectively captures audio characteristics and patterns related to speaker activity, while U-Net excels at pixel-level segmentation to precisely locate active speakers within video frames.

We examined two ways to implement the combination between audio and video:

1. The first approach involves the adjustment of the processed audio segment's dimensions to match those of the video segment, achieved through the utilization of a fully connected (FC) layer, audio-to-video concatenation, and the subsequent passage of the resulting audio-video composite through the U-net network.
2. In the second approach, the video data is transmitted through the U-net network until reaching the bottleneck stage. At this juncture, the audio component is concatenated with the video, and the combined data is then transmitted further up the U-net network.

Both combination methods you mentioned have their own advantages and disadvantages:

Method 1:

Advantages:

Effective Fusion: Concatenating the audio and video information before feeding them through the U-Net network can provide a holistic view of both

modalities, allowing the model to learn complex relationships between audio and video features more effectively.

Disadvantages:

Loss of Spatial Information: Early fusion may not take full advantage of the spatial information present in the video frames. The U-Net segmentation masks may not be fully utilized before audio and video are combined.

Increased Model Complexity: The introduction of a big fully connected (FC) layer to adjust audio dimensions can significantly increase the model's parameter count, potentially leading to overfitting and requiring more computational resources.

Method 2:

Advantages:

Preservation of Spatial Information: In this method, the video is processed through the U-Net network until the bottleneck stage before audio is concatenated. This allows the U-Net to capture spatial information from the video frames effectively.

Reduced Model Complexity: Since audio and video are concatenated at a later stage in the U-Net, there may be fewer parameters in the network compared to Method 1, which can help mitigate overfitting and reduce computational demands.

Disadvantages:

Limited Interaction: Delaying fusion until the bottleneck stage means that audio and video features may have limited interaction throughout the network. This could result in a suboptimal fusion of information, especially if there are complex temporal dependencies between the audio and video data that the model needs to capture.

3.2 metrics:

Accuracy: Accuracy is a commonly used metric that measures the proportion of correctly classified instances out of all instances. It provides an overall assessment of how well the model is performing in terms of classification accuracy.

In our project It gives you an idea of the model's ability to correctly classify active speakers.

Recall: measures the proportion of true positive predictions out of all actual positive instances. It assesses the model's ability to correctly identify positive cases.

In our project, we want high recall to ensure that we don't miss any instances of active speakers.

Precision: Precision measures the proportion of true positive predictions out of all positive predictions made by the model. In our project, it assesses whether the model correctly identifies speakers without mistakenly including non-speaker regions. High precision ensures that the model's predictions are reliable.

F1 Score: The F1 score balances precision and recall. In our project, it is essential because we want to avoid both false positives and false negatives when identifying active speakers. Achieving a high F1 score ensures that the model maintains a good balance between correctly identifying active speakers (precision) and not missing any (recall).

IoU (Intersection over Union): IoU is a metric commonly used in image segmentation tasks. It measures the overlap between the predicted and ground truth regions of interest. A higher IoU indicates better segmentation accuracy.

In our project, IoU measures the extent to which the model's segmented regions overlap with the actual speaker regions in the video frames. A higher IoU indicates that the model is accurately segmenting the regions where someone is talking.

Tversky Score: The Tversky score is a metric that balances the trade-off between precision and recall. We use the Tversky score to give more weight to either precision or recall, allowing you to fine-tune the model's behavior based on specific needs[11].

Loss function:

A loss function is a vital component in machine learning and deep learning models. Its core purpose is to measure how well a model performs a given task by quantifying the difference between its predictions and the true target values for a set of input data points. The loss function yields a numerical value that reflects the dissimilarity between predicted and actual outcomes, guiding the model's performance assessment.

The primary goal is to minimize this loss function, indicating that the model's predictions are becoming more accurate and closer to the actual values.

Loss functions play a role in balancing the trade-offs between overfitting (overly fitting the training data but performing poorly on new data) and underfitting (oversimplifying the model and missing data patterns). The selection of an appropriate loss function hinges on the specific task and data type under consideration [10].

Binary Cross Entropy Loss (BCELoss):

BCELoss is a crucial loss function in PyTorch used primarily for binary classification tasks. Its purpose is to measure the dissimilarity between two probability distributions: one representing the predicted probabilities of the positive class (1) and the other for the negative class (0).

The mathematical definition of BCELoss is as follows:

$$BCELoss(x, y) = -\frac{1}{n} \sum_{i=1}^n (y_i \log(x_i) + (1 - y_i) \log(1 - x_i))$$

BCELoss quantifies how well the predicted probabilities align with the actual labels. It calculates a penalty for discrepancies between the predicted probabilities and the ground truth. The more accurate the model's predictions, the lower the BCELoss.

In our binary classification project, where we determine whether an audio segment or pixel in the video segment contains an active speaker (class 1) or not (class 0). Throughout training, our model predicts whether a segment contains an active speaker (1) or not (0). BCELoss gauges how closely these predictions align with the actual dataset labels. It calculates a loss value, quantifying the disparity between predicted probabilities and ground truth labels. By minimizing this loss during

training, we enhance our model's capacity to accurately identify active speakers, bringing predictions closer to actual values.

