A Lightweight Multimodal Learning Model to Recognize User Sentiment in Mobile Devices

Jyotirmoy Karjee, Srinidhi N, Gargi Dwivedi, Arun Bhagavath, Prajwal Ranjan
Samsung R&D Institute India, Bangalore
Email: {j.karjee, srinidhi.n, gargi.d, arun.b01, prajwal.r}@samsung.com

2023 IEEE International Conference on Consumer Electronics (ICCE) | 978-1-6654-9130-3/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICCE56470.2023.10043524

***Abstract*—Communications through video/audio calling and text messaging has seen a meteoric rise over the last few years. As we progress towards the advanced wireless technology (i.e., 5G), the quality of call and network low latency communication provides better quality of services (QoS). This allows for addition of new features that would provide users with a more personalized experience such as providing sentiments analysis of users through mobile (sender/receiver) devices over text messaging or video calls or both. However to perform sentiment analysis requires Deep Neural Networks (DNN) model to extract multimodal data (such as text, audio and video) features which is very heavy to deploy in mobile devices due to its limited computational capabilities. In the past, various research works has been performed to develop multimodal feature extraction mechanisms, however none of the multimodal models are suited to be deployed in mobile devices for practical applications. To mitigate these issues, we propose a light weight multimodal learning model called Tri-Feature Fusion which can be easily deployable in mobile devices for client-server communications. The Tri-Feature Fusion model extracts the feature vectors of each modality and generate a single multimodal feature vector from the data generated from the call and perform sentiment analysis on each particular mode. We also have developed a light weight neural network for Tri-Feature Fusion to perform the sentiment analysis using the extracted features for real-time audio, video and text data displayed at clients and receiver mobile devices. We conduct extensive experiments to showcase the performance of Tri-Feature Fusion as light weight model in-terms of average time taken for prediction, time taken per step, CPU utilization and size of the models compared with the existing state of art.**

I. Introduction

The implementation of advanced wireless technologies (i.e., 5G) [1] has drastically eased in-terms of efficient way of computations and communications in mobile devices. With the transition into the 5G, video calling applications in mobile devices can be improved immensely with users sentiment [2] being provided with more features to further improve their experience. Millions of mobile users around the world can easily communicate with each other over text-based, audio-based or video-based communications.

Presently, many features are being provided to mobile users to improve their experience when using video calling applications. One feature that can be provided is the real-time user sentiments using multimodal communications [3], [4]. The sentiment can be displayed during a video-call or over text-messaging applications. However, such feature requires the implementation of a sophisticated sentiment analysis algorithms on the smart mobile devices in real-time for multimodal communications for synchronization of live audio, text and video applications to reduce the latency, which lags the present state of art in smart mobile applications.

Sentiment analysis is the process of identifying user's inclination towards a topic using Natural Language Processing (NLP). Using sentiment analysis, one can infer whether a user has a positive inclination, negative inclination or a neutral inclination towards a topic that they are discussing about. A sentiment analysis uses Deep Neural Networks (DNN) models; such as Recurrent Neural Networks (RNN), Convolutional Neural Network (CNN) and Long Short Term Memory (LSTM) networks, etc., that requires the audio, video and text features as inputs for further processing for identifying user sentiments. However, these DNN models are very heavy in-terms of time and computational latency to be deployed in smart phone devices due to its limited CPU/GPU capabilities.

Further, to perform sentiment analysis in mobile devices, the information is obtained from more than one input simultaneously called multimodal features. However, training (or testing) such multimodal features in mobile devices is very expensive in-terms of computational power as the features extraction for multimodal data (i.e, text, audio and video) uses multi dimensional data vectors (as an input) before fusion and feeding to DNN models. One such feature fusion strategies, is proposed recently called Hfusion [4] model.

The Hfusion model provides hierarchical fusion strategies to fuse two in two and only then fusing all the three modalities (i.e., text, audio and video). However, due to multi dimensional features extraction processed for each individual utterances processed at many steps (i.e, dimensionality equalization, 2-Modal Fusion and 3-Modal fusion and the feeding to DNN model) makes the model complex to be deployed in mobile devices. Further in Hfusion model, the extracted multimodal utterances are feed to RNN which can increase the computational inference time as RNN is a complex DNN model which consists of many dense layers.

