AR Comic Chat: A Work in Progress AR caption placement system

Dylan Bowald
Rochester Institute of Technology

Live speech transcription and captioning are important for the accessibility of deaf and hard of hearing folks, especially in situations with no visible ASL translators. If captioning is available at all, it is typically rendered in the form of "closed captions" on large video screens, away from the real event with the captions across the bottom of the screen, potentially dividing the focus of the viewer and detracting from the experience. In this frameless demo, I present my work so far on an alternative, Augmented Reality based approach to displaying these captions, using deep neural networks with video input to identify and localize the speaker, followed by placing a "chat bubble" with the given caption above the head of said speaker.

As far as the author is aware, the specific approach to live captioning outlined in this paper is novel, however some work has been done in the area of alternative live captioning in general. In [3], Koya et. al. outline an approach to live captioning by having participants standing in front of a screen. In [4] Jain et. al. explore AR captioning through the use of manipulable caption windows that the user places.

Finally, there has been some research done in the area of captions placed using facial recognition, such as in DEEP-HEAR[5]. This works well for use cases such as TV shows, where pre-existing labeled datasets for the faces of the speakers are generally well known and available. This is generally a less feasible approach for live environments, which is why a more general, tracking based approach is used in this work.

II. TECHNICAL APPROACH

An overview of the work done so far, along with future work, is as follows.

A. System Description

To reduce uncertainty in the capability of the AR hardware being used, server offloading is utilized. The AR device streams image and audio data to a server, which does the image/audio processing and returns instructions on where the bubbles are to be placed.

For the visual interface/chat bubble graphics, Unity, C#, and Google’s ARCore were used. Inference is done on the server utilizing Python, more specifically PyTorch and OpenCV.

B. Image Processing

The image processing pipeline starts with person tracking, accomplished through fine tuning the DeepSORT[6] object tracker on people using the MOT16 dataset[7]. This gives us 256-dimensional visual embeddings for each person in the scene, allowing for later visual identification without needing prior visual information of the speaker.
Once each person in the scene is being tracked, we run each person region through a 3D Convolutional Neural Network trained on a homemade “speaking/not speaking” dataset to classify the probability that each specific person is speaking. We then assign the given caption to the most likely speaker, and place the chat bubble to their right.

C. Audio Processing

To determine the text of a given speech caption, Google’s SpeechToText API is used. In future work, more audio processing is planned to “track” the voices. The goal would be to determine 256 dimensional embeddings for each speaker, and then associate audio embeddings to visual embeddings by using the 3D convolution that determines when someone is speaking.

III. CONCLUSION

While this is still a work in progress, results so far have been promising. The visual tracking can be done in real time with reasonable accuracy. This upcoming spring, I plan to make this work my master’s thesis, finishing the work on audio processing and doing more in depth comparisons on state of the art captioning techniques.

IV. ACKNOWLEDGEMENTS

The Author would like to thank Andy Head from the NTID Department of Performing Arts along with Benjamin Dow and Stephen Hoffman from RIT Players for allowing the recordings of their productions, used as data in the “speaking/not speaking” classifier.

V. REFERENCES


