





# An Approach for Audio-Visual Content Understanding of Video using Multimodal Deep Learning Methodology

 Emre Beray Boztepe<sup>1</sup>,  Bedirhan Karakaya<sup>2</sup>,  Bahadır Karasulu<sup>3</sup>,  İsmet Ünlü<sup>4</sup>

<sup>1</sup>Corresponding Author; Çanakkale Onsekiz Mart University, Department of Computer Engineering; berayboztepe@outlook.com

<sup>2</sup>Çanakkale Onsekiz Mart University, Department of Computer Engineering; bedirhankrky@gmail.com

<sup>3</sup>Çanakkale Onsekiz Mart University, Department of Computer Engineering; bahadirkarasulu@comu.edu.tr

<sup>4</sup>Çanakkale Onsekiz Mart University, Department of Computer Engineering; ismetyny@gmail.com

Received 02 July 2022; Revised 5 July 2022; Accepted 6 July 2022; Published online August 2022

## Abstract

This study contains an approach for recognizing the sound environment class from a video to understand the spoken content with its sentimental context via some sort of analysis that is achieved by the processing of audio-visual content using multimodal deep learning methodology. This approach begins with cutting the parts of a given video which the most action happened by using deep learning and this cutted parts get concanarated as a new video clip. With the help of a deep learning network model which was trained before for sound recognition, a sound prediction process takes place. The model was trained by using different sound clips of ten different categories to predict sound classes. These categories have been selected by where the action could have happened the most. Then, to strengthen the result of sound recognition if there is a speech in the new video, this speech has been taken. By using Natural Language Processing (NLP) and Named Entity Recognition (NER) this speech has been categorized according to if the word of a speech has connotation of any of the ten categories. Sentiment analysis and Apriori Algorithm from Association Rule Mining (ARM) processes are preceded by identifying the frequent categories in the concanarated video and helps us to define the relationship between the categories owned. According to the highest performance evaluation values from our experiments, the accuracy for sound environment recognition for a given video's processed scene is 70%, average Bilingual Evaluation Understudy (BLEU) score for speech to text with VOSK speech recognition toolkit's English language model is 90% on average and for Turkish language model is 81% on average. Discussion and conclusion based on scientific findings are included in our study.

**Keywords:** Multimodal Deep Learning, Association Rule Mining, Named Entity Recognition, Natural Language Processing

## 1. Introduction

Nowadays, decomposition of various environmental sounds for environment recognition has gained popularity. Various background sounds in videos could be classified with high success with deep learning and machine learning techniques. In this way, semantically enriched video scenes can be depicted. Also, different techniques have been used to strengthen the result from sound recognition to recognize the environment well and increase the accuracy which was gained from trained models for the environment. Sound recognition, which emerged with the development of technology, is a technology based on the analysis of the audio signal [1]. Environmental sound recognition is a technology which is used for detecting sounds that define everything about the environment such as animal sounds, human sounds, car sounds and so on. Technology for developing new techniques for recognizing environmental sounds by using deep learning has been improved so much nowadays. This technology is important to be used for different sectors such as cinema, different departments such as police and fire departments, and etc. It could help to catch the criminals by detecting the environment if there is a video recording with sound. Different fields such as multimedia systems, it is used for detection of sound scene events [2] and such as sentiment analysis techniques, it is used for audio emotional state classification using audio features [3] and so on.

There are different techniques to build a model for speech recognition by using deep learning. Getting a spectrogram from an audio signal and building a Convolutional Neural Network (CNN) model using

spectrograms as inputs is one of the most used techniques [4]. Adding Recurrent Neural Networks (RNN) after CNN to create a Convolutional Recurrent Neural Network (CRNN) model is another technique that could have been used for this problem [5].

In multimodal deep learning methodology, data is obtained from different sources. Furthermore, these data can be used to learn important features over multiple modalities. In this way, a representation would be constructed between different modalities (e.g., audio, visual and possibly textual data) in a shared manner as well. Instead of using single mode data for understanding of audio-visual content, the use of multiple modalities is expected to result in highest performance. In machine learning and deep learning, representation learning is a process that automatically proceeds for a learning system to discover the representations needed for feature detection, selection or classification from given data. With the help of the representation learning, the semantic gap between low level features and semantically rich high level features would be filled to achieve a successful understanding of content [6].

Audio data augmentation techniques can be built with some items such as an attribute type consisting of coefficients in the Mel scale Mel-frequency cepstral coefficient (MFCC) [2] type feature extraction method, Chroma [3] type feature extraction method, adding Gaussian noise to the audio track, shifting the audio track, and so on can be used to get an augmented data [2]. Using different Natural Language Processing (NLP) techniques to strengthen the result of sound recognition prediction can be beneficial to increase the accuracy because the raw sound can not be enough for detecting the environment. Displaying audio signals that are in the same frequencies is easy but building a model to detect sound from them is not easy enough than building image recognition models. These NLP techniques help us to do some processes. For example, the tokenization method which is used to get all the words individually from a sentence [7, 8].

The Part of Speech Tagging (POS tagger) method is another technique for the process of marking up a word in a text (i.e., corpus) as corresponding to a particular part of speech, based on both its definition and its context. A simplified form of this is commonly taught to everyone starting from childhood, in the identification of words as nouns, verbs, adjectives, adverbs, etc. The POS tagger technique is a method to help to define what is the job of the word in the sentence [9]. In addition, the stemming method which is another technique which is used to get stems of a word and some other methods could be given as an example [8]. Implementation of these techniques is different according to language that has been used.

The Named Entity Recognition (NER) is important to find the relation between the word and the category in which the word belongs to the related category [10]. For example, if someone heard some words from a sentence like "*penalty kick*", everyone would think that it is about football. NER helps to categorize these words to its related category. There are a lot of ways to do this process. In this scope, a novel dataset could be created by searching all the words and phrases and collecting them somewhere (in a spreadsheet file, database, and etc.), some libraries from some programming languages could be used or pre trained files could be found. But in order to implement this process, the first thing needed to do is to implement NLP techniques to the words in a sentence. Otherwise, all conjugations of the word have to be added to the created NER file to be checked. But that also means more time needed to search related words and create this NER file and more time to search if the word that is found is in the NER file or not.

The Association Rule Mining (ARM) is a kind of process used to determine the relation between the given categories [11]. It is mostly used in the marketing sector to determine the relation between the items that are shopped. International e-commerce and entertainment companies are using this method a lot and everyone could see that technique like "*this another item is bought with the item you bought, would you prefer to buy this item too?*" or "*this movie is watched by people who watched the movie you watch*". The implementation explained at this section of this work is to determine to make a comment like "*if this category is in this video, the rate of another category to be in this video increases this much*". This way could be used to collect data of categories and suggest a video to users about a video that is like "*the user's video is about this category and the user may also want to view this category too*". In our study, sound environment recognition for a given video's processed scene is

achieved with high accuracy. In addition, the speech to text process of our approach based on the VOSK offline speech recognition [12] toolkit's English language model and Turkish language model have highest performance results that prove our approach's superiority. Of the three English VOSK models, the second model was chosen by us to achieve optimized performance. This model is more accurate than the first, but faster than the most complex. There is just one VOSK model for the Turkish language. This model is used for getting the Turkish speech.

Our main contribution to the literature is to be able to realize and prove the content understanding mechanism for determination of the sentiment from spoken content based on the multimodal information obtained from multilingual audio-visual content extracted using a deep learning approach without any other manual intervention.

In addition, the second contribution of our study is applying sound scene classification and recognition to reach a refined sentiment analysis from spoken content that related sentences are taken from the processed video clip using speech recognition, natural language processing (NLP) and ARM techniques. In the literature, there are several analysis methods for sentimental context that are pre-trained and used with some published studies [13, 14]. Everyone could choose any one of this kind of pre-trained model by respect to its performance, by its language type, by its emotional representation styles (negative/positive speech, angry/happy speech etc.) and by its model size. These kinds of models are used to determine the given sentence's sentiment by its determination of any positive sentence, negative sentence or neutral sentence. For example, when the video is related to the football category and its spoken content covers news about a team called "*Galatasaray*" from the Turkish Super League a loss they have taken from yesterday's match then sentiment analysis model helps to determine the speech from the news. Furthermore, the model can say that "*this sentence is a negative sentence related to the football category*". It actually is not about finding the category or strengthening the found category from sound recognition, but it is a technique to find out if the model has negativity or positivity. With this way, it can be said about the example that is given, this actioned part from a video has a negativity. At a glance, it is a challenging issue for real world problems.

The remaining sections are organized as follows. Literature review is given as the second section. In the third section, materials and methods are presented with the details of video summarization, sound event detection and speech recognition, and also, multimodal deep learning methodology, system integration and Graphical User Interface (GUI) are given in the section as well. Experimental results are introduced in the fourth section which covers datasets used in the experiments, performance evaluation methodology, our experimental results and proof of concept of our approach. In the last section, the discussion and conclusion of scientific findings of our study are included to emphasize the contributions of our work.

## 2. Related Work

At a glance, there is a huge potential to extract useful information from raw data in contemporary multimedia systems. Multimedia is a term that covers text, audio, image and video as given as a processable object. In the literature, deep learning is used to process and extract these kinds of information for building a consequent knowledge about the solution of a real world problem using the above mentioned multimedia components. Nowadays, sound event recognition, speech recognition, automatic speaker recognition and speech to text translation are especially popular issues in the deep learning and machine learning research areas. With its well constructed paradigms. Deep learning provides an automatized feature extraction and selection scope belonging to its layered structure which proves a refined learning called representative learning [15, 16].

In this scope, video is an integrated multimedia component that involves audio and consecutive images as video frames. These frames mostly include some important information about actions or events. Since these actions or events may be grouped into one or more frame groups (i.e., video shot), a deep learning system can be easily used in these frames to detect and classify them from certain mechanisms. Audio features based sound event recognition is relatively less complex than visual feature based action recognition. For sound event detection and speaker recognition, the sound

characteristics obtained from sound samples assigned to the same sound class are compared with each other. If possible, the audio features of the sound samples assigned to a different class are desired to be much different than each other. Audio data representation is the audio signal content consisting of numerical values expressed as features such as MFCC, audio spectrum, and etc. These features are based on displaying the signal energy, frequency distribution, and signal change over time [2].

In the literature, the process of audio event recognition in a given audio scene is taken from a couple of features which feed the deep learning mechanism as low level features. Since the semantic gap, this kind of pure low level feature usage remains quite limited as a tool to interpret the content for complete understanding. Shel et al. [17], proposed a model which is one of the examples for multimodal-based video prediction using deep learning. The idea behind their work is to create CNN [4] features from the frames of the video and give them to generated Long Short Term Memory (LSTM) [14-16] layers and to create MFCC features from the audio tracks of these video frames and give them to a generated different LSTM layers. Therefore, multimodal feature learning is applied into these two outputs and the classification process is done at the end. Jiang's dataset [18] is used for this model and 78.6% of accuracy is gained for the test dataset. Agrawal et al., proposed a CNN model using the ESC-50 dataset [19]. Only augmenting the data by using MFCC feature extraction, 68.40% of accuracy gained. Mushtaq et al., used different data augmentation techniques and combined them to build multiple models [20]. One of the data augmentation techniques that is used is Chroma [3] with Short Time Fourier Transform (STFT). Using a model which is pre-trained for deep learning is called Dense161 [21] and freezing/unfreezing layers method with using only Chroma feature based data augmentation technique, 46.80% of accuracy is gained for ESC-50 dataset and 70.50% of accuracy is gained for ESC-10 dataset. The ESC-10 dataset is created by using ten different audio classes selected from the ESC-50 dataset. To give more example to proposed model about environmental sound recognition with using ESC-50 or ESC-10 dataset, Khamparia et al. [22], proposed a model by getting spectrograms with log-mel features and giving them into a model constructed with Tensor Deep Stacking Network (TDSN) [22], 56% of accuracy gained for ESC-10 dataset [22]. The ESC-50 dataset's creator Piczak proposed a model by getting log-mel features and giving them to a CNN model in a related study, thus 81.5% of accuracy was gained from the trained model by using the ESC-10 dataset which is a reduced version of ESC-50 [23].

In order to bridge the semantic gap between low level features and the high level semantically rich features as human perception of action/event, recent works have introduced an intermediate representation between the audio features and the video's affective content. These methods construct representations based on audio features and NLP based on the speech to text translated transcription which employ these representations for affective content analysis and understanding of videos. Khalil et al. [24], proposed a model to recognize the emotion from input speech signals. Both machine learning and deep learning techniques are used but a model which is constructed with the Deep Convolutional Neural Network (DCNN) [15, 16] is performed much better than other machine learning algorithms.