Hence, needs a requirement to develop a tool that provides the multi-modality operations in real-time for user sentiment using a neural network which is lightweight and computationally inexpensive. Development of such a tool can allow its implementation onto devices such as mobiles or smartwatches despite their lower computation capacities. To tackle the above challenges, we develop a client-server multimodal mobile system called Tri-Feature Fusion model that provides real-

978-1-6654-9130-3/23/$31.00 ©2023 IEEE

time communications for sentiment analysis for a video call applications among two mobile users. The contributions of our proposed model are listed below.

- The Tri-Feature Fusion model performs feature extraction of each multimodal data (i.e., video, audio and text) as an input. Since each feature would have its own data, we perform feature extraction for each individual feature to obtain their respective vector as input to the processing network. Thus, the Tri-Feature Fusion, extracts individual feature vector for every modality and then concatenates them into a single vector to produce the multimodal feature vector. Since all the modalities are within once vector in Tri-Feature Fusion its provides significant improvement in-terms of computation and data storage for training and testing in mobile devices.
- In the Tri-Feature Fusion model as all the feature vectors/modalities extraction are passed as a single dimension as input to the neural network of the model, the processing speeds (taken for each step run-time for feature extraction and CPU utilization) are reduced significantly in the model. To perform this operation, we have used TF-IDF (Term Frequency Inverse Document Frequency), OpenSMILE and OpenCV & dlib for text, audio and video feature extractions, respectively in the model. Thus our approach is novel in-terms of light weight model compared to existing model implemented in mobile devices.
- In the Tri-Feature Fusion model, the neural network consists of a input layer (which extracts the multimodal feature vector obtained by concatenation of vector of each modality), three fully connected dense neural networks with different kernel size and bias followed by two Rectified Linear Unit (ReLU) and Softmax activation function. Further, the output of the Softmax is passed to a dense-3 network to provide the final output in-terms of three modalities for sentiment analysis. The proposed neural network in the Tri-Feature Fusion model is a light weight model to be implemented in mobile devices (such as smart phone) providing unique as mobile devices cannot run complex deep learning model such as RNN.
- Finally, using the Tri-Feature Fusion model, we have built a client-server communication system where two users communicate with each other over text-messaging or a video call. This exchange would generate multimodal input data to provide sentiment generated from the neural network is then displayed on the user's screen along with their client's real-time audio, video and text data. We conduct extensive experiments and analysis to provide the performance of the Tri-Feature Fusion model in-terms of prediction time, computation time per step, CPU utilization and neural network model size to validate our results compared to existing model.

To the best of authors knowledge, this is the first work which facilitates a light weight multimodal learning model called Tri-Feature Fusion which is suitable for deploying in mobile devices. The rest of the paper consists of the following sections. Section II discusses the related works. Section III describes the Tri-Feature Fusion system model in details. The Section IV discusses the simulation and experimental results. Finally, in Section V, we conclude our work.

II. Related Work

Over the past few years in 5G [1] technology, sentiment analysis [2], [3] has been a growing field for multimodal learning to perform audio, video and text features extraction. In-terms of open source tools, such as OpenMM [5] have been presented for multimodal feature extraction. In [6], the authors proposed Select-Additive Learning (SAL) procedure for multimodal sentiment analysis. Their suggested method improves the ability to generalize pre-trained models that are used for multimodal sentiment analysis while improving prediction accuracy of all three modalities along with their fusion. In [7], the authors proposed a method for predicting sentiments in web videos. Their proposed method incorporated feature-level and decision-level fusion methods to combine relevant information from different modalities (i.e., audio, visual and textual). In [8], the authors built an open-source framework for extensive multimodal feature extraction which supports annotation of diverse that uses audio, text, image and video modalities. The authors in [9], provided an efficient approach for different feature extractions. Their framework restricts the constraints caused by whitening, while maintaining the same feature geometry. In the case of missing modalities, the framework makes semi-supervised adaptations to improve its discrimination abilities.