Additionally, BCELoss aids in achieving a delicate balance between overfitting and underfitting. Given that our project fuses information from both audio and video sources for active speaker identification, we can apply BCELoss separately to each modality. This approach enables independent performance assessment of audio-based and video-based models, ensuring both elements contribute effectively to the final decision.

Dice loss:

$$DL(y, \hat{p}) = 1 - \frac{2y\hat{p} + 1}{y + \hat{p} + 1}$$

The Dice Coefficient is a renowned evaluation metric for image segmentation, and it can also serve as a loss function. While it may not be as commonly used as binary cross-entropy (BCE), the Dice Coefficient proves to be highly effective, particularly in addressing class imbalance issues, which is crucial for our mission of determining the presence of an active speaker in each pixel of video frames.

Unlike BCE, which considers both the segmentation class and the background class, the Dice Coefficient exclusively focuses on the segmentation class. It categorizes pixels as True Positive (TP), False Negative (FN), or False Positive (FP).

The Dice Coefficient quantifies the overlap between the predicted mask and the ground truth. Since it does not account for the background class, it does not dominate over the smaller segmentation class. The Dice Coefficient generates a score within the range of [0, 1], where 1 signifies a perfect overlap. Therefore, (1 - DSC) can be effectively employed as a loss function.

By aiming to maximize the Dice Coefficient, the network naturally addresses class imbalance, making it a suitable choice for our imbalanced data. It excels in scenarios where most frames do not contain a speaker, and even when there is a speaker, most pixels are not in the speaker's location.

Pros:

1. **Effortless Handling of Class Imbalance:** Dice Coefficient provides a straightforward solution for dealing with class imbalance without requiring manual parameter tuning.
2. **Effective for Segmentation:** It excels in scenarios where accurate segmentation is essential, such as identifying active speakers within video frames.

Cons:

1. **Potential for Gradient Explosion:** During initial training phases, the Dice Coefficient can lead to unstable training due to the risk of exploding gradients. However, this can be managed with batch normalization and ReLU activation functions.
2. **Slower Training:** In comparison to BCE, training with the Dice Coefficient loss may be somewhat slower. However, the trade-off in addressing class imbalance makes it a favorable choice for certain applications.

Tversky loss:

Tversky Loss and its Relationship to Precision and Recall

The Tversky Loss is a modified loss function, derived from the Tversky Index, which introduces additional control parameters, α and β , to finely balance false positives (FP) and false negatives (FN) during training. This loss function is particularly valuable when dealing with imbalanced datasets, such as our task of detecting active speakers in video frames.

$$TI = \frac{TP}{TP + \alpha FN + \beta FP}$$

- α and β are parameters that control the balance between FP and FN. Importantly, $\alpha + \beta = 1$.

When $\alpha=\beta=0.5$, the Tversky Loss is equivalent to the Dice Loss. However, by adjusting α and β , you can tailor the loss function to your specific needs. This flexibility is particularly useful for addressing imbalanced datasets.

How Tversky Loss Relates to Precision and Recall:

- **Precision:** Precision measures the ratio of true positives (TP) to the total number of positive predictions made by the model ($TP/(TP+FP)$). In the context of the Tversky Loss, adjusting α and β allows you to emphasize either FP or FN. Setting $\alpha>\beta$ increases the importance of reducing FP, thereby potentially improving precision by minimizing false positive predictions.
- **Recall:** Recall measures the ratio of TP to the total number of actual positives in the dataset ($TP/(TP+FN)$). By fine-tuning α and β , you can control the trade-off between FP and FN. If your objective is to enhance recall, you can set $\beta>\alpha$ to prioritize minimizing false negatives, ensuring that more actual positives are correctly identified.

Pros of Tversky Loss compared to Dice Loss and BCE Loss:

1. **Customizable Trade-off:** The Tversky Loss provides a customizable trade-off between precision and recall by adjusting α and β , making it adaptable to the specific requirements of your dataset and task.
2. **Imbalanced Data Handling:** Tversky Loss excels in scenarios with imbalanced data, where precision and recall become crucial metrics for evaluation.
3. **Fine-grained Control:** Unlike Dice Loss, which provides a balanced approach, and BCE Loss, which doesn't offer control over FP and FN trade-offs, Tversky Loss offers fine-grained control over these trade-offs, enhancing the model's ability to optimize precision and recall.
4. **Real-world Application:** Tversky Loss has been successfully applied in various real-world scenarios, including medical image segmentation, where precision and recall are often of paramount importance.

In summary, the Tversky Loss is a powerful tool in the deep learning toolkit, offering a way to finely control the trade-off between precision and recall, particularly in the

context of imbalanced datasets. Its adaptability and control over error types make it a suitable choice for our mission of detecting active speakers in video frames, where optimizing precision and recall is critical, given the imbalanced data distribution.

Focal Tversky loss:

The Focal Tversky Loss (FTL) represents a powerful refinement of the Tversky Loss, introducing non-linearity to the loss function, which enables precise control over its behavior at various Tversky Index (TI) values.