In the literature, video summarization is a process which selects representative video frames to summarize key events present in the entire video. In our study, video summarization is used to localize the positions of most important actions and/or events in full length video. This process is similar to a combinatorial optimization problem which selects video frames due to action-basis to group them into meaningful segments as video clip. These kinds of video clips are a shorter form of original full length video, but it is more refined to represent a more compact and concise form of action/events as well. As told above, video summarization is addressed as an optimization problem in many contemporary studies [25-28] as well.

As told above, the accuracy from the proposed model of Piczak [23] was gained 81% of accuracy by only training the model using the ESC-10 dataset. At this point of view, unlike Piczak's proposed model, our model is trained with seven categories from the ESC-50 dataset which could be different categories from the ESC-10 dataset, and three extra categories which are designed and collected by us with new samples. Our gathered dataset is more complex than the ESC-10 dataset. Our work is proposed with a quite different performance evaluation scope than Piczak. Therefore, unlike all the



unimodal (e.g., audio signal, or spectrogram as still image) works that are explained about this section, our model strengthens the result of environmental sound classification with the support of NLP [7] techniques and different sound classes are used in the experiments.

The remaining sections explain the details about our methodology, and also, the different parts than other studies or superior sides of our study with related experiments.

### 3. Materials and Method

In order to build the multimodal deep learning infrastructure with audio, video and image processing, that has been described in this work, multiple libraries have been used that are created to use with the Python language [29]. These libraries have been selected by their properties, ease of use and some other criterias. First of all, to build all deep learning models, Tensorflow [30] and Keras [31] libraries have been used. And also, a pre-trained model which is ready to use for everyone in Keras library to build a deep learning architectural structure, MobileNetV2 [32] has been used as a base model for sound event recognition. Some libraries that allowed us to do some operating systems operations like creating a folder, deleting a folder, copying and pasting files, reading images and videos from folder and so on such as OS [29], Glob [29] and some libraries that help to do machine learning operations, mathematical operations, plotting operations, NLP operations, video operations and so on such as Pandas [33], Matplotlib [34], Seaborn [35], Numpy [36], SpaCy [37] and Pydub [38] have been used which their detailed information can be seen from Python website [29]. The OpenCV [39] library has been used to do reshaping, resizing and some other operations for frames of video clip as given as image sequence. To generate audio and video files, Moviepy [40] and Pydub [38] libraries have been used. The VOSK [12] library and a model from this library have been used to get the part of speech from a given video. For data augmentation methods, along with Tensorflow, OpenCV libraries, Librosa [41] library also have been used. For NLP and sentiment analysis operations, NLTK [42], SpaCy [37] and re [29] (Regular Expression Operations) libraries have been used to prepare text data for NLP methods and Transformers [43] library has been used to use Bidirectional Encoder Representations from Transformers (BERT) [43] models to do sentiment analysis. For NER operations, to score the similarity between the word in the sentence and the word in NER dataset, DiffliB [44], Zeyrek [45] and Jellyfish [46] libraries have been used. By using all these libraries, a suitable model and its infrastructure has been generated. To create Graphical User Interface (GUI), Gradio [47] library has been used.

The underlying methodology of our approach is described as follows. First of all, a video summarization technique has been generated to summarize the video to get scored action points from extracted video shots with related key frames of given video. After that, the sound event detection technique has been applied to do speech recognition from the above mentioned summarized video which is given as a short video clip with sound. The multimodal deep learning approach has been constructed to get speech from the video and do some NLP operations onto that speech to strengthen the result of sound recognition and to perform sentiment analysis. In the end, video summarization, sound event detection, NLP and sentiment analysis techniques have been integrated into a compact generated GUI that is presented for use by end users with online interaction capabilities as well.

#### 3.1 Dataset

In order to train the sound recognition model, data from ten different categories have been used. Seven of these categories have been chosen from a dataset that has been created using environmental sounds. The dataset that has been used to train the model is the Environmental Sound Classification called ESC-50 [48] dataset which is a labeled collection of 2000 environmental audio recordings. The dataset has five different main topic categories and a total fifty sound classes. These main categories are “Animals”, “Natural soundscapes & water sounds”, “Human, non-speech sounds”, “Interior/Domestic Sounds”, “Exterior/Urban noises”. Among these main topic categories, seven different categories from these main subject categories were chosen according to their evocation of places. These chosen sound categories are “Chirping Birds”, “Rain”, “Sea Waves” from “Natural soundscapes & water

sounds” main category, “Clapping”, “Laughing”, “Footsteps” from “Human, non-speech sounds” and “Car Horn” sound from “Exterior/Urban Noises” main category. For example, hearing chirping birds sounds may evoke natural forest to the people, hearing sea waves sounds may evoke seaside to people, hearing clapping may evoke somewhere where people are celebrating something in a place such as theater, concert etc. All of the sound clips are made of 5000 milliseconds and all of these sound categories have forty different sound clips. To expand the sound event categories of ESC-50, we collected some new audio data to generate three new sound event categories which contain different audio samples from ESC-50 original data. These three categories that have been used to train the model are “Football”, “Racing” and “Human Speech”. In order to collect data for “Football”, different sound clips have been cut such as fan chants, goal celebrating, missing goal reaction from fans, hit the ball sound, referee whistling and some other things. By considering the collection of samples of fan chants, the model finds a similarity between the “Human Speech” and “Football” categories. In order to collect “Racing” sounds, different sound clips have been cut as it was done for “Football” such as different engine sounds from different cars from racings, car sound from a long distance and short distance, sounds from pit stops and so on. And last, for the “Human Speech” sounds have been collected from a dataset called Librispeech [49]. Librispeech has samples that are 1000 hours total. Forty different english speeches have been selected randomly from this dataset. All of these categories have forty samples, and all of the samples are 5000 milliseconds long. In total, there are 400 counts and 2000 seconds of samples. Each sound sample is a single channel (i.e., mono) audio clip. These data were used in the training process of our audio-visual content understanding processes.

For testing our audio-visual content understanding methodology, we used a real world dataset called CPSM “sports minute” dataset [50] which has 74 videos collected from Youtube [51] in total with 10 different sports activities involving two years worth news highlights about college sports. Also, the proportions of each activity and number of labels per clip are variable for this CPSM dataset [50]. For “Turkish” videos to test the Turkish VOSK [12] model, “MediaSpeech” dataset has been used [52]. This dataset is generated by using short speech videos from Youtube [51] and some other websites for media videos by using news videos which are mostly about politics. The dataset has 4 different languages and Turkish is one of them. There are a total of 10 hours of speech for each language.

Above mentioned datasets are used in our experiments to achieve a reliable sound event recognition for analyzing the action topics in audio-visual content of given video which is based on our deep learning approach to understand the spoken content with its emotional scope.

### 3.2 Video Summarization

By considering the optimization methodology, the video summarization process is treated as an abstraction method which selects representative video frames to summarize key events present in the entire video. In the literature, recent studies on video summarization have learned how to select informative video subsets (i.e., video clips rearranged with selected video frames) close to summaries generated barehand by humans [53]. In the last few years, several studies have been presented [54,57] to solve the video summarization problem, and also, this problem is treated as a combinatorial and constrained optimization problem with the use of labeled full length videos. By detecting changes in visual features of video frames, Kernel Temporal Segmentation (KTS) produces segment boundaries. Video segment is built up with a bunch of video frames. Video shot involves above mentioned video frames grouped as an action related video segment. In this way, a video segment tends to be long if visual features do not change considerably [58, 59].

In our study with deep learning scope, in order to produce video summarization, it has been firstly done to get every frame from the video. These video frames have been reshaped as a tensor as (224, 224, 3) to be processed. Therefore, with the help of a pre-trained deep learning model called VGG19 [60], related visual features extracted from these video frames have been used to determine the change points in the whole video as treated as indicators of the variations of action or event locations [61]. The KTS process [62] has been applied to these obtained features for temporal segmentation. While ensuring that the summary length does not exceed a defined limit, video shots that involve actions are selected to generate a summary by maximizing the total scores. The summary length limit is circa 10%

to 15% of the full video length. Knapsack problem is a combinatorial optimization problem in computer science, which is known as NP-hard. The 0/1 Knapsack problem's dynamic programming solution is a more preferred solution in the literature for optimized video summarization as well. Above mentioned maximization step in the video summarization process is based on the 0/1 Knapsack problem. A near-optimal solution via dynamic programming can usually be obtained [63]. In the manner of combinatorics, dynamic programming requires an optimal substructure and overlapping sub-problems. These are present in the 0/1 Knapsack problem [64].

To ensure the selection reliability, the trained models' group with two parts as an integrated meta model was used to compute the necessary scores. The first part of the above mentioned model was used to extract visual features to comply with temporal segmentation as change points. The second part as a frame-relation learner part was to predict the frame selection probabilities as importance scores. The video shot level scores are computed by averaging frame level scores within the same video shot as well. By using both obtained importance scores and detected change points (i.e., temporal segmentation) within the scope of KTS process, thus related video segments needed for video summarization have been chosen. For long-term temporal dependencies between video frames and shots located far away each other, the above mentioned model's frame-relation learner part is used to build data for dynamic programming (i.e., Knapsack problem) that this part of the model is based on the bidirectional Long-Short Term Memory (Bi-LSTM) [14-16] and its structure was built on two Bi-LSTM layers which have 256 and 128 neural nodes, respectively. These layers work as a processor for revealing the forward hidden states and backward hidden states represented in a sequence of frames of video. Thus, this part learns the frame relationship with scores. At the end of this structure, a dense layer with 1024 neural nodes was used to classify the frames into some frame groups with frame probabilities. The results form these frame groups and video segments feed the dynamic programming infrastructure for solving the problem of Knapsack such as those given as input for Knapsack part. Thus, the whole video is summarized with the use of results as selected frames as "one" or not selected as "zero" by the dynamic programming solution of Knapsack.

In order to construct the video summarization in dynamic memory allocations, full length video has been split into smaller pieces. The reason for it was to split video into pieces to get optimal summaries separately, and thus, these summaries are concatenated to get a novel summarized video as a video clip as well. For the testing of video summarization problem solving processes, there are a bunch of video dataset with annotations in the literature. The SumMe is a video summarization dataset in which some unedited or minimally edited videos are presented as seen as 25 personal videos obtained from YouTube [51]. There are 15 to 18 reference summaries for each video which are individually annotated by human annotators [65]. The TVSum is another video summarization dataset in the literature. It has 50 YouTube videos with metadata that each one of videos is presented with a title and a category label. For every two seconds of each video, human annotated importance scores are provided to construct the reference summaries with a predefined length are generated. Therefore, as in the generated summary, these videos are separated into short video segments. To obtain a segment-level importance score, the importance scores within a video segment are averaged. At the end, a reference summary is generated by finding a subset of segments that maximizes the total importance score in the summary [61, 63].

In our study, we used the above mentioned video summarization meta model to test and validate our approach for summarizing the full (long) length videos that a summarized short video clip presents the important parts involving action/event related to the topic of spoken content. With these video summarization datasets, the performance of summarization was tested and validated that the overall success is used to show as indicator of optimal selection of frames for video clip involving salient action or events. The performance evaluation is based on the comparison between automatic summaries and ground truth summaries. In our study, an average performance score (i.e., *F-metric*) [3] was obtained as equals to 51% with the summarization tests on above mentioned TvSum and SumMe datasets with the integrated meta model's VGG19 [60], Bi-LSTM [14-16] and dynamic programming parts as told above. This performance score is the same as given in Zhou et al. [61] study that they obtained this kind of average score in their study as equal to 50% (i.e., for SumMe as 42.1% and TVSum as 58.1%). The detail of this metric is given in the following sections. Since the experiments

show that our video summarization stage of the complete audio-visual content understanding approach (as given as “*Proof of Concept*”) is more accurate to detect the action/event location in a given video, the summarized video as a video clip is more efficient in terms of computational complexities and space complexities.