In [10], the authors proposed a feature extraction model to optimally predict short-term electric load forecasts. They extract multimodal spatial temporal features and perform forecasting on it. In [11], the authors performed multimodal feature extraction on audio and video input. They obtained features to perform voice recognition and identify active speakers present within a gathering. The authors in [12], considered Convolutional Neural Networks (CNN) to obtain visual features, such as lips movement from the ultrasound and video inputs. They obtained features to perform speech recognition using visual inputs only. In [13], the authors performed sentiment analysis of football videos. They proposed improvement of response time and precision along with emphasising the elimination for the requirements of human intervention in video analysis. The authors in [14], proposed a system to evaluate candidate's profile specific to a job profile. They do so by rating features such as verbal contents, judgments and personalities. Although in the above literatures, authors proposed various multimodal models, however none of the multimodal models can provide a unique system, where individual feature modality vectors are captured into a single vector to produce multimodal operation in mobile devices. Further none of the works mentioned about a light-weight multimodal model that can be deployable in mobile devices that can reduce computation and communication cost, respectively.

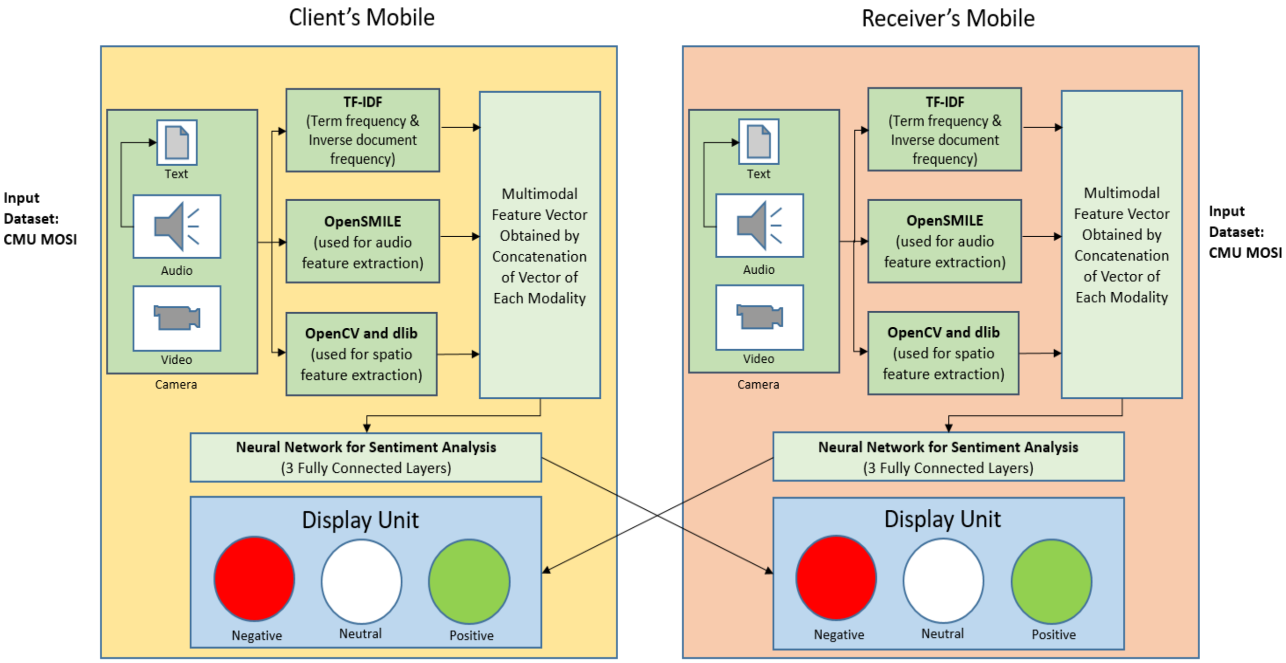
Client's Mobile
Receiver's Mobile
Input Dataset: CMU MOSI
Text
Audio
Video
Camera
TF-IDF (Term frequency & Inverse document frequency)
OpenSMILE (used for audio feature extraction)
OpenCV and dlib (used for spatio feature extraction)
Multimodal Feature Vector Obtained by Concatenation of Vector of Each Modality
Neural Network for Sentiment Analysis (3 Fully Connected Layers)
Display Unit
Negative
Neutral
Positive
Input Dataset: CMU MOSI

Fig. 1. Tri-Feature Fusion model: Multimodal Feature Extraction and Neural Network Architecture

III. System Model

In this section, we explain the proposed Tri-Feature Fusion model as shown in Fig. 1. The proposed model consists of three parts: the multimodal feature extraction, the neural network for sentiment analysis and the client-server system integrated with Tri-Feature Fusion as discussed below.