- **γ Parameter:** At the core of FTL lies the γ parameter, which governs the non-linearity of the loss. The γ value plays a pivotal role in shaping the behavior of the loss. As γ tends towards positive infinity, the loss gradient increases exponentially as TI approaches 1. Conversely, when γ tends towards 0, the gradient becomes minimal as TI approaches 1.
- **Behavior with $\gamma < 1$:** When $\gamma < 1$, the loss gradient becomes more pronounced for examples where $TI > 0.5$. This compels the model to prioritize learning from such instances. This behavior proves valuable towards the end of training when TI is close to convergence. However, during the initial stages, it can lead to the model assigning higher weights to easier examples, potentially hindering learning.
- **Addressing Class Imbalance with $\gamma > 1$:** FTL truly shines in cases of class imbalance, where $\gamma > 1$. In this scenario, the loss gradient becomes significantly higher for examples where $TI < 0.5$. This strategic adjustment compels the model to focus on challenging examples, especially those involving small-scale segmentations that often yield low TI scores.

In real-world experiments, specifically with the BUS dataset, FTL demonstrated significant improvements across all evaluation categories. Although the ISIC dataset exhibited the highest overall Dice Coefficient with FTL, it did not consistently yield the highest precision or recall.

Pros of Focal Tversky Loss compared to Tversky Loss, Dice Loss, and BCE Loss:

1. **Adaptive Non-linearity:** FTL's adaptive non-linearity, controlled by the γ parameter, enables tailored behavior, making it highly versatile in different training scenarios.
2. **Effective Class Imbalance Handling:** With $\gamma > 1$, FTL excels at addressing class imbalance by focusing on challenging examples, crucial for tasks like detecting active speakers in video frames with imbalanced data distributions.
3. **Optimal Precision-Recall Trade-off:** FTL strikes a balance between precision and recall, ensuring that class imbalance is effectively tackled. The result is often an improved overall Dice Coefficient.

Conclusion

In summary, the Focal Tversky Loss emerges as a robust and adaptable solution for handling class imbalance in deep learning tasks. Its fine-tuned non-linearity through the γ parameter enables effective control over the loss function, making it suitable for various stages of training and datasets with varying imbalance levels. For our mission of detecting active speakers in video frames with imbalanced data, the Focal Tversky Loss provides an easy-to-implement drop-in solution to enhance model performance and tackle class imbalance effectively.

3.3 Additional ways to handle the unbalanced data:

Grayscale

Another way we used to try and handle the imbalance of our data is grayscale. Grayscale is a type of image representation that encodes visual information in shades of gray. It typically uses a single channel, and omits color information, simplifying images while retaining essential contrast and brightness details.

Grayscale images can be beneficial in addressing imbalanced data in our project, which involves the localization of active audio and video speakers using deep learning. Grayscale images have only one channel (intensity), while color images have three (red, green, blue). Using grayscale simplifies our data, potentially reducing the complexity of our model. We tried to use grayscale in order to help the model focus on relevant features. Additionally, Grayscale images typically require fewer computational resources during both training and inference compared to color images. This can lead to faster training times and more efficient model predictions, making it easier to work with imbalanced data.

Down sampling

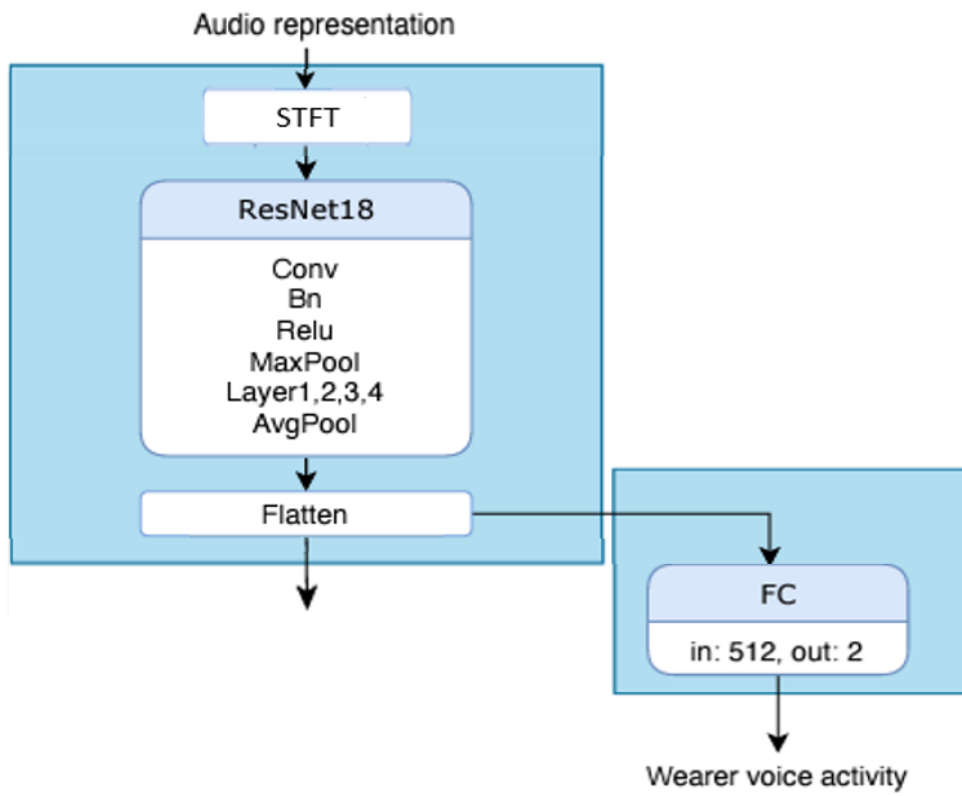
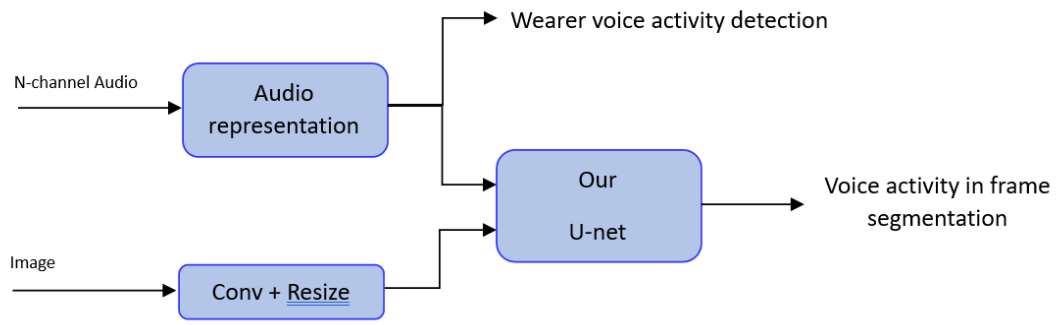
Down sampling involves reducing the size of the overrepresented class (majority class) by randomly selecting a subset of its samples, matching it with the size of the underrepresented class (minority class). This helps prevent the model from being biased towards the majority class and can lead to improved model performance on imbalanced datasets.