### 3.3 Sound Event Detection and Speech Recognition

The preparation part is the most challenging part in this multimodal data processing and underlying learning process. With a raw form, classifying the labels of chosen sound classes is very hard by using deep learning. Because spectrograms of different sound clips from different classes may show a similarity which is very difficult to separate them because they are very similar to each other in manner of feature extraction and process. In order to use the model efficiently, different techniques might have applied to the spectrogram such as attention based technique. In this model, the multiple masking [67] technique has been applied to increase the distribution of the dataset. At first, with the help of the Tensorflow library, data in process that is in an audio shape in the dataset has been transformed into spectrogram shape as an image. These spectrograms have been generated by using log-mel features of the audio. Then, multiple data augmentation methods have been used to increase the data variations in amount and presentation for the dataset, and multiple models have been also generated. To give examples for these data augmentation methods, as seen in Figure 1, applying MFCC type feature extraction method [2], applying Chroma type feature extraction method [3], applying Contrast Limited Adaptive Histogram Equalization (CLAHE) [66] to spectrogram images, applying multiple masking to these images, adding Gaussian noise (background noise) [3] to audio segments, adding shifting method to audio segments, getting log-mel spectrograms using Librosa library [41], applying some operations to these images such as flipping, rotating, sharpening and all combinations of these method have been used. To test these rebuilt and extended datasets, multiple programmatic structures have been generated. If an overfitting problem for model training has occurred, the programmatic structure has been regenerated in order to prevent this overfitting. The best result gained from multiple training processes, audio dataset augmented by background noise and image dataset increased by CLAHE and multiple masking methods. In other words, if it is said in an objective evaluation manner in terms of performance metrics for machine/deep learning processes that have been used for the training process, the highest accuracy and the lowest loss has been gained from this training process.

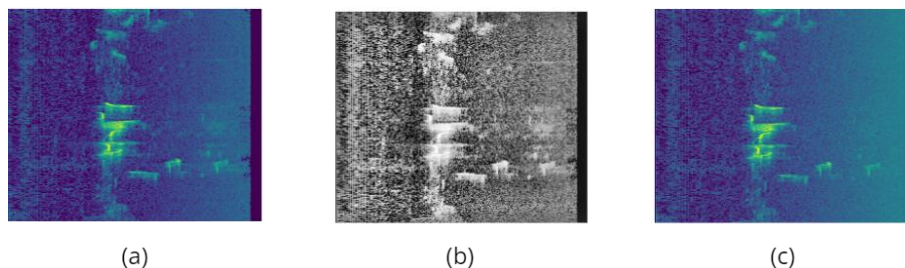


Figure 1 An example of a spectrogram from the Chirping Birds category (a). CLAHE applied (b), Gaussian Noise added (c)

MFCC feature extraction is a method which is a very popular method to be used for feature extraction of an audio file. One of the reasons why MFCC is used so much is that MFCC coefficients are much less affected by changes, audio wave structure [68]. But it is often used in speaker recognition applications. MFCC coefficients are obtained by de-correlating the yield log energies of a channel bank which comprises triangular channels, straightly dispersed on the Mel recurrence scale [69]. There are some libraries that help to apply MFCC type feature extraction to spectrograms, but Librosa library is the most used one.

Chroma feature extraction is a method which is used for obtaining the quality of a pitch class alluding to the color of a musical pitch of a given audio file which has a sampling rate [2, 3]. This pitch can be decomposed into an octave invariant value called chroma, and there is a pitch height value which shows the octave the pitch is already in. Each frame of audio is windowed, by this scope, a short-time



audio window is presented by Chroma that window has the harmonic content such as keys or chords, and etc. Chroma's related feature vector is extricated from the magnitude spectrum with the help of a short time fourier transform (STFT), Constant-Q transform (CQT), Chroma Energy Normalised (CENS), respectively [2, 3, 70].

CLAHE is a different strategy which is used in images. It is used to improve image contrast. To limit the contrast amplification, CLAHE aims to avoid amplification. By considering the transformation of slope function, amplification contrast around the given pixel value is determined. By clipping, the CLAHE limits introduce amplification. It takes a basis which has a predefined value in the histogram prior to cumulative distribution function (CDF) calculation [66]. The main idea of applying it is to improve the contrast of the parts where audio-wave is denoted in the histogram, meanwhile decreasing the contrast of the other parts. With this way, the model can be focused more to the parts where it is wanted the model to focus. The OpenCV [39] library has been used to apply CLAHE to spectrograms. In the manner of sound recognition, a deep learning model is constructed to classify and determine the sound classes that it is based on above mentioned techniques and processes. This model and its multimodal aspect's usage is explained at the following sections of this study as well.

Audio shifting [3] is another method which is used for data augmentation by shifting the audio signal content  $n$  steps ( $n$  denotes time) and saves shifted audio file as a new audio file. Overflowing audio data from the end of this audio file is cut and added to the beginning of the shifted audio file. With this way, limited data is being increased and with not using the same audio file, possible overfitting problem may be prevented.

Flipping, sharpening, and rotating a spectrogram are the techniques which are used to increase the amount of the dataset. Obtained spectrograms are shown as matrices while programming. Each spectrogram denotes the  $m \times n$  matrix by its width and height. If it is said a spectrogram is shown as a  $4 \times 1$  matrix  $[1, 1, 0, 0]$ , flipping this matrix will be resulted as  $[0, 0, 1, 1]$  which means horizontally reversed. Rotating an image is used to rotate an image by a given degree. A matrix that has  $x$  and  $y$  is the image. Multiplying a matrix that has sine and cosine with the image matrix is a rotated matrix that shows the degree of how many degrees the image is rotated [71]. Sharpening is a method to remove blur, enhancing details and dehazing. There are some methods that help to sharpen an image. The used sharpen matrix is given as  $[-1, -1, -1; -1, 10, -1; -1, -1, -1]$  as a two dimensional matrix. Finally, here is Figure 2 that has an example that each sharpening, flipping and rotating are applied to a spectrogram from the chirping birds category. With the help of trigonometric functions, a rotating process is applied where the rotating angle theta is 45 degrees. All of these processes have been applied using OpenCV library functions and all of these processes have been applied after applying CLAHE method to every spectrogram which can be seen from Figure 2 below.

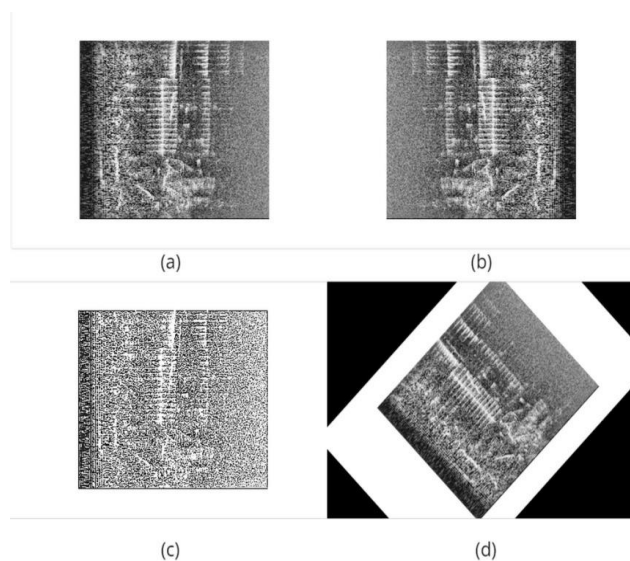


Figure 2 Examples of every method applied to a spectrogram. (a) denotes normal spectrogram, (b) denotes flipped spectrogram, (c) denotes sharpened spectrogram and (d) denotes rotated spectrogram

At first, after the dataset has been gained, background noise [3] has been added to all of the audio files and has been saved as another audio file in order to augment the dataset. This background noise audio clip is a clip that has Gaussian noise in it for one minute long. In order to add this clip to audios, a 5000 millisecond long sample has been taken from the Gaussian noise clip. Then, this clip has been applied to audio segments and has been saved by using Librosa [41] library. With this process, the dataset has increased two times of its original amount. The Gaussian noise formula can be seen as denoted by Eq. 1 as below where  $z$  represents the gray level,  $\mu$  represents the mean gray value and  $\sigma$  represents its standard deviation.

$$\text{GaussianNoise}(z) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(z-\mu)^2}{2\sigma^2}} \quad (1)$$

Another method which has been used to augment data is applying the multiple masking method. It is a method which is used for data augmentation and to get a sort of mixed data. It is split into two steps as applying frequency and time mask by the given parameters (i.e., consecutive time frames  $T$ , consecutive frequency bins  $F$ ) as the first step and mixing them as the second. [67]. The frequency mask is based on a masking method which is applied to the spectrogram which is in a frequency zone. With this method, frequency channels are masked. This mask is chosen from a random uniform distribution between 0 to  $F$  frequency value. However, the time mask is based on a masking method which is applied for masking consecutive time steps. This mask is chosen from a random uniform distribution between the number of frames 0 to  $T$  frames as presented as time steps in milliseconds. According to a parameter that denotes how many times this frequency and time mask will be applied ( $t_0, f_0$ ) these masking processes are applied. After parameters are given, the distribution for training purposes is expanded by drawing the shear plane from random parts. There finishes the first step. Thinking about a spectrogram that has been generated by its frequency/time features, horizontal shear plane denotes time mask and vertical shear plane denotes frequency mask. Given parameters to perform the multiple masking process that have been chosen from different trials can be shown as Table 1 below.

Table 1 Given parameters to perform Multiple Masking

$T$	$F$	$t_0$	$f_0$
24	36	2	2

The second step begins where the first step ends. This step is about mixing spectrograms which frequency and time masks have been applied as seen in Figure 3. The most important part about mixing spectrograms is to decide which categories have to be mixed. In other words, mixing which categories to augment the data helps to build the most efficient sound recognition model. In this part, categories which are shown the most similarity between their spectrograms are chosen to be mixed. For example “Footsteps” and “Chirping Birds” sounds are both a sound which their spectrograms have similarity in the same frequency. The other categories which have chosen to be mixed are: “car horn” - “clapping”, “football” - “racing”, “human speech”- “sea waves” and “rain” - “laughing”. After applying the multiple masking process, the amount of dataset has increased, and the distribution provided.

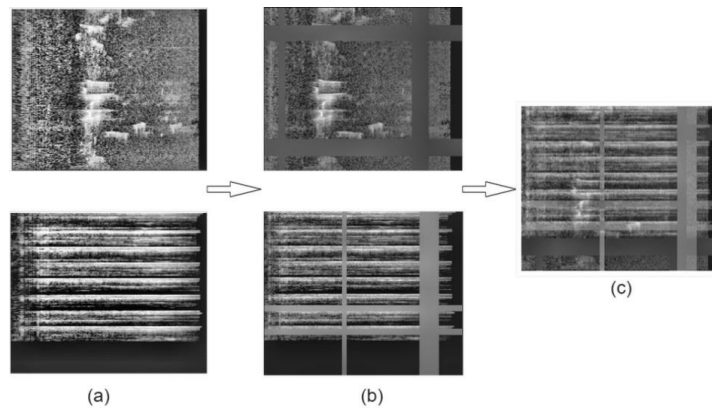


Figure 3 Example of applying multiple masking process in “Chirping birds”- “Footsteps” categories. (a) denotes normal spectrograms. (b) denotes multiple masking applied spectrograms and (c) denotes mixed spectrogram

### 3.4 Multimodal Deep Learning Methodology

The deep learning methodology has some superior properties among other machine learning paradigms [15]. For a learning system, representation learning is an important process that helps to discover the feature patterns in given data [72], and also, it automatically proceeds to discover the representations needed for feature detection, selection or classification from given data. For image based learning, the semantic gap means that there is a difference between primitive (low level) features of an image and semantic meanings that humans recognize from the image. With the help of the representation learning to achieve a successful understanding of content, the semantic gap between low level features and semantically rich high level features would be filled. In the deep learning concept, to fill the semantic gap between low level features such as color, texture, shape etc. and high level semantically rich features, representation learning is used. With the use of transfer learning, representation learning usually can produce a good result of content understanding. To solve a classification problem in solving another similar problem, transfer learning is used as a training style where the original problem’s knowledge and experience are used to solve this similar problem. In this scope, our base model for sound event recognition is constructed with MobileNetV2 [32] which is trained for by using 1000 classes of ImageNet dataset [73]. Its knowledge about learned salient features is transferred to our integrated model with the transfer learning process, and then we apply the fine tuning over this kind of base model as shown in Table 2.

Video summarization is used to generate a short synopsis. By the manner of selecting the video's most informative and important parts, this synopsis summarizes the video content. To form a shorter video, this summarized video is usually composed of a set of representative video frames (i.e., video’s keyframes), video fragments or video segments (i.e., video’s shots) that have been stitched in proper chronological order [54, 55, 59]. In our study, video summarization is built on the dynamic programming that solves a combinatorial optimization problem treated as a knapsack problem [64] for video’s action parts selection as given as frames in the video.

