Audio Captcha Decoder Using Whisper

The Dataset for the audio captcha solver contains two folders, one containing the audio files in .wav format and one containing the image files in .png format consisting of the correct corresponding texts for the given audio.

For the preprocessing step, I extracted the texts from the given image files into an extractedt_texts.csv file using easyocr and manually corrected some misinterpreted texts on the csv file. This is because easyocr had extracted white spaces from the images which would cause an error while comparing, as the dataset is too large only a few were dealt with manually.

Then, the extracted texts were mapped to their corresponding audio file name into another audio_to_text_mapping.csv file to compare with the ground truths.

To decode the texts from the audio files I have used whisper. Whisper preprocesses audio files as part of the transcription pipeline to make them suitable for input into the model. This preprocessing step involves several tasks, including resampling, trimming, and normalizing the audio data to match the input requirements of the Whisper model.

While decoding, the whisper was decoding the entire speech of the audio file, for example, “Small A, Capital B”. Hence, a post-processing step was carried out to correctly interpret the captchas, so that Small A prints “a” only.

To evaluate the model, CER (Character Error Rate) and accuracy for each audio file was calculated. The CER ranges from 0 - 400%, 400% for extremely bad cases where the model detects an entirely different text. A CER of 16.67% showed one misinterpreted letter and so on.

For accuracy, it is either 0 or 100%, 0% even if there is one misinterpreted letter by the model, and 100% if the entire text is correctly transcribed.

These large discrepancies in CER are also a result of incorrectly extracted texts from the image files, hence even if the model predicts the correct letter it shows a high CER and 0% accuracy. Some of the audio files are also very difficult to interpret due to the speech not being very clear, hence a high CER.

Hence, more improvements could be made by leveraging the use of Deep Learning models where the audio files can be preprocessed and fed into the model for training ensuring high accuracies.