Down sampling can help address imbalanced data in our project by reducing the amount of data in the majority class (non-active speakers) to balance it with the minority class (active speakers). This can prevent our model from being biased towards the majority class and improve its ability to learn features related to active speakers.

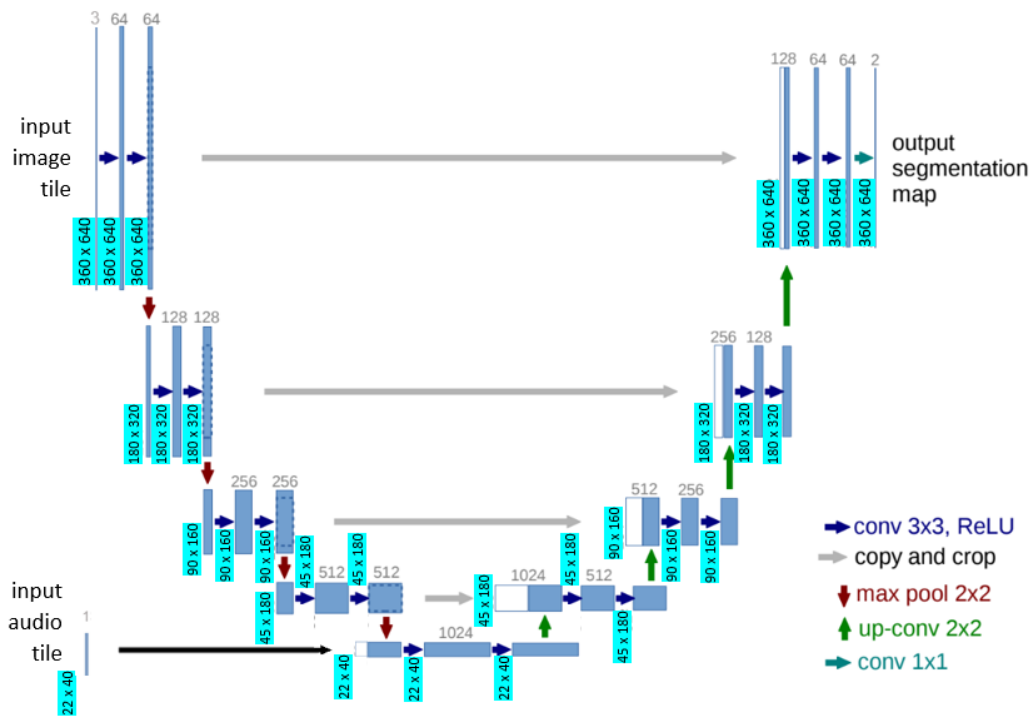
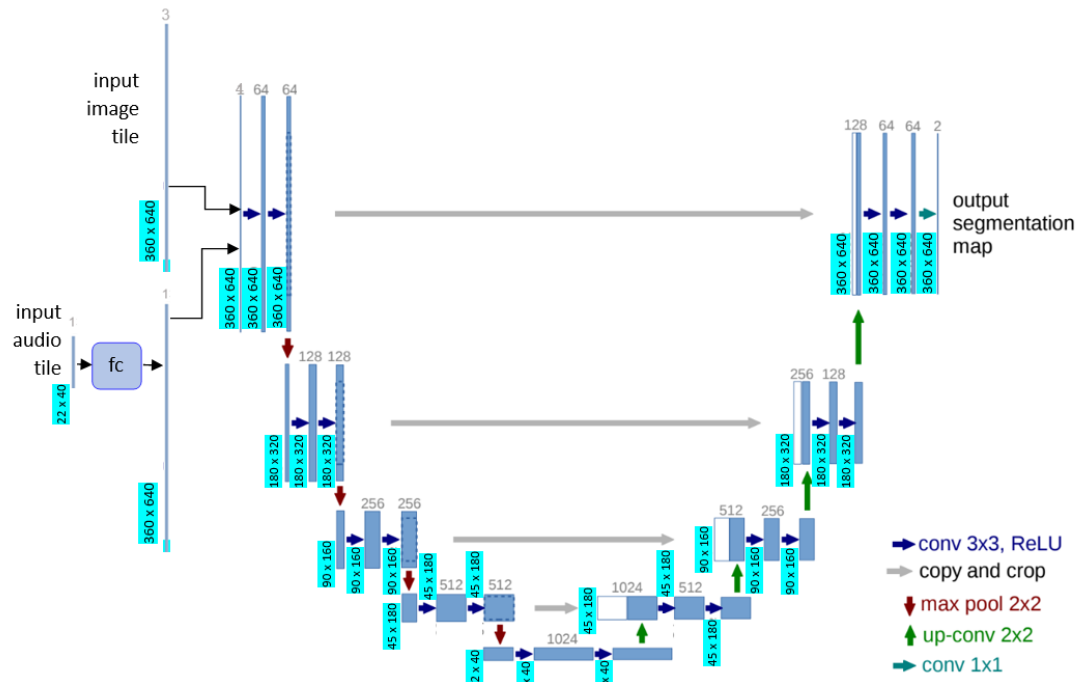
Training exclusively on frames featuring speakers