Our audio-visual content understanding methodology starts with obtaining the above mentioned summarized video and sound recognition process applied on this video. This methodology works with the following steps. At the first step as Stage I in Figure 4, the relevant video summarization datasets were used to train a meta deep learning model to detect parts of the video involving action to summarize the video. In the scope of KTS process [62], related video segments (i.e., video shots) needed for video summarization have been chosen based on change points that have been detected. This meta model was built with VGG19 [60] and Bi-LSTM [14-16] parts as told in the above sections of this study. After this kind of summarization training, the original full length video may come as an input when it is used to summarize with the use of shot level scores as average frames scores. After these processes, the summary has been obtained by using *Dynamic Programming Solution* for Knapsack Problem. Since action/event location detection is applied to this full length video to split the parts of this video as given input for Stage III in Figure 4 as told below that involve most important

actions or events in it, and then these multiple parts or just one splitted part get concanarated as a new video clip with selected new frames in a summarized manner. At the second step as Stage II in Figure 4, the audio dataset which is used for training the sound event recognition model, has been illustrated as spectrograms and increased by data augmentation techniques such as CLAHE [66], gaussian noise [3], and multiple masking [67]. With this way, the data has been prepared for training the model. Then, this audio dataset has been given to the MobileNetV2 based integrated deep learning model for training and the appropriate model has been produced to use for predicting sound classes. After this point, our content understanding process also has two complementary steps. At the first step of this process as Stage III in Figure 4, first, the original full length input video can be processed to be summarized as a video clip with newly chosen frames. These frames can also be chosen by producing a summary at Stage I using dynamic programming as well. Therefore, the sound recognition process is applied to this new video clip to recognize the sound classes of the environment. In order to strengthen the result of sound recognition, a content understanding process is defined to compile the speech from the video taken as plain text in the form of a script. After the prediction results, NER [10] techniques are applied to the script of speech that is taken from the new video clip to categorize the words or phrases from the speech if these words or phrases evoke any of the words or phrases that are prepared for each category owned. With this way, the sound recognition result is getting stronger. At the second step of this process as Stage IV in Figure 4, our contribution basis on the sentiment analysis is applied to the speech that is taken from the new video to determine whether the sentiment of the sentence is positive, negative or neutral. In Figure 4, a general presentation of our deep learning based audio-visual content understanding methodology is figured out which is explained as above.

For sound processing, we applied data augmentation and then data preprocessing methods, in order to generate an efficient model based on the training of a sound recognition model. The training data has been split as 70% train, 20% validation and 10% test part. Our integrated sound recognition network structure that has been generated programmatically to train the model can be shown as the structure as given in Table 2. The training process of the integrated sound event recognition model has been split into two subparts. First part is started by deploying a pre-trained model from Keras library MobileNetV2 to the generated model. MobileNetV2 is a convolutional neural network architecture with 53 layers equipped with a deep learning concept. At a glance, it is based on an inverted residual structure where the residual connections are between the bottleneck layers of a given network. Furthermore, it seeks to perform very well on mobile devices with low memory consumption. Its model works with a specified resolution for a given input image which has exactly 3 input channels as (224, 224, 3) [32].

Before starting the training process, the added layers from MobileNetV2 have been frozen. The model has been compiled by using batch size equals 8. In Table 2, the architectural structure of the integrated network model for sound event recognition can be seen.

Table 2 Architectural structure of the integrated sound event recognition network model

<b>MobileNetV2 and Convolutional Structure</b>
MobileNetV2
GlobalAveragePooling2D
512 Dense-ReLU
Dropout 0.5
256 Dense-ReLU
10 Dense-Softmax



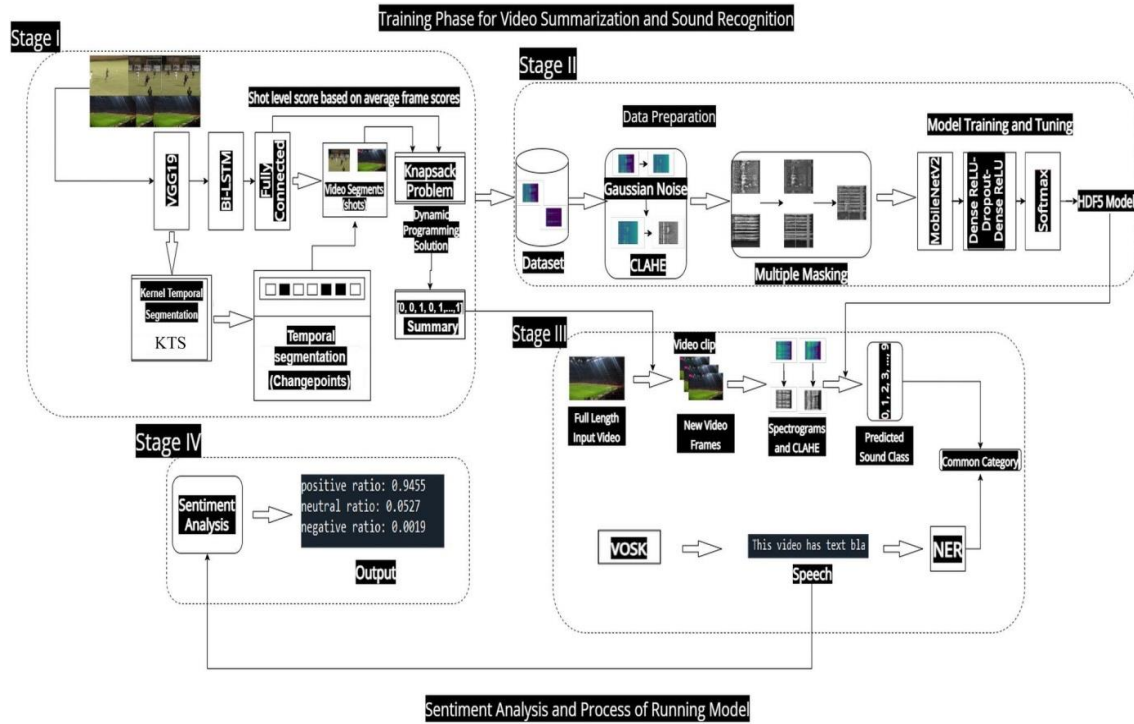


Figure 4 General representation of our deep learning based audio-visual content understanding methodology.

Adaptive Moment Estimation (ADAM) is an optimization method that keeps an exponentially decaying average past gradient to the similar momentum value. In this way, it computes adaptive learning rates for each parameter as well [74]. In our study, the ADAM method is used by us as optimization function and its determined first learning rate is given equals as  $2 * e^{-5}$ . Categorical cross entropy is used as the loss function and the training process lasts 100 epochs. Model's checkpoint record has been enabled so that the optimum part or training model can be taken before the overfitting problem occurs. Therefore, the *GlobalAveragePooling2D* method is applied to use average pooling on the spatial dimensions. A dense layer that has 512 neural nodes and Rectified Linear Unit (ReLU) activation function in it is added to this structure. ReLU function is a function which is commonly used in hidden layers [15, 16]. A dropout process has been applied to the connections between 512 neural nodes layer and 256 neural nodes layer to find the strong connections between each layer and to prevent possible overfitting problems. Another dense layer which has 256 neural nodes and again the ReLU activation function has been added. And the last dense layer which has 10 neural nodes denotes the output layer. This layer is where the classification happens. Each layer denotes categories owned. As having ten categories, there are 10 neural nodes at the output layer. The activation function that has been used at the output layer is the Softmax function which is commonly used in the output layer as the classifier. It turns a vector of  $K$  real values into a vector of  $K$  real values that sum to 1. This  $K$  term usually denotes the number of classes in the multiclass classifier. Its normalization properties ensure that all the output values of the function will sum to 1, The softmax function transforms the input values into values between 0 and 1. The equation of Softmax function can be seen in Eq. 2. In Eq. 2, the  $s_i$  values are the elements of the  $s$  input vector and can take any real value, The term on the bottom side of the Eq. 2 is the normalization term [75].

$$\text{Softmax}(\vec{s})_i = \frac{e^{(-s_i)}}{\sum_{j=1}^K e^{(s_j)}} \quad (2)$$

Beyond the MobileNetV2 as the second part of Table 2 denotes the fine tuning part of network architecture of our work. This time, the frozen MobileNetV2 layers have been unfrozen and the model's fine tuning process has begun. Only different parameter between training and tuning is the learning rate. In the tuning part, learning rate has been chosen as  $1 * e^{-5}$ . After 100 epochs of the tuning process, the model has been saved with h5 file extension. The Hierarchical Data Format version 5

(HDF5) is an open source file format, and also, it is given as "h5" formatted file in our study. It is capable of involving complex, large heterogeneous data [76]. In our experiments, the model has been trained and tuned about 200 epochs and after the processes have been done, the incoming result has not been a good result and the model has started to overfit. Also, a checkpoint has been used to save the model while the model has the minimum validation loss value. This checkpoint has saved the model structure with its weights in the tuning process while it has been in the 48th epochs. It could have been seen from the plots given at the experimental results section, the model also has started to overfit after 48th epochs. In this way, the training and tuning processes have taken 148 epochs in total.

With the help of NER [10] techniques, the result that is incoming from the sound recognition process has been strengthened. In order to use these techniques, the first thing that has been done is to get the data that evokes any of the categories owned. For example, "shoot", "goal" from "football" category, "Lewis Hamilton", "drifting" from "racing" category, "sparrow" from "chirping birds" category, "applause" from "clapping" category can be given as some examples. These data have been collected for both "English" and "Turkish" languages. There are at least 50 words and phrases that could evoke any of the categories owned. Gained text from the video that has been taken by using VOSK [12] library has been processed by some NLP [7] techniques. First, if the language is "English", with the help of SpaCy [37] library, the words from the text have been lemmatized which means all the suffixes from the word have been expelled. For example, the present tense's token "-s" has been expelled, and the past tense's token "-ed" will be expelled. Also, if the word is an irregular verb, the word has been transformed into its base form. All the letters from a word have been transformed into lower case. Characters such as ".", ",", etc. and numbers have been removed. It is a little bit different in "Turkish" language because there is a lack of libraries for Turkish NLP techniques and Turkish language is an agglutinative language. A word could have more than one stem and it could be a bit hard to find which stem created the word. Also, a new algorithm has been created for this case. This algorithm works as follows. First, with the help of Zeyrek [45] library, all the stems for a word have been found and words from the sentence have been split into a list. Among these stems, if the word has more than one stem, the lowest length between word and the stem most likely denotes the correct stem of the word. Otherwise, if the word has one stem, this stem has been selected as the correct stem of the word. After that, the same processes that have been done as the "English" sentence have been applied such as transforming lower case, expelling the characters and numbers. After applying the preparation of the sentence processes, the process of matching words and phrases from the sentence between words and phrases from the previously prepared NER dataset has been started. Matching words have been controlled like this: Because of having a maximum of six words of phrases in the NER dataset, the first word has been selected first and it has been controlled if it is matched word from the dataset. Then, the second word has been added to the first one and these two words of phrase have been controlled and this matching process has been continued like this starting from the first word and by getting six words of phrases. Later that, the second word has been selected and it has been controlled with the post-added five words until the last word. Applying this process to the "English" words is easy. Because words from the dataset and words from the sentence are without suffixes. So, the matching is easy. But it is hard for "Turkish" words. So, another algorithm has been generated for the Turkish ones. This algorithm depends on similarities between the word or the phrase from the sentence and the word or the phrase from the dataset. In order to calculate similarities, four different calculating methods have been used. First method is from the DiffLib [44] library and it is used to find the similarity between two strings by using similarity ratio [77]. It is calculated by using the formula of ratio as given in Eq. 3, where  $M$  denotes matches and  $T$  denotes the total number of elements.

$$\text{Similarity ratio} = 2.0 * \frac{M}{T} \quad (3)$$

The second method which is used to find the similarity score is by the help of Jellyfish [46] library, it is used to find the Jaro [78] distance between the words or the phrases. It depends on words' length and the number of matching characters. It returns 1 when all the characters between two strings are the same and in the same order. But, if all characters are the same but not in the same order or not all the characters are the same, transpositions between words are taken. In order to take transpositions, the

matching words are controlled. The half of the number of the unmatching words in the same index denotes transpositions [78]. The third method is used with the help of NLTK [42] library and it is used to calculate Bilingual Evaluation Understudy (BLEU) Score [79]. The method which is used to calculate BLEU Score is Sentence BLEU Score. To find the BLEU score, the number of candidate words that are in the reference word are divided with the number of total words in the candidate word. N grams term denotes that the words that are the same and match in the same order. Every match changes the N. When the possible proposing high-precision hypothesis translations are to short, to compensate them, a brevity penalty (BP) is given as calculated in Eq. 4 as below [80].