A. Multimodal Feature Extraction

Since the available input from video calls can be of various types, we perform separate feature extraction for each respective required feature. In this paper, we perform feature extraction for video feature, text feature and audio feature.

1) Video Feature Extraction: In order to extract the video features, we split the incoming video stream (V_{in}) into video frames as $\{v_1, v_2, \ldots, v_n\}$. The software such as *Mediapipe* [15], *OpenCV* and *dlib* are used to perform the feature extraction from the input video frames. The primary area of focus for video feature extraction is the facial expression of the client during the video call and the same information is provided to the server (i.e., the receiver). To work with this, we consider $P = \{\{p_{1,1}, \ldots, p_{1,50}\}, \ldots, \{p_{n,1}, \ldots, p_{n,50}\}\}$ as a set of 50 facial landmarks out of 468 facial landmarks (or points) that are identified by *Mediapipe* for each video frame. Once the points are obtained, we calculate the distance, $D = \{\{d_{1,1,1}, \ldots, d_{50,50,1}\}, \ldots, \{d_{1,1,n}, \ldots, d_{50,50,n}\}\}$ between between every pair of facial points for every video frame. Further, we find the average of this distance, $D_{avg} = \{\mu_{1,1}, \ldots, \mu_{50,50}\}$ over facial point for each frame of the video clip. The average distance $\mu_{i,j}$ is computed as,

$$\mu_{i,j} = \frac{\sum_{k=0}^{k=n} d_{i,j,k}}{n} \tag{1}$$

where n denotes the number of frames, and $d_{i,j,k}$ denotes the distance between the i^{th} and j^{th} facial point, $i \leq j$ and $i, j \in \{1, \ldots, 50\}$, in the k^{th} frame. Similarly, the standard deviation of the distance is also calculated by performing,

$$\sigma_{i,j} = \sqrt{\frac{1}{n} \sum_{k=0}^{k=n} (d_{i,j,k} - \mu_{i,j})^2} \tag{2}$$

for $i, j \in \{1, \ldots, 50\}$ and $i \leq j$. It is then concatenated along with the mean to generate the video feature vector (V).

2) Audio Feature Extraction: The input audio stream (A_{in}) contains the client speech extracted using *OpenSMILE*. Hence, we perform speech recognition techniques on the audio stream to get text from the audio clip. To perform this, we have used *pocketsphinx* [16] which is an offline tool in Python's speech recognition library for audio to text conversion. Out of the many possible features in the audio clip, our extraction mechanism focuses on the following five values, (i) Mel-Frequency Cepstral Coefficients (MFCC) denoted as $mfcc$ provides cepstral representation of audio clip, (ii) Chroma Features (cf) captures pitch class profiles of audio (iii) Mel Spectrogram (ms) where audio frequencies are converted to mel scale, (iv) Spectral Contrast (sc), is the measure of frequency energy at each time frame and, (v) Tonal Centroid Features (tonnetz) denoted as tcf captures central tones of audio. These five features are extracted from an audio clip of small duration which is taken from the audio stream that comes from the client. The extraction is done using the *Librosa* library. Using each of the five extractions, a audio feature vector for each feature is obtained. These individual feature vectors are then combined to generate the audio feature vector denoted as $A = \{mf, c, ms, s, t\}$.

3) Text Feature Extraction: The Term Frequency Inverse Document Frequency *(TF-IDF)* is used to calculate the relevance of a word in a series or corpus to a text (T_{in}). The relevance of a word increases proportionally with the number of times it appears in a text, but gets counteract by the frequency of the word in the corpus. The text feature extraction can be divided into two parts, training and actual feature extraction. First, a large text dataset is used to create a usable vocabulary and then the TF-IDF of the text whose features are to be extracted. This gives us the text feature vector as $T = \{tf\}$. This TF-IDF is performed to get the features from the text generated from the audio file to construct the text feature vector.