In an effort to address the issue of imbalance, we made an attempt to train the model exclusively on frames featuring speakers. In this training, although there were speakers in all the frames, the number of pixels containing a speaker is still much smaller than the number of pixels that do not contain a speaker.

Model Block Diagram:



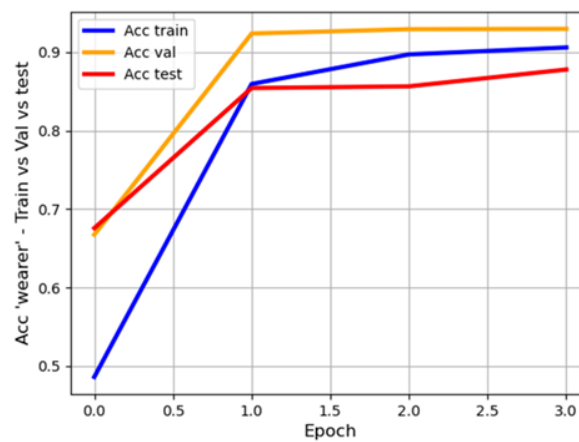
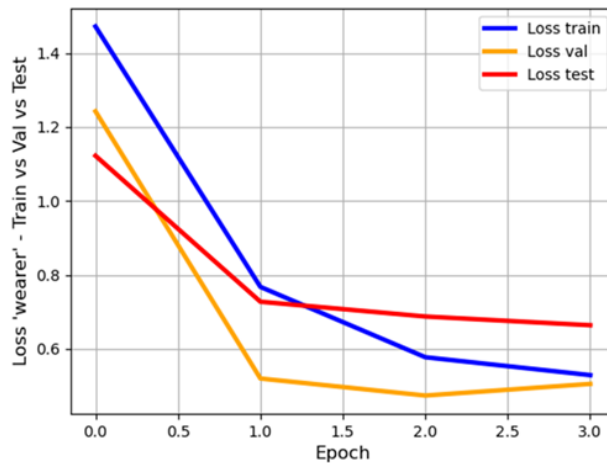
Our U-Net - First and second approach



4. The Results:

4.1 Wearer voice activity detection task

For this task after 4 epochs the validation loss stopped decreasing together with the train and thus, we finished the learning process of the model for the task.

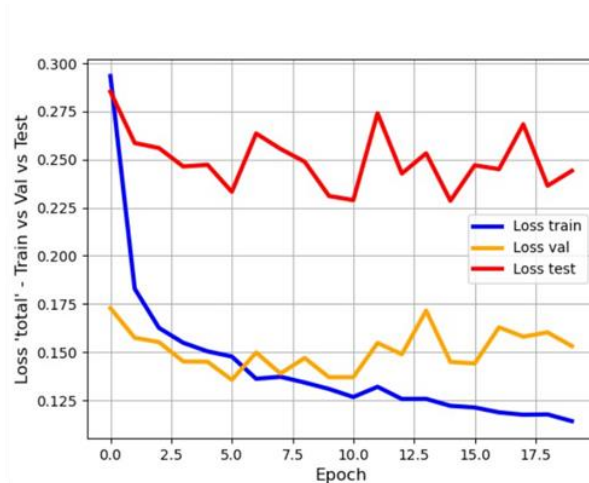


We got good accuracy and a good balance between recall and precision. Although the recall we received at the beginning of the learning was higher, the precision improved and overall, the model improved, as reflected in the F1 (which weighs recall and precision) which increased during the learning.