$$BP = \begin{cases} 1 & \text{if } c > r \\ e^{1-r/c} & \text{if } c \leq r \end{cases} \quad (4)$$

where the length of the corpus of hypothesis translations is denoted by  $c$  term, and  $r$  is the effective reference corpus length. The BLEU Score is given in Eq. 5 as it calculated as:

$$BLEU \text{ Score} = BP * \exp\left(\sum_{n=1}^N \omega_n \log p_n\right) \quad (5)$$

In Eq. 5, each modified N gram precision  $p_n$  is combined up to length N and can be weighted by specifying a weight  $\omega_n$ ; this formula is used in our study as well. In order to smooth that match, there are some smoothing functions. In this work, a related method for smoothing functions has been used. This function does that instead of scaling the similarity score between the words by using the multiplicative inverse of  $2^m$ , where the  $m$  term is word size, it uses the multiplicative inverse of the natural logarithm value of the length of the translation. And the last method as fourth has been used to find cosine similarity. It uses two vectors. The first one is generated by if the word from the first sentence, phrase etc. exists in the other sentence, the value of that index from the first vector becomes 1 else it becomes 0. The same process is made for the other vector [81]. Then, by the help of Eq. 6, the cosine similarity score is found. A and B denote these two vectors.

$$\text{Cosine similarity} = \cos(\theta) = \frac{A \cdot B}{\|A\| \|B\|} \quad (6)$$

All of these similarity score metrics return a value between 0 and 1. In order to find the similarity between the word or the phrase from a sentence that has been gained from the video and the word or the phrase from the NER [10] dataset from these collected values from four different metrics, the minimum and the maximum of them have been selected in our study. Therefore, the harmonic mean has been calculated and if the calculated value is higher than the threshold that has been assigned, it is said that there is a similarity between them. This threshold value has been assigned as equals as 0.6. At the end, if the model finds a topic from the NER dataset which is spoken in the video, the model will give it as an output to detect the related category. This output is limited by showing only the top-first two of the most spoken topics.

With the help of sentiment analysis, the sentiment of the sentence has been determined. In order to determine it, previously prepared BERT [82] techniques with related models have been used. These BERT models have been taken from the HuggingFace [83] services. As a platform, HuggingFace helps users to create their own interface in which pre-trained deep learning models can be tested by other end-users. For “English”, a model which is named “Twitter Roberta Based Sentiment” has been used [84]. This model is generated by using 58 million tweets. For “Turkish”, a model named “Bert-base Turkish Sentiment Model” has been used [85]. For training this model, 5331 positive and 5331 negative sentences from different comments of different websites and a dataset which is generated for collecting tweets from twitter have been used [86]. In order to apply this model to sentences, first words from the sentence have been prepared. This preparation process includes having lower case letters, deleting all characters and numbers from words and deleting words which are tagged as stopwords. Stopwords are words that do not make sense in a sentence. Also, it is preferable to discard such words rather than occupy space in the sentence [87]. As an example of these words for “English” language “the”, “a”, “my” and for “Turkish” language “ama”, “her”, “ne”. After these processes, the sentiment analysis process has begun. Sentiment analysis model returns one of three labels based on whether the sentence is “Positive”, “Negative” or “Neutral”. But the result returns as probability. The probability of being “Positive”, “Neutral”, “Negative” can be seen as a result.

The Association Rule Mining (ARM) [11] is used in our study to determine the relationship between categories used in different videos. It is used for the model to comment. In order to apply ARM, the Apriori Algorithm is used [88]. The Apriori algorithm is one of the most used techniques for ARM. The reason for using Apriori, it is a successful algorithm to reveal the connection between categories. In order to apply Apriori, firstly, minimum support number equals 0.01 and minimum confidence number equals 0.2 has been selected. Support value of all categories has been found. If there is a value lower than minimum support value, this category has been disabled. Dual partnerships have been generated from remaining categories. If there is a partnership that has a value lower than minimum support value, this partnership has been disabled. Then, triple partnerships have been generated and so on. The algorithm goes like this to find multiple partnerships between categories from a dataset that has been generated before.

The experimental results of our study are presented in the following sections as well as some of them are also obtained with the use of graphical user interface (GUI) on the basis of open source Gradio interface. Gradio is a web based interface to deploy and test machine learning models with user interactions [47]. For testing purposes, our “*Proof of Concept*” demonstration is introduced with the Gradio web interface as given in the Github repository [89]. The end-users can effortlessly reach and use interactively from the web site of our GUI based application as given in the Github platform. It is called “*Audio-Visual Event Sentiment Analysis*” (AVESA) that works with sample videos and uploaded videos by end-users as well [89].

Our GUI is presented in two sections: the first one (i.e., left hand side at Figure 5) is a video input container to acquire video from a file, and the second one (i.e., right hand side at Figure 5) is able to show the experimental results and the output video as given in Figure 5 as given below. By dragging or clicking to the section for uploading video, the video can be uploaded. Therefore, the language should be selected from the section for selecting language. The language of GUI depends on the user’s computer language. By clicking the “*Yükle*” or “*Upload*” button, the process will begin. By clicking the “*Temizle*” or “*Clear*” button, the video can be deleted from the section for uploading video or after the video is uploaded, the button “*X*” from the top of the video can be clicked to delete the video. After the processes are done, the output will be shown in the second side (i.e., right hand side). All figures related our web GUI are taken from our web GUI application in its original form which is based on the open source Gradio interface Python library [47].

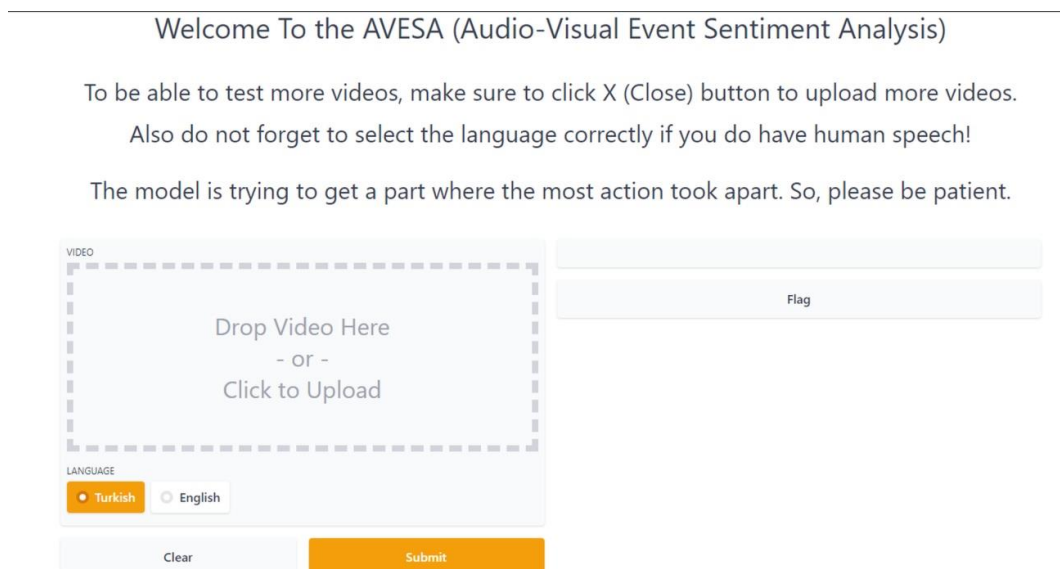


Figure 5: Our graphical user interface in action to recognize the sound event from video [47, 89]

In addition, videos that were chosen by us before can be used by clicking any of the videos below the “*Examples*” or “*Örnekler*” section. There are three different videos chosen by having any of the two languages. By clicking any of them, results for any of them can be seen at the second section. This section can be seen at Figure 6 below.



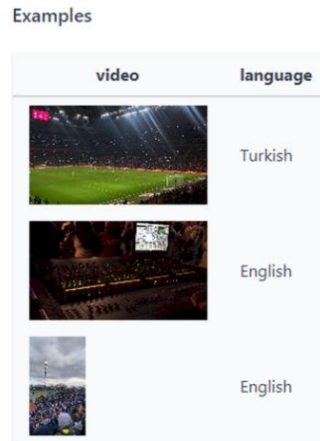


Figure 6 Examples shown by our graphical user interface [47, 89]

By considering the GUI, it seems to be able for an end-user to interact with a specific or randomly chosen attribute of our deep learning based audio-visual content understanding approach. By this way, it can be concluded that a sound event's outcome (i.e., class) and the mostly correct sentiment from this event is determined by our system which is shown in our GUI [89].

#### 4. Experimental Results

The experimental setup of our study is applied to establish our deep learning based audio-visual content understanding concept. In our experiments, we used a mix of sounds from a subset of ESC-50 [48] and other sound samples to detect and recognize the sound event classes. In other words, we applied an infrastructure to handle the above mentioned dataset which explained in Section 3.1. After that, we used some videos to test our video based NLP [7], NER [10] and other speech to text infrastructure of our methodology. These videos were chosen from CPSM dataset [50] and no-copyrighted materials from the Internet as provided on the Pexels website [90]. Since our real world problem, we are addressing is based on the real time speech to text translation and sound event recognition, the complete model design and its work on the knowledge discovery and processing results to a certain event class decision with the help of salient emotional situations come to a conclusion with underlying sentiment analysis.

In our experiments we used some computers equipped with Intel Core i7-8750H CPU that works with 2.20 GHz and has 16 GB RAM memory, NVidia GTX1060 graphic card (GPU) with 6 GB RAM and Intel Core i7-9750H CPU that works with 2.6 GHz and has 8 GB RAM, NVidia GTX1650 GPU with 4 GB RAM. In addition, the experiments were carried out with some experimental setups that were run several times on these machines for training and testing purposes. Our underlying software platform is based on some well-known libraries and frameworks which are supported by the Python programming language such as Keras [31], Tensorflow [30], NLTK [42] and etc.

In the following subsections, we address the objective performance evaluation via metrics for speech recognition, deep learning based classification and appropriate sentiment analysis. In addition, these kinds of evaluation in this section are supported with related experimental results given in detail. With the help of the experimental results, we clarified our concept and proof it as a “*Proof of Concept*” as well.

##### 4.1 Performance Evaluation

In this section, we introduce the objective performance evaluation based on metrics for sound event classification, speech recognition and appropriate sentiment analysis. Accuracy rate is one of the objective criteria commonly used to determine the class discrimination ability of the classifier on the dataset in an experiment. According to the confusion matrix table in the literature, it is evaluated by

true positive ( $TP$ ), false positive ( $FP$ ), true negative ( $TN$ ) and false negative ( $FN$ ) measurements. This value can be shown as both decimal and expanded as a percentage value in studies. It is given as Eq. 7 below [91].

$$Accuracy = (TP + TN)/(TP + FP + TN + FN) \quad (7)$$

The Precision metric expresses how many of the values estimated to be positively labeled are actually positively labeled in the experiments. This is shown in the Eq. 8 as below [91].

$$Precision = TP/(TP + FP) \quad (8)$$

The Recall metric denotes how many values the model predicted positively while they should have predicted as positively labeled in the experiments. This is shown in the Eq. 9 as below [91].

$$Recall = TP/(TP + FN) \quad (9)$$

The  $F1$  Score is known as the harmonic mean of precision and recall metrics. In the literature, it also called  $F$  measure,  $F$  metric or  $F$  Score as well. It creates a useful evaluation result in order to consider the extreme situations in the experiment where performance is affected. This is shown as Eq. 10 [91].

$$F1 = 2 * (Precision * Recall)/(Precision + Recall) \quad (10)$$

The Word Error Rate (WER) Score is one of the other common metrics to calculate accuracy between words. In performance evaluation scope, the WER can be found by dividing the number of errors to total words. These error numbers can be said as the addition of the substitution number, the addition number and the deletion number. The *Substitution* (S) occurs by replacing a word with another. As an example of this, transcribing “noose” as “moose”. The *Insertion* (I) occurs by adding a word which was not said before. For example, transcribing “hostess” as “host is”. The *Deletion* (D) occurs by deleting the word from the transcript. By considering this, transcribing “let it burn” is converted to “let burn”. Therefore, the sum of S, I and D terms denotes the total number of errors. Dividing this value to the total number of words will give the WER Score [92].

