How Loud are Silent Films?

Roderick Huang, Aryan Kumar
Massachusetts Institute of Technology
77 Massachusetts Avenue Cambridge, MA 02139
rwxhuang@mit.edu, aryan02@mit.edu

Abstract

From movies to video games, sound plays a crucial role in our perception of the world. In this project, we aim to predict the volume of visually indicated sounds (VIS) from silent video scenes. This problem is intriguing because its completion will bring us one step closer to the general task of predicting visually indicated sounds, which can help in numerous applications from generating the accompanying sound in silent films to producing sound effects for video entertainment. We present a model involving a recurrent-neural network composed of a CNN and an LSTM, trained on an existing drumsticks dataset and a novel impacts dataset of our making. We find that our model generally predicts the occurrences of auditory inflection points in video correctly, but needs further improvement in estimating the volume of the inflection points accurately.

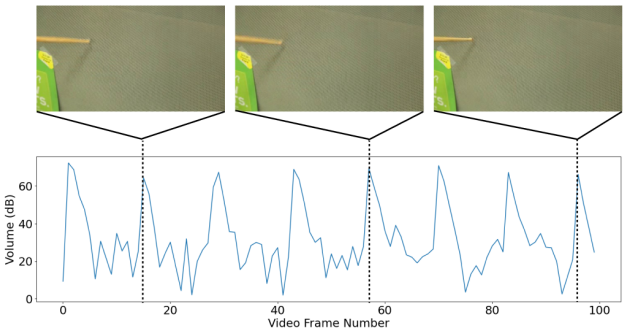

Figure 1. We train a model whose goal is to predict volume in the form of decibel (dB) values from silent video. Each video contains visual indications for sources of sound. As depicted above, for 100 frames of a sample video, a few frames are displayed corresponding to the vertical dashed lines.

1. Introduction

There are many types of sound we encounter on a daily basis. Some are ambient/held-out sounds, and some are visually indicated sounds. For example, the sound of someone swinging a baseball bat into a wall would be a visually-indicated sound, as there is a physical interaction that indicates the sound; an ambient or held-out sound, on other hand, could be the background sound of waves crashing at a beach.

In this project, our goal is to predict the frame-by-frame volume levels for a silent video with visually indicated sounds. We focus on visually indicated sounds as there is a direct relationship between the image data in the video and the sound produced.

There has been a decent amount of work done in the field of visually indicated sounds. However, much of the work focuses on accomplishing tasks for a very narrow domain (i.e. for only specific types of videos). We instead aim to solve the slightly simpler task of frame-by-frame volume prediction for a broader domain of video data. We hope this will bring us one step closer to the general goal of predicting visually indicated sounds, and that this will further the ability of modern computer vision algorithms to coherently relate various forms of sensory input together.

We first process the drumstick and novel dataset and generate the associated volume data (in decibels) for each video. We then train a model consisting of an LSTM and a CNN on the data. Finally, we evaluate our model's performance by producing plots of the predicted volume vs actual volume, and by computing error statistics.

2. Related Work

Due to the interest in the field of sound generation from video, there exists a modest amount of related work.

Processing Visual Data Attempts to predict sound from video commonly comprise of multi-level architectures with CNNs at the bottom. For example, [1] utilizes a CNN with clustering and [2] utilizes a CNN with an LSTM in its model architecture. These architectures use a CNN to generate an embedding of the image data for the later layers to work with directly. This removes the need for the later layers to

understand the visual data.

Capturing Motion and State [2] utilizes an LSTM to capture state in the video. As the image embeddings of each frame is fed, the LSTM updates its state and captures the relevant information needed (including motion information) to predict the sound cochleagram for future frames. [2] furthermore does not simply pass in frames to the model's CNN for processing. It instead passes in a space-time image to the model, which consists of a grey-scale version of the previous, current, and next frames of the video in addition to the first color frame. The first color frame is passed in to provide a baseline about the color information in the image, but no additional color frames are passed in for efficiency. Adjacent frames are passed to the model, since it succinctly captures the instantaneous motion occurring at the current frame, which is essential for the model to accurately predict sound.

Predicting Sound [1] clusters similar image embeddings to predict statistics on the ambient sound present in the image. The authors' model produces results equivalent to other state of the art models in this field, but their model predicts general sound statistics rather than giving information about the sound at any given instance. On the other hand, [2] utilizes a CNN-LSTM architecture as previously discussed to generate frame-by-frame cochleograms (sounds) for videos of drumsticks hitting various objects. The authors' model did produce sounds capable of fooling humans, but it was for a very specific type of videos.