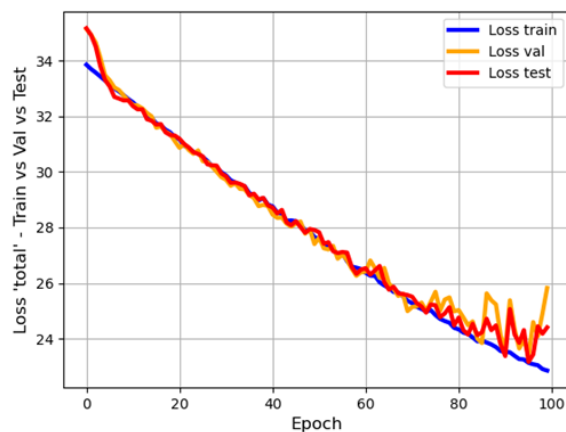
	Accuracy	Recall	Precision	F1
Epoch 1	67.56%	84.78%	55.48%	65.52%
Epoch 4	87.78%	75.08%	91.67%	80.82%

4.2 Voice activity in frame segmentation task

For the speaker segmentation task we received that already in the first epoch the loss of the validation does not decrease together with the loss of the train.

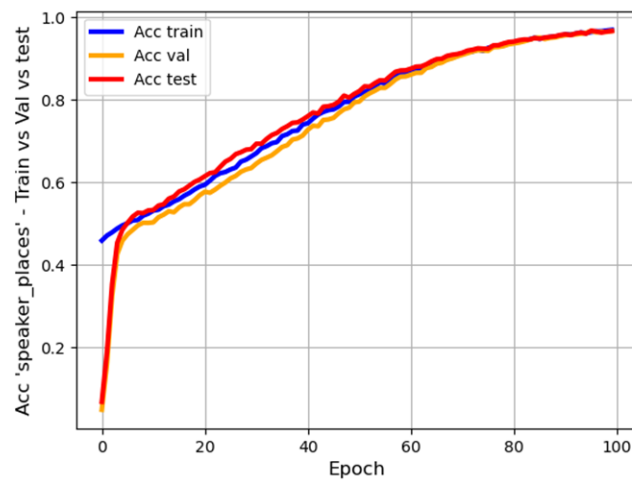


We realized that the learning of the model is done quickly and in the early stages of the training. To properly see the learning process we defined mini-epochs and trained the model on them.



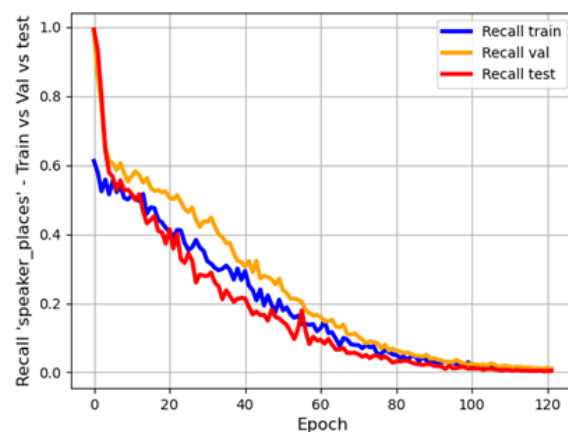
Now it can be seen that learning is indeed taking place in the initial stages of training. After about 80 mini-epochs, the validation loss stopped decreasing along with the train, thus ending the learning process of the model for this task.

We received that the model during learning the accuracy improves and reaches an accuracy level of almost 100%.

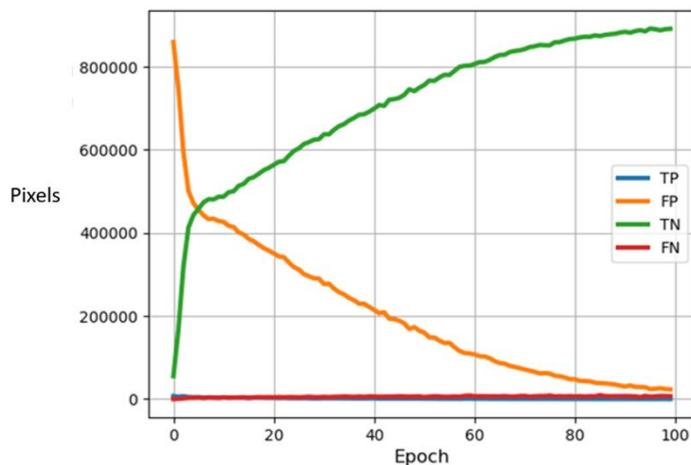


However, this does not indicate that the model is good enough to identify active speakers in the frame. Since the number of pixels containing an active speaker is a negligible minority of all pixels (a ratio of 1:100 in favor of the number of pixels that do not contain an active speaker in our dataset), an arbitrary guess that the pixel does not contain an active speaker will be correct in the absolute majority and will yield high accuracy. This is the data imbalancing problem that we expanded on above.