Above mentioned metrics and measurements are used in our study to prove the consistency and reliability of objective performance evaluation of our deep learning base audio-visual content analysis approach.

#### 4.2 Experimental Results with the Proof of Concept

In this section, the experimental results of our deep learning based approach to understand audio-visual content from videos are presented in an objective evaluation manner with plots and metric values which are well-known in the literature. Our experiments conducted on the above mentioned computer and software equipped with appropriate infrastructure. Every experimental setup was tried at least on 5 different runs to ensure the reliability of the test. Firstly, the training and validation of our sound recognition model and its fine tuning procedures results are given. In this scope, the changes of accuracy value while model tuning and training processes is given as accuracy plot in Figure 7 below. In our training process of sound recognition, loss function is treated as an objective function to analyze and reduce the average error that occurs in learning with given epochs for deep neural networks. The loss function’s plot from the training and tuning process can be seen in Figure 7 below.

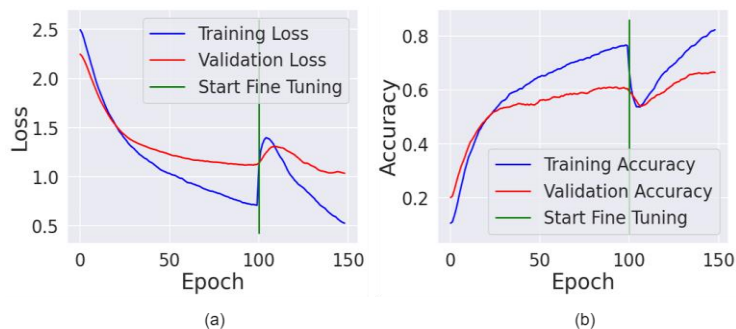


Figure 7: Loss plot (a) and Accuracy plot (b) for sound recognition model training.

It can be clearly seen from Figure 7, when the accuracy is increasing, the loss value at related epochs is decreasing as well. Values which are training loss, training accuracy, validation loss and validation accuracy started in the first epoch as respectively: 2.39, 0.10, 2.24, and 0.20. And it became the 148th epoch as respectively: 0.48, 0.83, 0.99, and 0.67. With the help of the beginning of fine tuning stage, the loss values a little bit increase for a number of epochs, but then it decreased as fast as possible than more before fine tuning point of training process. After the training and validation process, the model obtained an accuracy value as 0.70 and its respective loss values as 1.00 in the manner of fine tuning. Macro Average and Weighted Average values are equal to 0.73 for precision, 0.70 for recall and 0.70 for *F1* Score, where total sample test size equal to 120. Therefore, it can be said that the model has performed quite well with the result of 70% accuracy in terms of percentage ratio.

By considering the objective performance metrics in testing of our model, the performance results of sound recognition and confusion matrix for sound recognition in the experiments in our study can be seen from Figure 8 to Figure 9. Every class has an outcome in training with 12 samples as given in experiments. Total test sample size is defined as 120 samples for ten classes of overall testing.

In this sound event recognition’s performance evaluation, the average precision, average recall and average *F1* Score are obtained as 0.72, 0.70 and 0.69, respectively.

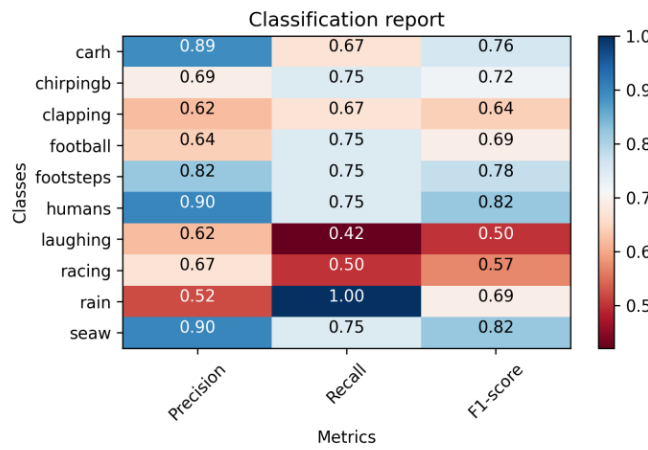


Figure 8 Performance results for sound recognition

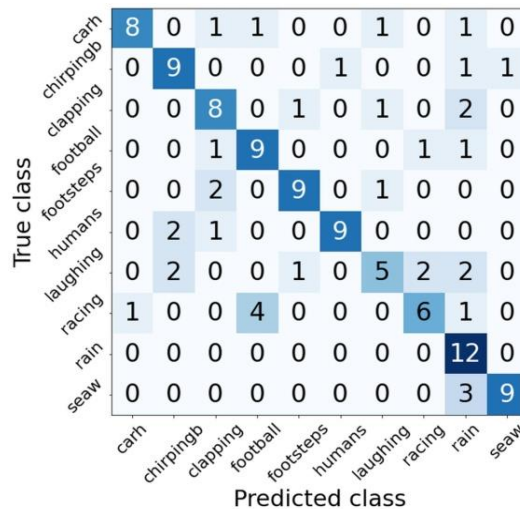


Figure 9 Confusion matrix for sound recognition

In addition, the performance results of getting speech with VOSK [12] language models for both languages in the experiments in our study can be seen in Figure 10. As seen from Figure 10, these tests for “*English*” texts have been done by using five different videos which are randomly chosen from the “*English*” speech dataset [50]. As seen from Figure 10, for “*Turkish*” texts have been done by using

four different videos which are randomly chosen from the “Turkish” speech dataset [52]. According to the sentiment analysis reports given in these figures, the best results in the experiments are obtained with “video-4” for “Turkish” texts, and “video-2” for “English” texts as well. In these tests, we assemble independent ground-truth text for each one of the videos in the test for NLP [7] and NER [10] processes that these texts are used to determine the similarity between predicted results and original human understanding of the spoken content with some metrics. These test results are given in Figure 10 as below. By considering the above mentioned VOSK speech tests for English videos, the average precision, average recall, average *F1* score [91], average cosine similarity score [81], average BLEU score [79] and WER score [92] are obtained as 0.89, 0.95, 0.92, 0.92, 0.90 and 0.21, respectively. In addition, these metrics obtained values for Turkish videos in VOSK speech tests are 0.96, 0.94, 0.95, 0.95, 0.81 and 0.41, respectively.

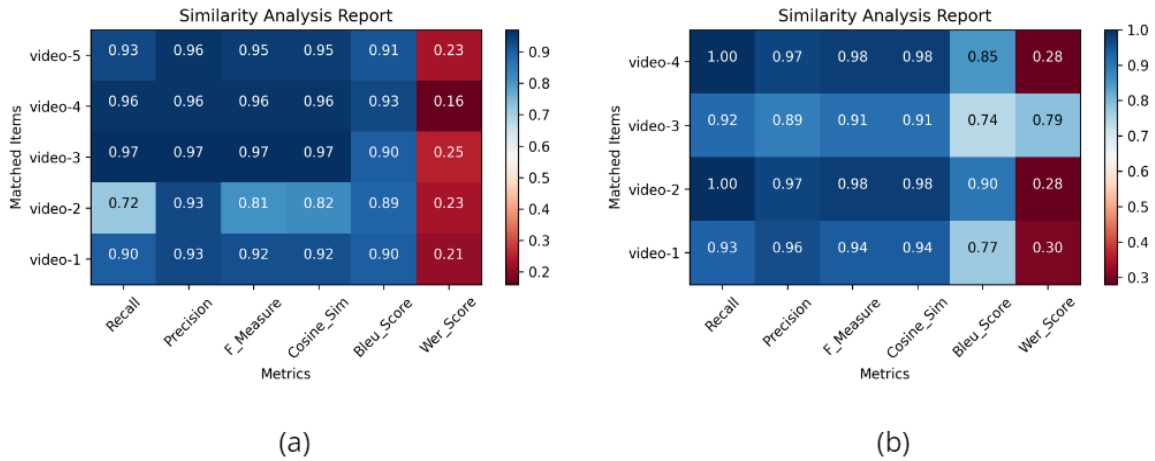


Figure 10 VOSK speech tests applied to (a) the English, and (b) Turkish videos.

At a glance, the cosine similarity score and BLEU score are much higher as they are already expected to show higher performance achievement for similarity analysis between original ground truth transcribed text and predicted text with our model. In addition, WER score is also much lower as it is expected to reflect the low error rate of prediction for accurate sound event class(es) and its correct sentimental conclusion in given videos as well.

The Receiver Operating Characteristics (ROC) Curve is created based on the confusion matrix and the experimental results are given as plot in Figure 11 below. Accordingly, if the curve approaches the upper left corner of the graph, it is understood that the model’s ability to distinguish between classes in classification is quite good. As can be seen from Figure 11, it provides evidence that a very good performance was obtained in the experiments in our study in terms of separation between classes [93].

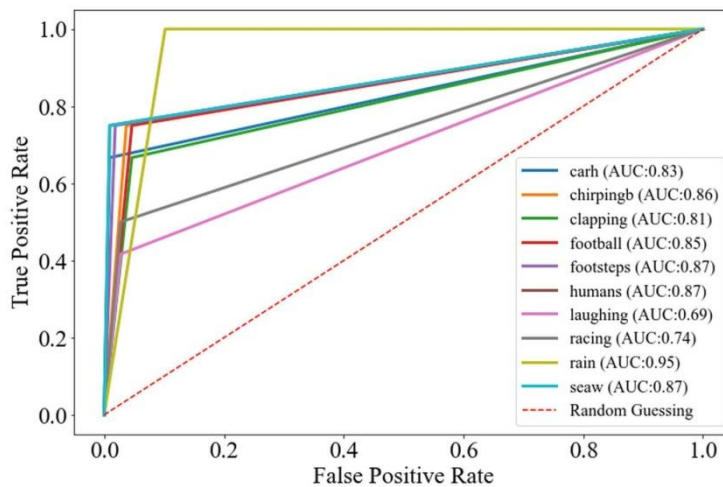


Figure 11 ROC Curve and AUC values for sound recognition process



The Area Under Curve (AUC) value shows the numerical value of the area under the ROC curve, proving how successful this model is in distinguishing between the given sound classes. This value is in the range of 0 to 1, and the closer it is to 1, the higher the performance of the model. These ROC-AUC values can be seen in Figure 11 above. Accordingly, this value proves that our model can make a very successful classification for data of this scale [94]. It is clearly seen from the ROC plot, the “rain” sound class is a more distinguishing class for our model than other recognized sound classes with the ROC-AUC value obtained as 0.95 as well.

In Figure 12, some experimental results taken from an English video are given which involve the prediction results for sound event class in the given video and its sentiment analysis with its ratio. And also in Figure 13, some sound event class prediction is given from a Turkish video without spoken content.