Broadening the Domain [2]'s approach does successfully work for their intended task, but it must be noted that this was for a very limited domain. It's highly likely that if videos of other scenes were tested on their model, the results would be poor. The same is true for [1]. To ensure that a model can understand a broader scope, a more general dataset should be constructed as done in Section 3.

3. Dataset

Our dataset requires videos that clearly display movement to visually indicate sources of sound. In this study, we utilize two separate datasets that simplify the goal. The first is the drumstick video dataset from [2] that contains denoised videos of drumsticks hitting various objects. The second dataset is a novel dataset of our own construction called the impacts dataset, containing videos of various objects hitting (or colloquially impacting) a surface. We produced this dataset to try and broaden the scope of visually indicated sounds our model could learn to predict the volume of. As part of the impacts dataset, we recorded videos from our iPhones including scenes of hitting a desk with a water bottle, hitting a desk with a hand, kicking a wall, etc.

To construct our dataset to have image frames as input and dB levels as output, multiple Python scripts were implemented.

Image Frames To extract image frames from each video file, we utilized the Python OpenCV library to clip image files. From the drumstick dataset, we extracted each image frame from every $\frac{1}{15}$ second of each video. From the custom dataset, we extracted each image frame from every $\frac{1}{30}$ second of each video. Due to the limited time to train our model, we decided to take a window of 100 image frames from each video of our dataset to reduce the amount of computation.

Volume Output Using the Python SciPy I/O wavfile library, each video file was clipped to produce chunks of sound. To extract the dB levels from each chunk, we used the following formula:

$$\text{dB} = 20 \cdot \log_{10}\left(\sqrt{\Sigma_{i=1}^{n}\text{chunk}[i]^2}\right) \quad (1)$$

To relate to the image frames, we took the corresponding window of 100 dB values. As a result, each input image frame k_i from a video has a corresponding dB level y_i as output.

Larger-scale Datasets The VGG Sound dataset from [3] in the project assignment offers a large-scale audio-visual dataset containing more than 200,000 videos extracted from YouTube clips. However, we believed there was a lack of clear visually indicated sounds from many videos, and we think that a future study on general sound generation may be better suited to using this dataset. In addition, much more computational resources than we have access to are required to deal with more robust datasets such as this.

4. Methodology

As a regression problem, given a sequence of image frames $k_1, k_2, ..., k_n$ that display visual indications of sound, we would like to predict the corresponding volume dB levels $\hat{y}_1, \hat{y}_2, .., \hat{y}_n$. Figure 2 visually explains our methodology. Our model's architecture will consist of a CNN and an LSTM. The CNN will be used to develop an embedding of the image data, and the LSTM will be used to capture state and motion in our video. Our goal is to produce a CNN-LSTM model capable of predicting the frame-by-frame dB levels of a silent video.

4.1. Pre-processing

Spacetime Image Inputs For each image frame k_i, we constructed a corresponding feature vector x_i to be the input to our model. Since our model discussed further in 4.2 utilizes Resnet, we first made the following transformation f to each image frame: grayscale, resize, and scale (between -1 and 1). Then, to capture the instantaneous motion around image frame k_i, we construct a spacetime image (as described in [2]) to be our input feature vector x_i. The three channels of the space-time image will be the grey-scale ver-