As you can see, after the training our model does tend to guess that Pixel is not an active speaker and thus his accuracy increases, but at the same time his recall decreases. That is, the model is correct most of the time but is not sensitive to active speakers.

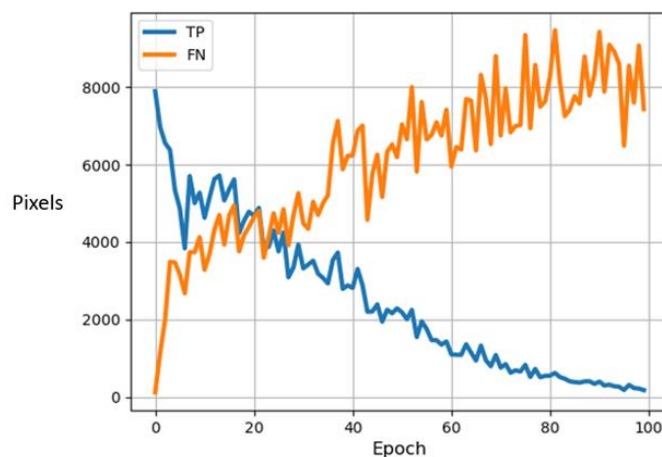


In the graph you can see the values of the confusion matrix:



The FP decreases while the TN increases. There are also changes in TP and FN but since their amount is negligible, we will accept that overall, the accuracy whose formula $(TP+TN)/ALL$ increases.

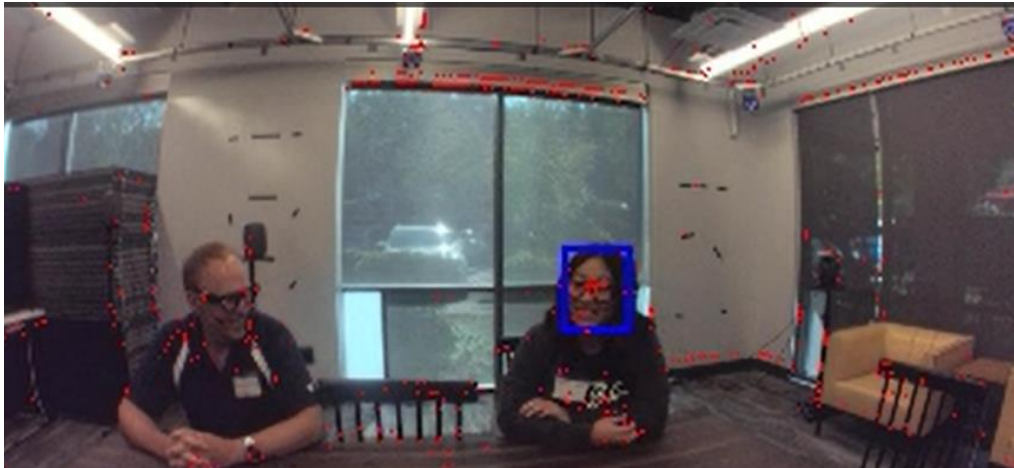
The recall, on the other hand, is determined by the TP and FN only and its formula is $TP/(TP+FN)$, we will look at them:



Together with the increase of the TN, the FN also increases and together with the increase of the TN, the TP also decreases. Therefore the recall decreases despite the increase in accuracy.

If we run the model in the early stages of learning, we will get a correct detection of speaker location, but also a false detection of speakers - low accuracy and high recall as we explained.

This is the result obtained after two mini-epochs. In a blue square - pixels where there is an active speaker. In red dots - the prediction of the model for the location of the active speakers.



And this is the result obtained after a large number of mini-epochs. There are almost no detections of an active speaker in the frame.



The results we presented are for BCE loss. To solve the problem of imbalancing data, we tried to use different loss functions (which we expanded on in the chapter on Loss functions) that may be more suitable for this situation, but there was no noticeable change in the results.

5. Conclusions and Summary:

Achievements:

Our project endeavors to address the intricate challenge of localizing an active speaker in both audio and video data through innovative deep learning methods. The integration of audio and video data demonstrates practical applications across diverse domains.

While our project has shown promising advancements, particularly in achieving commendable accuracy and a balanced trade-off between recall and precision for voice activity detection in device wearers, the overall success of the model in the segmentation task of active speakers in video frames remains a work in progress. Despite continual improvement in accuracy during training, the segmentation performance reveals lingering challenges. The model's struggle with class imbalance in video frames has resulted in high accuracy but low recall, leading to a notable number of false negatives. This issue underscores the significance of addressing data imbalances to enhance model performance in such scenarios.

Advantages of our Solution:

1. Our project addresses the challenging problem of localizing an active speaker in both audio and video data, using deep learning methods. The combination of audio and video data for this task is innovative and can have practical applications in various domains.
2. We achieved good accuracy and a balanced trade-off between recall and precision in the voice activity detection task for the wearer. This indicates that our model can accurately identify whether the wearer is an active speaker.