The Tri-Feature Fusion model is summarized in Algorithm 1. The algorithm consists of one-dimensional feature vector (M) array which concatenate the video feature vector (V), the audio feature vector (A), and the text feature vector (T). The array can be a *NumPy* array. Using the *hstack()* function in NumPy, the three features vectors are combined to get a one-dimensional feature vector representing the joint multimodal data. Mathematically, we can represent tri-feature vector as,

$$M = V \cup A \cup T \tag{3}$$

This feature vector is then passed to the neural network to determine the sentiment of the user. The neural network architecture is described in the following sub-section.

B. Neural Network for Sentiment Analysis

Once the multimodal feature vector is generated from the input stream, it is passed onto a pre-trained three-layer fully-connected neural network to predict the client's sentiment. The sentiment have three classes; namely positive, negative and neutral as an output from neural networks. For example, positive sentiment means video feature user smile on their face or user giggling/laughing proving the respective audio features or some words in positive context thereby providing with the required text feature. In Fig. 2, a neural network is demonstrated with input dimensions, hidden layers, activation functions and output dimensions. A single dimensional vector of 2843 input features are feed to three dense (fully connected) networks with 900, 200 and 60 neurons each along with the *ReLU* activation function. Every dense layer is followed by a dropout layer with a dropout rate of 0.2. The output layer consists of three neurons with each neuron representing a sentiment class- positive, negative and neutral. It also consists of a *Softmax* activation function. Therefore in Tri-Feature Fusion model, the number of layers has been significantly reduced for training the multimodal concatenate data thereby making the model light-weight as compared to traditional deep learning models (such as RNN and LSTM, etc.) suitable for deploying in mobile devices.

Algorithm 1 Tri-Feature Fusion Model

```
Input: $V_{in}$: Video Stream Input, $A_{in}$: Audio Stream Input
Video Feature Extraction:
    $\Rightarrow \{v_1, v_2, \ldots, v_n \rightarrow$ video frames obtained from input video stream where $n$ denotes the number of frames.
    $\Rightarrow P = \{\{p_{1,1}, \ldots, p_{1,50}\}, \ldots, \{p_{n,1}, \ldots, p_{n,50}\}\} \rightarrow$ sets of 50 facial landmark points identified by MediaPipe for each video frame.
    $\Rightarrow D = \{\{d_{1,1,1}, \ldots, d_{50,50,1}\}, \ldots, \{d_{1,1,n}, \ldots, d_{50,50,n}\}\}$ pairwise distance between each facial point for every video frame.
    $\Rightarrow D_{avg} = \{\mu_{1,1}, \ldots, \mu_{50,50}\} \rightarrow$ average of distances across each frame, where,
        $\mu_{i,j} = \frac{\sum_{k=0}^{k=n} d_{i,j,k}}{n}$
    for $i, j \in \{1, \ldots, 50\}$ and $i \leq j$.
    $\Rightarrow S = \{\sigma_{1,1}, \ldots, \sigma_{50,50}\} \rightarrow$ standard deviation of distances across each frame, where,
        $\sigma_{i,j} = \sqrt{\frac{1}{n}\sum_{k=0}^{k=n}(d_{i,j,k} - \mu_{i,j})^2}$
    for $i, j \in \{1, \ldots, 50\}$ and $i \leq j$.
    Video vector$\Rightarrow V = \{\mu_{1,1}, \ldots, \mu_{50,50}, \sigma_{1,1}, \ldots, \sigma_{50,50}\}$
Audio Feature Extraction:
    $\Rightarrow mfcc \rightarrow$ mel-frequency cepstral coefficient, $cf \rightarrow$ chroma feature, $ms \rightarrow$ mel-spectrogram, $sc \rightarrow$ spectral contrast and $tc \rightarrow$ tonal centroid.
    Audio feature vector $\Rightarrow A = \{mfcc, cf, ms, sc, tc\}$
Text Feature Extraction:
    $\Rightarrow T_{in} \rightarrow$ text snippet obtained from audio stream input after using pocketsphinx
    $\Rightarrow tf \rightarrow$ TF-IDF of text snippet
    Text feature vector $\Rightarrow T = \{tf\}$
Concatenation:
    $\Rightarrow M \rightarrow$ tri-feature vector
    $\Rightarrow M = V \cup A \cup T$
    $\Rightarrow M = \{\mu_{1,1}, \ldots, \sigma_{1,1}, \ldots, mfcc, cf, ms, sc, tc, tf\}$
```

