DEEPLY VISUAL MICROPHONE

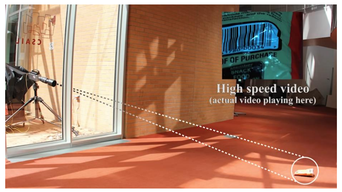
Nischal BK, Arnaav Anand, NC Sathya, Bhanuprakash N
School of Informatics, Computing & Engineering, Indiana University

1. MOTIVATION

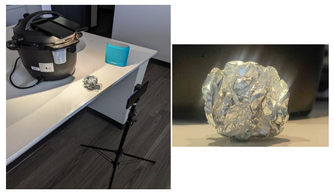
- We found out that it is possible to **extract audio from moving images** and wanted to see if we could replicate the same concept using deep learning.
- Given a slowed-down video split into frames having very **subtle changes in pixel intensities**, can a deep learning model be trained to produce logits corresponding to these changes?
- The generated output waveform should ideally **match the input waveform.**

2. DATA

- Objects that can reflect the motion property need to have high **damping** but also very **light** to **move easily** with changes in **air pressure**. Ex: a bag of **chips**, a **plant**.
- We are using the dataset of a **3 video samples** used by the team at MIT as they have been **captured suitably** for the problem task.
- We augmented the dataset with a **custom set-up** of a speaker playing the audio over aluminum foil, captured by a smartphone.



The original set-up involves a high-speed camera behind a soundproof glass with sound playing over a bag of chips [Davis et al.]



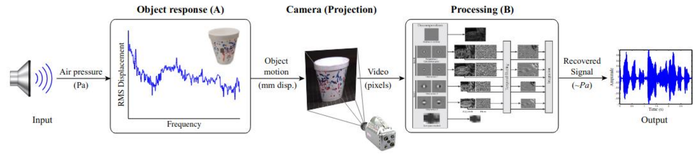
Our set-up involves a OnePlus 9 Pro shooting at 480 fps at 1280 x 720p, propped up on a tripod with two other phones illuminating it and a Bose speaker playing the audio over crumpled aluminum foil.

- We then extracted an **image dataset** of approximately **99,927 frames** from all the input videos.
- We simultaneously **extract audio samples** from a .wav file and **sub-sample** it to ensure that every frame has its own audio sample.

3. CHALLENGES

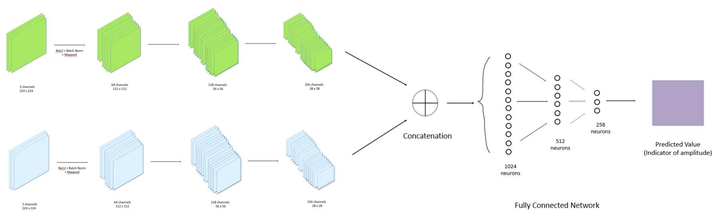
- Acquiring **sufficient, diverse**, and **relevant** data.
- Building a custom model that can detect even **the most minor edge changes** in objects.
- The input video sizes and frame datasets are too large.
- The model's **computational overhead** requires a lot more resources than available.

4. METHODS



High-level overview of the process flow to synthesize a waveform based on the input video recording [Davis et al.]

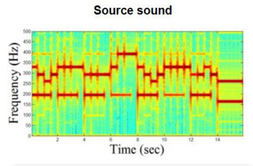
- The video is passed through a **complex steerable pyramid**, which is a mathematical model that generates **scale and orientation invariant feature maps** for every frame in the video.
- Before training the model, we take the extracted frames from the dataset and **concatenate** it with the **correct reference frame** (usually the first frame of the video)
- We **normalized the frames** by the mean and standard deviation of the training dataset. We also **normalized the audio samples** to bring the range of values down from (-32k, 32k) to (-1, 1).



Siamese Network with custom CNN architecture

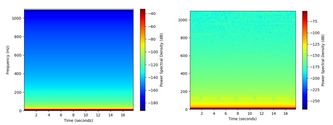
- We propose a **Siamese Architecture** to simultaneously generate **feature maps** for the reference frame and the current frame. We concatenate these feature maps and feed it into a **fully connected network** which approximates the **sound sample for the current frame.**
- We have used two different types of CNNs to extract features from the image. The first CNN is **ResNet50** with pre-trained weights and the other is a **custom CNN** designed to **extract low level features** through a variation of **successive 3x3 kernels**.
- The loss metric for training the network is **Sum of Squared Differences**, which is essentially Mean Squared Error with the reduction set to sum.
- The metric for understanding the performance of the model is **Root Mean Squared Error**, which gives us an idea about the average error across all our samples.

5. RESULTS



The input sound file: A 15-second clip of "Mary had a little lamb" capped at 480 Hz [Davis et al.]

- Our reconstructed code from the MIT experiment was able to generate and recreate the original sound to a sufficient degree, but our deep learning augmentation **failed to do so**.



The spectrogram generated by the ResNet model and Custom CNN model respectively

- The **net loss value** incurred by both models appear to be **very large**.
- The Average RMSE suggests the predicted values and actual values are off by ~0.48 and ~0.37 in ResNet and Custom CNN respectively.
- This isn't reflected in the **spectrogram** because the variations in the actual values are **so high** that on average the values **don't differ by a huge margin.**

Model	Epochs	Test Size	RMSE	Loss
ResNet	10	38,985 (39%)	0.489	19.143
CNN	10	38,985 (39%)	0.371	18.366

Table comparing the RMSE and loss for both models within the Siamese network

6. CONCLUSION & FUTURE WORK

- So far, our deep learning approach can **not effectively predict** the **amplitude of sounds** for each frame and **reconstruct the sound**.
- ResNet is not able to determine the **subtle differences** between subsequent frames and the custom CNN is able to distinguish more, but is unable to capture the **lower-level features** to a **greater degree**.
- The **data set is limited** so we can try to enhance the model by feeding it **more data that we record**.
- We can **fine-tune** the deep learning model and try different models for a **comparison-based approach**.

References

The Visual Microphone: Passive Recovery of Sound from Video by Abe Davis, Michael Rubinstein, Neal Wadhwa, Gautham Mysore, Frédo Durand, William T. Freeman