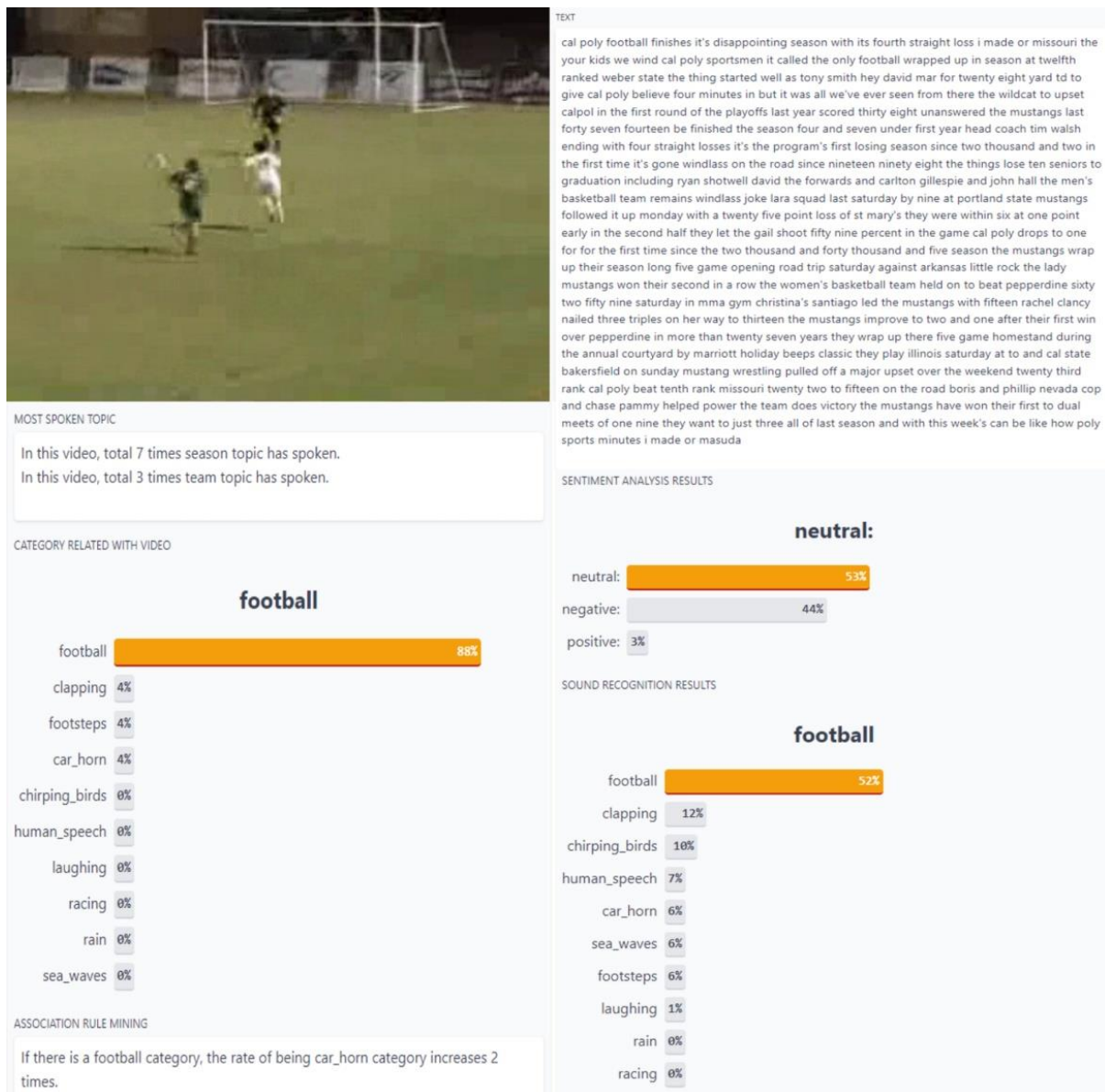


Figure 12 An experimental result on GUI for English video presented with its sentiment analysis, sound event recognition result and the related topic category [47, 89]

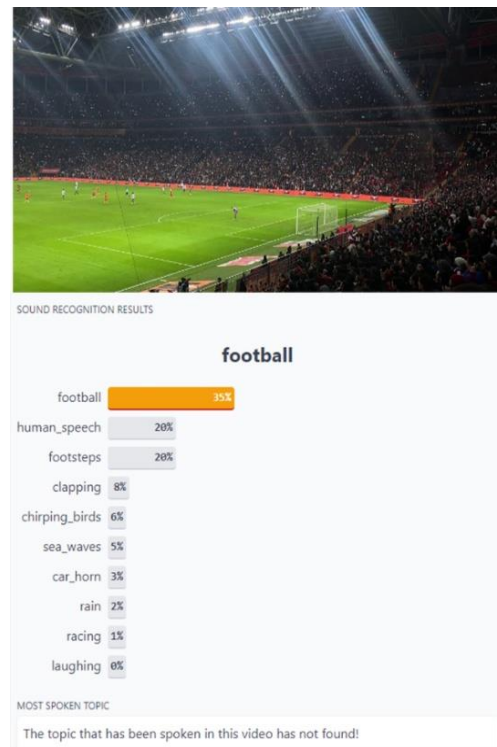


Figure 13 Experimented result for predicted sound event class for Turkish video without spoken content [47, 89]

In this scope, our study achieves consistent and reliable prediction results to prove our main concept about audio-visual content understanding with the help of deep learning.

## 5. Conclusion

In this study, the content understanding methodology for determination of the sentiment from spoken content based on the multimodal information is presented in which it is given in the manner of audio and video processing. This is based on the multilingual audio-visual content extracted using a deep learning approach without any other manual intervention. In addition, the sound scene classification and sound event/action recognition is applied by our approach to reach a refined sentiment analysis from spoken content. Its related sentences are taken from the processed video clip using speech recognition, natural language processing (NLP) [7] and Apriori algorithm [88]. To present our “*Proof of Concept*”, our experiments were conducted to prove the reliability and consistency of our methodology for real world scenarios with the help of deep learning’s high level interpretation of abstract features on raw data. As seen from these experiments’ results, both English and Turkish spoken content with related sentiment were processed and analyzed quite well with high accuracy to reach a conclusion of event/action on a given video clip in the tests. By considering average WER scores in our experiments, lower WER scores indicates its superiority that our audio-visual content understanding approach is more accurate to recognize topic and sentiment, emotional states in videos. In further studies, we plan to add some other deep learning models to our methodology, and thus, we will expand the test set with different scenarios involving other sound classes synthetic or natural sounds in the environment that our approach can be used to concluding the interaction with different event/action pairs for resolving the overlapping issues in complex audio signals in need to cleaning the channels for source separation.

## References

- [1] B. Karakaya, E.B. Boztepe, and B. Karasulu, "Development of a Deep Learning Based Model for Recognizing the Environmental Sounds in Videos," in *The SETSCI Conference Proceedings*

- Book, vol. 5, no. 1, pp. 53-58, 2022.
- [2] B. Karasulu, "Çoklu Ortam Sistemleri İçin Siber Güvenlik Kapsamında Derin Öğrenme Kullanarak Ses Sahne ve Olaylarının Tespiti," *Acta Infologica*, vol. 3, no. 2, pp. 60-82, 2019.
- [3] E. A. Kıvrak, B. Karasulu, C. Sözbir ve A. Türkay, "Ses Özneliklerini Kullanan Ses Duygu Durum Sınıflandırma İçin Derin Öğrenme Tabanlı Bir Yazılımsal Araç," *Veri Bilim Dergisi*, vol. 4, no. 3, pp.14-27, 2021.
- [4] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a Convolutional Neural Network," in *Proceedings of the International Conference on Engineering and Technology (ICET)*, Antalya, Turkey, pp. 1-6, 2018.
- [5] Y. Zhao, X. Jin, and X. Hu, "Recurrent Convolutional Neural Network for Speech Processing," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5300-5304, 2017.
- [6] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng, "Multimodal Deep Learning," in *Proceedings of the 28th International Conference on Machine Learning (ICML11)*, Bellevue, Washington, USA, pp. 689–696, 2011.
- [7] S. Bird, E. Loper, and J. Baldridge, "Multidisciplinary Instruction with the Natural Language Toolkit," in *Proceedings of the Third Workshop on Issues in Teaching Computational Linguistics*, Columbus, Ohio, pp. 62–70, 2008.
- [8] J. Joseph, and J. R. Jeba, "Information Extraction Using Tokenization And Clustering Methods," *International Journal of Recent Technology and Engineering*, vol. 8 no. 4, pp. 3680-3692, 2019.
- [9] H. van Halteren, J. Zavrel, and W. Daelemans, "Improving Accuracy in NLP Through Combination of Machine Learning Systems," *Computational Linguistics*. vol. 27, no. 2, pp. 199–229, 2001.
- [10] A. Roy, "Recent Trends in Named Entity Recognition (NER)," arXiv preprint arXiv:2101.11420 [cs.CL], 2021.
- [11] K. Shaukat, S. Zaheer, and I. Nawaz, "Association Rule Mining: An Application Perspective," *International Journal of Computer Science and Innovation*, vol. 2015, no. 1, pp.29-38, 2015.
- [12] VOSK Offline Speech Recognition Library Website, 2022, [Online]. Available: <https://alphacephei.com/vosk/>. [Accessed: 01-July-2022]
- [13] Ö. Şahinaslan, H. Dalyan ve E. Şahinaslan, "Naive Bayes Sınıflandırıcısı Kullanılarak YouTube Verileri Üzerinden Çok Dilli Duygu Analizi," *Bilişim Teknolojileri Dergisi*, vol. 15, no. 2, pp. 221-229, 2022.
- [14] M.C. Yılmaz ve Z. Orman, "LSTM Derin Öğrenme Yaklaşımı ile Covid-19 Pandemi Sürecinde Twitter Verilerinden Duygu Analizi," *Acta Infologica*, vol. 5, no. 2, pp. 359-372. 2021.
- [15] N. Buduma and N. Lacascio, *Designing Next-Generation Machine Intelligence Algorithms Fundamentals of Deep Learning*, O'Reilly Media UK Ltd., 2017.
- [16] F. Chollet, *Deep Learning with Python*, Manning Publications, 2017.
- [17] Y. Shen, C.-H. Demarty, and N.Q.K. Duong, "Deep Learning for Multimodal-Based Video Interestingness Prediction," in *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)*, pp. 1003-1008, 2017.
- [18] Y.-G. Jiang, Y. Wang, R. Feng, X. Xue, Y. Zheng, and H. Yang, "Understanding and Predicting Interestingness of Videos," in *Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence*, pp. 1113–1119, 2013.
- [19] D. M. Agrawal, H. B. Sailor, M. H. Soni, and H. A. Patil, "Novel TEO-based Gammatone Features for Environmental Sound Classification," in *Proceedings of the 25th European Signal Processing Conference*, pp.1859-1863, 2017.
- [20] Z. Mushtaq and S.-F. Su, "Efficient Classification of Environmental Sounds through Multiple Features Aggregation and Data Enhancement Techniques for Spectrogram Images," *Symmetry*, vol. 12, no. 11:1822, pp. 1-34, 2020.
- [21] DenseNet Documentation, 2022, [Online]. Available: <https://github.com/liuzhuang13/DenseNet>. [Accessed: 01-July-2022].
- [22] A. Khamparia, D. Gupta, N.G. Nguyen, A. Khanna, B. Pandey, and P. Tiwari, "Sound Classification Using Convolutional Neural Network and Tensor Deep Stacking Network," *IEEE Access*, vol. 7, pp. 7717-7727, 2019.

- [23] K.J. Piczak, "Environmental sound classification with convolutional neural networks," in *Proceedings of the IEEE 25th International Workshop on Machine Learning for Signal Processing (MLSP)*, Boston, MA, USA pp. 1-6. 2015.
- [24] R. A. Khalil, E. Jones, M. I. Babar, T. Jan, M. Haseeb Z., and T. Alhussain, "Speech Emotion Recognition Using Deep Learning Techniques: A Review," *IEEE Access*, vol. 7 pp. 117327-117345, 2019.
- [25] M. Gygli, H. Grabner, and L. V. Gool, "Video Summarization By Learning Submodular Mixtures Of Objectives," in *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, USA, pp. 3090-3098, 2015.
- [26] B. A. Plummer, M. Brown, and S. Lazebnik, "Enhancing Video Summarization Via Vision-Language Embedding," in *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, pp. 1052-1060, 2017.
- [27] K. Zhang, W.-L. Chao, F. Sha, and K. Grauman, "Summary Transfer: Exemplar-Based Subset Selection For Video Summarization," in *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 1059-1067, 2016.
- [28] K. Petros, and M. Petros, "SUSiNet: See, Understand and Summarize It," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Long Beach, CA, USA 16-17 June, pp. 809-819, 2019.
- [29] Python Programming Language and Python Modules Documentation, 2022, [Online]. Available: <https://www.python.org/doc/>. [Accessed: 01-July-2022]
- [30] Tensorflow Library Documentation, 2022, [Online]. Available: [https://www.tensorflow.org/api\\_docs](https://www.tensorflow.org/api_docs). [Accessed: 01-July-2022]
- [31] Keras Library Documentation, 2022, [Online]. Available: <https://keras.io/api/>. [Accessed: 01-July-2022]
- [32] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," in *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Salt Lake City, UT, USA, pp. 4510-4520, 2018.
- [33] Python Data Analysis Library (Pandas) Website 2022, [Online]. Available: <https://pandas.pydata.org/>. [Accessed: 01-July-2022].
- [34] Library for Visualization with Python (Matplotlib) Website, 2022, [Online]. Available: <https://matplotlib.org/>. [Accessed: 01-July-2022].
- [35] Python Statistical Data Visualization Library (Seaborn) Website, 2022, [Online]. Available: <https://seaborn.pydata.org/introduction.html>. [Accessed: 01-July-2022].
- [36] Numerical Library for Python (NumPy), 2022, [Online]. Available: <https://numpy.org/>. [Accessed: 01-July-2022]
- [37] SpaCy Natural Language Processing Library for Python, 2022, [Online]. Available: <https://spacy.io/api/doc>. [Accessed: 01-July-2022].
- [38] Manipulate Audio Library (PyDub) Website, 2022, [Online]. Available: <https://pydub.com/>. [Accessed: 01-July-2022].
- [39] OpenCV Library Documentation, 2022, [Online]. Available: <https://docs.opencv.org/4.6.0/>. [Accessed: 01-July-2022].
- [40] Moviepy Library Documentation, 2022, [Online]. Available: <https://zulko.github.io/moviepy/>. [Accessed: 01-July-2022].
- [41] B. McFee, C. Raffel, D. Liang, D. Ellis, M. Mcvicar, E. Battenberg, and O. Nieto, "Librosa: Audio and Music Signal Analysis in Python," in *Proceedings of the Python in Science Conference*, 2015.
- [42] E. Loper and S. Bird, "NLTK: the Natural Language Toolkit," in *Proceedings of the ACL-02 Workshop on Effective tools and methodologies for teaching natural language processing and computational linguistics*, vol. 1, pp. 63-70, 2002.
- [43] Transformers Library Documentation, 2022, [Online]. Available: <https://huggingface.co/docs/transformers/main/en/index>. [Accessed: 01-July-2022].
- [44] Diffilib module computing deltas for Python, 2022, [Online]. Available: <https://docs.python.org/3/library/diffilib.html>. [Accessed: 01-July-2022].
- [45] Zeyrek: Morphological Analyzer and Lemmatizer GitHub Website, 2022, [Online], Available:



- <https://github.com/obulat/zeyrek>. [Accessed: 01-July-2022].
- [46] Library for approximate and phonetic matching of strings for Python, 2022, [Online]. Available: <https://github.com/jamesturk/jellyfish>. [Accessed: 01-July-2022].
- [47] Gradio Library Documentation, 2022, [Online]. Available: <https://gradio.app/docs/>. [Accessed: 01-July-2022].
- [48] K. J. Piczak, "ESC: Dataset for Environmental Sound Classification," in *Proceedings of the 23rd ACM international conference on Multimedia*, pp. 1015–1018, 2015.
- [49] V. Panayotov, G. Chen, D. Povey and S. Khudanpur, "Librispeech: An ASR corpus based on public domain audio books," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, South Brisbane, QLD, Australia, pp. 5206 - 5210, 2015.
- [50] T. M. Hospedales, S. Gong, and T. Xiang, "Learning Tags from Unsegmented Videos of Multiple Human Actions," in *Proceedings of the IEEE 11th International Conference on Data Mining*, Vancouver, BC, Canada, pp. 251-259, 2011.
- [51] Youtube. 2022. [Online]. Available: <https://www.youtube.com>. [Accessed: 01-July-2022].
- [52] R. Kolobov et al., "MediaSpeech: Multilanguage ASR Benchmark and Dataset," arXiv preprint arXiv:2103.16193, 2021.
- [53] M. Rochan, L. Ye, and Y. Wang, "Video Summarization Using Fully Convolutional Sequence Networks," in *Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds) Computer Vision – ECCV 2018. ECCV 2018. Lecture Notes in Computer Science*, vol. 11216. pp 358–374, 2018.
- [54] S. Jadon and M. Jasim, "Unsupervised video summarization framework using keyframe extraction and video skimming," in *Proceedings of the IEEE 5th International Conference on Computing Communication and Automation (ICCCA)*, Greater Noida, UP, India, Oct 30-31, pp. 140-145, 2020.
- [55] J. Park, J. Lee, S. Jeon, and K. Sohn, "Video Summarization by Learning Relationships between Action and Scene," in *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, Seoul, Korea (South), 27-28 October, pp. 1545-1552, 2019.
- [56] Z. Li, G. M. Schuster, A. K. Katsaggelos, and B. Gandhi, "Rate-distortion optimal video summarization: a dynamic programming solution," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, Montreal, QC, Canada, vol. 3, pp. iii-457, 2004.
- [57] S. Lu, M. R. Lyu, and I. King, "Video summarization by spatial-temporal graph optimization," in *Proceedings of the 2004 IEEE International Symposium on Circuits and Systems (ISCAS)*, Vancouver, BC, Canada, pp. II-197, 2004.
- [58] D. Potapov, M. Douze, Z. Harchaoui, and C. Schmid, "Category-specific video summarization", in *Proceedings of the European Conference on Computer Vision (ECCV)*, Zurich, Switzerland, 6-12 September, pp. 540–555, 2014.
- [59] M. Otani, Y. Nakashima, E. Rahtu, and J. Heikkila, "Rethinking the Evaluation of Video Summaries," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA, pp. 7588-7596, 2019.
- [60] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition", arXiv preprint arXiv:1409.1556v6 [cs.CV], 2015.
- [61] K. Zhou, Y. Qiao and T. Xiang, "Deep Reinforcement Learning for Unsupervised Video Summarization with Diversity-Representativeness Reward," arXiv preprint arXiv:1801.00054, 2018.
- [62] Kernel Temporal Segmentation (KTS). 2022. [Online]. Available: <https://github.com/TatsuyaShirakawa/KTS>. [Accessed: 01-July-2022]
- [63] Y. Song, J. Vallmitjana, A. Stent, and A. Jaimes, "TVSum: Summarizing web videos using titles," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, USA, 07-12 June, pp. 5179-5187, 2015.
- [64] R. Andonov, V. Poirriez, and S. Rajopadhye, "Unbounded knapsack problem: Dynamic programming revisited," *European Journal of Operational Research*, vol. 123, no. 2, pp. 394-407, 2000.
- [65] M. Gygli, H. Grabner, H. Riemenschneider, and L. van Goo, "Creating Summaries From User



- Videos,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, Zurich, Switzerland, 6-12 September, pp. 505–520, 2014.
- [66] P. Musa, F. Rafi, and M. Lamsani, “A Review: Contrast-Limited Adaptive Histogram Equalization (CLAHE) Methods to Help the Application of Face Recognition,” in *Proceedings of the Third International Conference on Informatics and Computing (ICIC)*, Palembang, Indonesia, 17-18 October, pp. 1-6, 2018.
- [67] Z. Zhang, S. Xu, S. Zhang, T. Qiao, and S. Cao, “Learning Attentive Representations for Environmental Sound Classification,” *IEEE Access*, vol. 7, pp. 130327 - 130339, 2019.
- [68] Ö. Eskidere ve F. Ertaş, “Mel Frekansı Kepstrum Katsayılarındaki Değişimlerin Konuşmacı Tanımaya Etkisi,” *Uludağ Üniversitesi Mühendislik-Mimarlık Fakültesi Dergisi*, vol. 14, no. 2, pp. 93-110, 2009.
- [69] Md. A. Hossan, S. Memon, and M. A. Gregory, “A Novel Approach for MFCC Feature Extraction,” in *Proceedings of the 4th International Conference on Signal Processing and Communication Systems (ICSPCS)*, Gold Coast, QLD, Australia, 13-15 December, pp. 1-5, 2010.
- [70] N. Jiang, P. Grosche, V. Konz, and M. Müller, "Analyzing chroma feature types for automated chord recognition", in *Proceedings of the 42nd AES International Conference on Semantic Audio*. Ilmenau, Germany, pp. 285-294, 22-24 July, 2011.
- [71] Rotating Images Information Website, 2022, [Online]. Available: <https://datagenetics.com/blog/august32013/index.html>. [Accessed: 01-July-2022].
- [72] Y. Bengio, A. Courville, and Pa. Vincent, “Representation Learning: A Review and New Perspectives,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 8, pp. 1798-1828, 2013.
- [73] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “ImageNet: A Large-Scale Hierarchical Image Database,” in *Proceedings of 2009 IEEE Conference on Computer Vision and Pattern Recognition*, Miami, FL, USA, pp. 248-255, 20-25 June, 2009.
- [74] D. P. Kingma, and J. Ba, “Adam: A Method for Stochastic Optimization,” in *Proceedings of the 3rd International Conference for Learning Representations*, San Diego, USA, pp. 1-13, 2015.
- [75] S. Albelwi and A. Mahmood, “A framework for designing the architectures of deep convolutional neural networks,” *Entropy*, vol. 19, no. 6:242, 2017.
- [76] M. Folk, G. Heber, Q. Koziol, E. Pourmal, and D. Robinson, “An Overview of the HDF5 Technology Suite and its Applications,” in *Proceedings of the EDBT/ICDT 2011 Workshop on Array Databases*, Uppsala, Sweden, March 25, pp. 36-47, 2011.
- [77] M. Mednis and M. K. Aurich, “Application of String Similarity Ratio and Edit Distance in Automatic Metabolite Reconciliation Comparing Reconstructions and Models,” *Biosystems and Information Technology*, vol.1, no.1, pp. 14-18, 2012.
- [78] K. Dreßler and A.-C. Ngonga Ngomo, “On the Efficient Execution of Bounded Jaro-Winkler Distances,” *Semantic Web, Issue title: Ontology and linked data matching*, vol. 8, no 2, pp 185–196, 2017.
- [79] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, “BLEU: a Method for Automatic Evaluation of Machine Translation,” in *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, Philadelphia Pennsylvania, USA, 7 - 12 July, pp. 311–318, 2002.
- [80] C. Callison-Burch, M. Osborne, and P. Koehn, “Re-evaluating the Role of BLEU in Machine Translation Research,” in *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, Trento, Italy, 3-7 April, pp. 249-256, 2006.
- [81] F. Rahutomo, T. Kitasuka, and M. Aritsugi, “Semantic Cosine Similarity,” in *Proceedings of the 7th International Student Conference on Advanced Science and Technology ICAST*, Seoul, South Korea, 2012.
- [82] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," *Computation and Language (cs.CL)*, arXiv preprint arXiv:1810.04805 [cs.CL], 2018.
- [83] Hugging Face Services Documentation, 2022, [Online]. Available: <https://huggingface.co/docs>. [Accessed: 01-July-2022].
- [84] Roberta Sentiment Model Documentation, 2022, [Online]. Available:

- <https://huggingface.co/cardiffnlp/twitter-roberta-base-sentiment>. [Accessed: 01-July-2022].
- [85] BERT-Turkish Sentiment Model Documentation, 2022, [Online]. Available: <https://huggingface.co/savasy/bert-base-turkish-sentiment-cased>. [Accessed: 01-July-2022].
- [86] S. Yildirim, "Comparing Deep Neural Networks to Traditional Models for Sentiment Analysis in Turkish Language," In: B. Agarwal, R. Nayak, N. Mittal, and S. Patnaik, (eds) *Deep Learning-Based Approaches for Sentiment Analysis. Algorithms for Intelligent Systems*. Springer, Singapore, pp. 311-319, 2020.
- [87] S. Sarica and J. Luo, "Stopwords in Technical Language Processing," *Plos One*, vol.16, no.8, pp. 1-13, 2021.
- [88] S. Panjaitan, Sulindawaty, M. Amin, S. Lindawati, R. Watrionthos, H. T. Sihotang, and B. Sinaga, "Implementation of Apriori Algorithm for Analysis of Consumer Purchase Patterns," in *Proceedings of the International Conference on Computer Science and Applied Mathematic, IOP Conf. Series: Journal of Physics: Conf. Series*, vol. 1255, no. 1, pp. 1-8, 2019.
- [89] AVESA GitHub Repository, 2022, [Online]. Available: <https://github.com/berayboztepe/AVESA>. [Accessed: 01-July-2022].
- [90] Pexels Website, 2022, [Online]. Available: <https://www.pexels.com>. [Accessed: 01-July-2022].
- [91] B. Karasulu, "Kısıtlanmış Boltzmann makinesi ve farklı sınıflandırıcılarla oluşturulan sınıflandırma iş hatlarının başarımının değerlendirilmesi", *Bilişim Teknolojileri Dergisi*, vol. 11, no. 3, pp. 223-233, 2018.
- [92] A. Ali and S. Renals, "Word Error Rate Estimation for Speech Recognition: e-WER," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, Melbourne, Australia, 15 - 20 July, pp. 20-24, 2018.
- [93] T. Fawcett, "Introduction to ROC analysis," *Pattern Recognition Letters*, vol. 27, no. 8, pp. 861-874, 2006.
- [94] D. M. W. Powers, "The Problem of Area Under the Curve," in *Proceedings of the IEEE International Conference on Information Science and Technology (ICIST)*, Wuhan, China, 23-25 March, pp. 567-573, 2012.