C. Client-Server Communication System

We establish a client-server mobile communication architecture using sockets and threading libraries. The separate files are made to handle the client mobile device and receiver mobile device (as a server). Upon execution of the server file, a video and audio socket are generated. These sockets then listen for a client that is looking to connect to the server. Once the connection is established, the client begins to transmit video and audio from their camera and microphone to the server. Multithreading is implemented to allow the server to handle reception of audio and video, simultaneously. This audio and video input is then divided into smaller frames which are then used for feature extraction and sentiment analysis. The

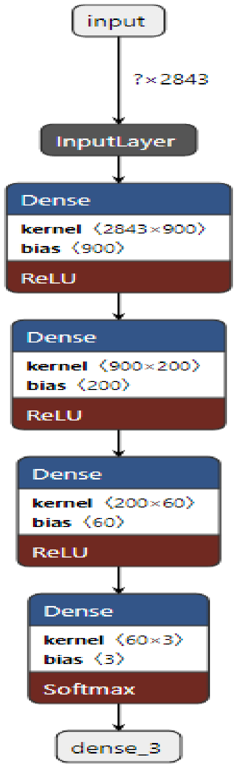

Fig. 2. Neural Network used for Sentiment Analysis

video input is split into hundred frames and passed to a queue. Similarly, the audio input is split into hundred audio frames and passed to the same queue. Sentiment analysis is performed based on the inputs present in the queue. Once generated, this sentiment is displayed in the client's webpage. In order to build the sentiment tool and the video call setup on the webpage, *Flask* [17], which is a lightweight web-framework, is used.

IV. Experimental Results and Validations

To simulate the implementation of sentiment analysis for video calling applications, we have used an input audio and video data through a socket and threading module present in a client file and received it in a server file. We use an Intel Xeon Platinum 8180 2.494 GHz CPU. For the video input data, we use the CMU MOSI dataset [18]. For training the Tri-Feature Fusion model, we consider learning rate=0.001, batch size=10 using Adam optimizer for 500 epochs. The client's sentiment is displayed along with the real time video on a webpage developed using Flask. We compare the results of our proposed feature extraction model, Tri-Feature Fusion with those produced by Hfusion [4]. The parameters considered for evaluating the results of our system are as follows: average time taken for prediction, time taken by the models per step, CPU utilization and size of the model. The comparison for these parameters between Tri-Feature Fusion and HFusion models are collated in Table I.

TABLE I
Comparison of Key Parameters

Parameter	Tri-Feature Fusion	HFusion
Average time taken for prediction	0.2s	1.6132s
Time taken per step	2 ms/step	515 ms/step
CPU Utilization	53.5%	75.1%
Size of the model	31.53 MB	152.2 MB

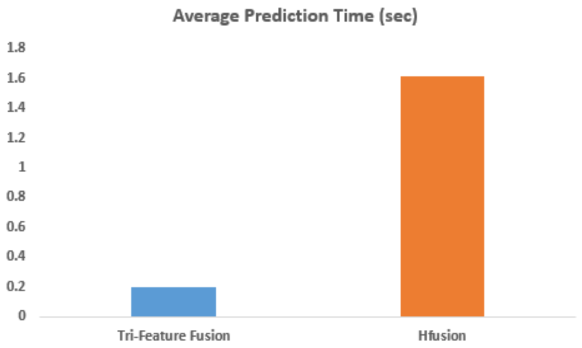

Fig. 3. Average Prediction Time for Tri-Feature Fusion and HFusion

Average Prediction Time: The Tri-Feature Fusion model takes an average prediction (or computational) time of just 0.2 seconds, whereas the Hfusion model takes an average prediction time of 1.6132 seconds as shown in Fig. 3. The avearge prediction time is defined as the time taken to validate the model w.r.t to training the model. Hence, Tri-Feature Fusion model is 12.4% faster than Hfusion in-terms of prediction time complexity. The average prediction time is reduced significantly because Tri-Feature Fusion extracts individual feature vector for every modality and then concatenates them into a single vector. Further, the neural inference layers are reduced in Tri-Feature Fusion making the prediction time fast compared to Hfusion which uses RNN.