Disadvantages of our Solution:

1. Our model struggles with the class imbalance problem when identifying active speakers in video frames. The vast majority of pixels do not contain an active speaker, leading to high accuracy but low recall. This is a significant challenge and can result in many false negatives.

2. While accuracy improved during training, the overall segmentation performance is not satisfactory. The model lacks sensitivity to active speakers, which is a crucial issue for this task. The imbalance in data distribution significantly affects our results.
3. Combining audio and video data is a complex task, and it's possible that our model might need more data and a more sophisticated architecture to address the challenges.

Project Insights:

Throughout the course of our project, we gained valuable insights into various aspects of audio and video preprocessing for deep learning model training. Our journey involved adapting an existing dataset to effectively address the challenges associated with our unique problem.

A key highlight of our learning process was the exploration of diverse classification and segmentation models designed for audio and video applications, respectively. This exploration extended to a comprehensive understanding of the architectural layers within these models and their functional significance.

An essential skill we developed was the ability to customize existing models to align with our specific requirements. This involved adapting our data to fit seamlessly into established models and making necessary modifications to cater to our project's needs. An intriguing aspect of our work revolved around integrating an audio model with a video model. This fusion aimed at creating a solution for speaker localization within a frame.

Our journey in training deep learning models encompassed fundamental steps, such as segregating data into training and validation sets, and rigorously evaluating model performance with a test set. The critical decision-making process involved selecting an appropriate loss function tailored to our unique problem domain, prompting a thorough examination of multiple options.

To gauge the efficacy of our models, we delved into various metrics for measuring performance and extracting meaningful insights. Additionally, our project brought us face-to-face with the challenge of data imbalance, prompting us to explore and

implement diverse strategies to address this issue.

In essence, our project not only enriched our understanding of audio and video processing within the context of deep learning but also equipped us with a diverse set of skills, from model adaptation to performance evaluation, essential for navigating the intricacies of modern machine learning projects.

6. possible future projects:

There are several possible future projects that can build upon our project:

1. Leveraging transfer learning from models that have been pre trained on large audio and video datasets. Fine-tuning a pretrained model can enhance our system's understanding of complex audio and video features.
2. Utilizing head pose and eye gaze to assist the system in identifying and tracking active speakers, even when they are outside the camera's field of view. For instance, if a user is looking at a person who is speaking, the system could use this information to infer that the individual is an active speaker.
3. Developing a real-time implementation of the system that can be deployed on mobile devices. This would expand the system's utility across various applications, including video conferencing and social media.
4. Concentrate on enhancing the system's accuracy and robustness, particularly in challenging conditions such as low-light and noisy environments.
5. Expanding our project to focus on localizing multiple speakers in both audio and video.

7. Bibliography:

1. "Egocentric Deep Multi-Channel Audio-Visual Active Speaker Localization", Hao Jiang, Calvin Murdock, Vamsi Krishna Ithapu
https://openaccess.thecvf.com/content/CVPR2022/papers/Jiang_Egocentric_Deep_Multi-Channel_Audio-Visual_Active_Speaker_Localization_CVPR_2022_paper.pdf
2. "Egocentric Audio-Visual Object Localization", Chao Huang, Yapeng Tian, Anurag Kumar, Chenliang Xu, University of Rochester, Meta Reality Labs Research
https://openaccess.thecvf.com/content/CVPR2023/papers/Huang_Egocentric_Audio-Visual_Object_Localization_CVPR_2023_paper.pdf
3. "Egocentric Auditory Attention Localization in Conversations", Fiona Ryan, Hao Jiang, Abhinav Shukla, James M. Rehg, Vamsi Krishna Ithapu Georgia Institute of Technology , Meta Reality Labs Research
https://openaccess.thecvf.com/content/CVPR2023/papers/Ryan_Egocentric_Auditory_Attention_Localization_in_Conversations_CVPR_2023_paper.pdf
4. What is deep learning? , IBM <https://www.ibm.com/topics/deep-learning>
5. What are neural networks? , IBM <https://www.ibm.com/topics/neural-networks>
6. What are convolutional neural networks? , IBM
<https://www.ibm.com/topics/convolutional-neural-networks>
7. Short-Time Fourier Transform (STFT), ScintDirect
<https://www.sciencedirect.com/topics/engineering/short-time-fourier-transform>
8. Facebook EasyCom Dataset, github
<https://github.com/facebookresearch/EasyComDataset>
9. "Understanding U-Net", Minh Tran, Medium
<https://towardsdatascience.com/understanding-u-net-61276b10f360>
10. loss functions:
 - saturn cloud, <https://saturncloud.io/blog/how-to-use-bceloss-in-pytorch/>
 - "A survey of loss functions for semantic segmentation", Shruti Jadon
<https://arxiv.org/pdf/2006.14822.pdf>

11. "Dealing with class imbalanced image datasets using the Focal Tversky Loss",

Robin Vinod

<https://towardsdatascience.com/dealing-with-class-imbalanced-image-datasets-1cbd17de76b5>

Appendix:

The code and the results:

<https://github.com/RoyDovrat/Audio-Video-Active-Speaker-Localization>