Time Taken per Step: Since the goal of our proposed Tri-Feature Fusion model is to display the real-time multimodal data instantly (after capturing the video calling applications), it is necessary to have quick implementation of the neural networks and sentiment analysis in the model in lesser time. To achieve this requirement, the Tri-Feature Fusion is designed to be light-weight model by reducing the number of neural inference layers. The Fig. 4, depicts that the Tri-Feature Fusion takes approximately 2 ms of the time taken per step, whereas HFusion takes an aggregate time of 515 ms, per step while identifying the sentiment of the user.

CPU Utilization: Running neural networks, generally needs lot of computing abilities. Complex architectures such as RNNs requires heavy usages. Since, the Tri-Feature Fusion does not employ the use of RNNs, it uses only 53.5% of the CPU utilization. As shown in Fig. 5, this is about 40.4% times faster than HFusion, where the model consumes up to 75.1% of the CPU utilization. In Tri-Feature Fusion, all the modalities are within one dimensional vector that reduces training time; thereby reducing the CPU utilizations. On the other hand,

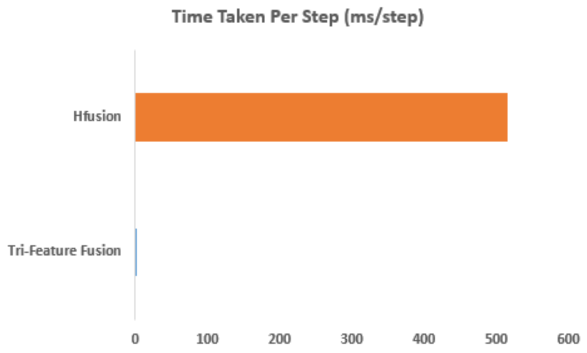

Fig. 4. Time Taken per Step by Tri-Feature Fusion and HFusion

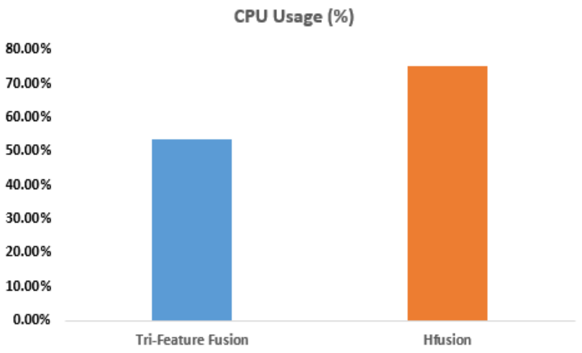

Fig. 5. CPU Usage by Tri-Feature Fusion and HFusion

HFusion uses RNN for which the training time is more due to its deep hidden layer networks (i.e., 2-Model Fusion and 3-Model Fusion with DNN model).

Model Size: One of the crucial aspects in mobile devices (such as in smart phones) is to build lightweight applications. However, it can have limited computation capabilities and takes lower amount of storage space. To full-fill the requirements, we develop the Tri-Feature Fusion model that provides multimodal learning for sentiment analysis. The neural networks used in the Tri-Feature Fusion takes only 31.53 MB of inference layers (size), whereas the Hfusion takes 152.2 MB. This depicts that the Tri-Feature Fusion acquires 4.8 times lesser storage than the Hfusion as depicted in Fig. 6. Thus, the Tri-Feature Fusion provides significant improvement over the Hfusion for model size, thereby reducing the computation complexity.

V. Conclusion and Future Works

In this paper, we have proposed a lightweight sentiment analysis model called Tri-Feature Fusion for video calling on mobile devices (for client-server applications). The proposed model shows significant improvement over Hfusion model; in-terms of prediction time, time taken per step, CPU usage and model size. As computation capabilities of the mobile device tends to be lower, the proposed Tri-Feature Fusion model can be deployed on such low computing device. Since such lightweight neural networks lack the accuracy when compared with complex RNN/LSTM methods, research works can be performed in the future in studying to improve the model accuracy, while maintaining its light-weight characteristics. Finally, we are planning to deploy the Tri-Feature Fusion model in mobile device for commercialization purposes.

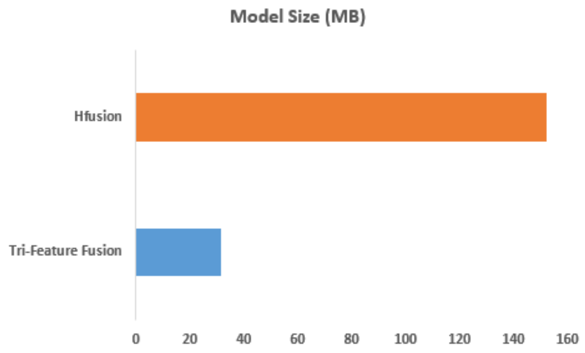

Fig. 6. Comparison of Model Size: Tri-Feature Fusion versus HFusion

References

[1] J. Karjee, K. Anand, V.N. Bhargav, P. Naik, R. Dabbiru and N. Srinidhi, “Split Computing: Dynamic Partitioning and Reliable Communications in IoT-Edge for 6G Vision”, *8th Int. Conference on FiCloud*, 2021.

[2] T. Seçkin, Z. H. Kilimci, “The Evaluation of 5G technology from Sentiment Analysis Perspective in Twitter”, *IEEE Xplore*, 2020.

[3] Y. Zhang, H. Lu, C. Jiang, X. Li, X. Tian, “Aspect-Based Sentiment Analysis of User Reviews in 5G Networks”, *IEEE Network*, vol. 35, no. 4, pp. 228–233, 2021.

[4] N. Majumder, D. Hazarika, A. Gelbukh, E. Cambria, S. Poria, “Multimodal Sentiment Analysis using Hierarchical Fusion with Context Modeling”, *arXiv:1806.06228*, 2018.

[5] M. R. Morales, S. Scherer, R. Levitan, “OpenMM: An Open-Source Multimodal Feature Extraction Tool”, *Interspeech 2017*, 2017.

[6] H. Wang, A. Meghawat, L.P. Morency, E. P. Xing, “Select-Additive Learning: Improving Generalization in Multimodal Sentiment Analysis”, *arxiv.org*, 2016.

[7] S. Poria, E. Cambria, N. Howard, G.B. Huang, A. Hussain, “Fusing audio, visual and textual clues for sentiment analysis from multimodal content”, *Neurocomputing*, vol. 174, pp. 50–59, 2016.

[8] Q. McNamara, A. De La Vega, T. Yarkoni, “Developing a Comprehensive Framework for Multimodal Feature Extraction”, *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2017.

[9] L. Wang et al., “An Efficient Approach to Informative Feature Extraction from Multimodal Data”, *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 5281–5288, 2019.

[10] Z. Kong, C. Zhang, H. Lv, F. Xiong, Z. Fu, “Multimodal Feature Extraction and Fusiong Deep Neural Networks for Short-Term Load Forecasting”, *IEEE Access*, vol. 8, pp. 185373-185383, 2020.

[11] M. Gurban, “Multimodal feature extraction and fusion for audio-visual speech recognition”. *Lausanne: EPFL*, 2009.

[12] E. Tatulli, T. Hueber, “Feature extraction using multimodal convolutional neural networks for visual speech recognition,” *IEEE Xplore*, 2017.

[13] P. Oskouie, S. Alipour, A.M. Eftekhari-Moghadam, “Multimodal feature extraction and fusion for semantic mining of soccer video: a survey”, *Artificial Intelligence Review*, vol. 42, no. 2, pp. 173–210, 2012.

[14] L. Chen et all., “Automated scoring of interview videos using Doc2Vec multimodal feature extraction paradigm”, *Proceedings of the 18th ACM International Conf. on Multimodal Interaction*, 2016.

[15] MediaPipe [Online], Available: *https://mediapipe.dev*

[16] Pocketsphinx [Online]. Available: *https://pypi.org/project/pocketsphinx*

[17] Flask [Online], Available: *https://pypi.org/project/Flask*

[18] CMU MOSI [Online],Available: *http://multicomp.cs.cmu.edu/resources*