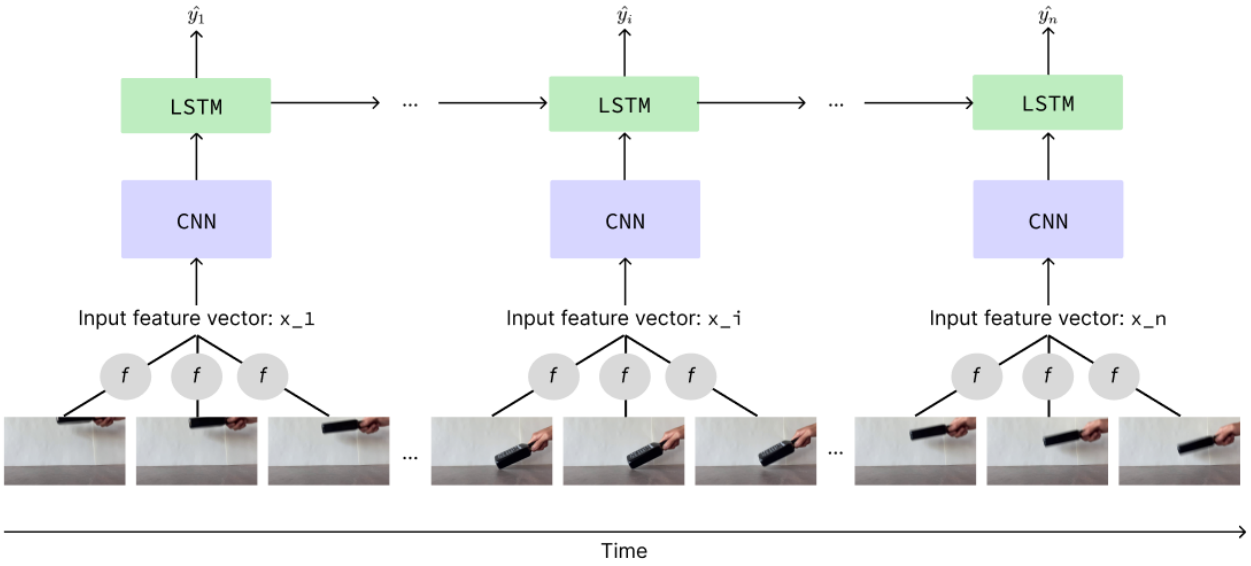

Figure 2. Illustrates the methodology used. Given the image frames of a video, they are transformed and combined to form input feature vectors. By running through a CNN-LSTM model, it will predict the volume of the frame by recognizing features through the CNN and memorizing states through the LSTM.

sions of the previous, current, and next frames. x_i can thus be written as follows:

$$x_i = \begin{bmatrix} f(k_{i-1}) \\ f(k_i) \\ f(k_{i+1}). \end{bmatrix}$$

Blur and Downsampling Due to computational limitations, we also had to blur and downsample the image frames in both our datasets. The frames in the drumsticks dataset were originally 256 by 456 pixels, and we reduced their resolution to 64 by 114 pixels. Similarly, the frames in the novel impacts dataset were reduced from 1080 by 1920 to 224 by 224.

4.2. CNN-LSTM Model

We feed each feature vector x_t into our CNN to generate an embedding of the spacetime image data. We utilize Resnet-101 for our CNN, due to its accuracy and conciseness due to simple kernel compositions. We then feed the embedding into our LSTM to update its state, and the LSTM will output its new state h_t as a vector. We perform a simple linear transformation (whose parameters will need to be learned) on h_t to get the predicted scalar volume in decibels for the current frame.

We trained our model using Google Collab Pro, on both the drumsticks and novel impacts dataset. We utilize mean-squared error (MSE) as our loss function, since this is a standard regression problem and MSE penalizes larger errors more heavily. This will ensure that there is a significant penalty to predicting silence during the brief spikes in volume in our video, thus pressuring our model to learn more complex behaviour. For performance metrics purposes, we keep track of the mean-squared error through each epoch of training the model.

Due to the lack of publicly available implementations, we programmed our model from scratch.

5. Experimental Results and Discussion

We applied our CNN-LSTM model to two datasets, and evaluated it with a combination of qualitative and quantitative metrics.

5.1. Drumstick Dataset Results

Given that [2] conveniently provided the drumstick dataset, we first trained and evaluated our CNN-LSTM model on this. Figure 3 shows the volume predicted by our trained model alongside the actual volume (in decibels) for a video from this dataset. Unfortunately, as can be seen on the figure, the predicted volume is relatively constantly throughout this video, with very minor variation throughout. We found this to be the case for other videos as well, with our model consistently predicting roughly the mean volume for a video.

Given a mean-squared error objective function, the optimal constant prediction for a series of values is their mean. Thus, we hypothesized that our CNN and LSTM layers were producing a roughly similar output for all of the frames in our video, and the final linear transformation ap-

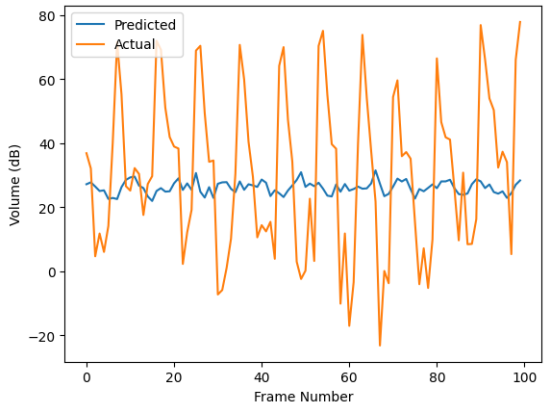

Figure 3. Prediction by our CNN-LSTM model for a sample video. Our model performed poorly on the drumstick dataset as we believe it was hard to recognize the source of movement from the drumstick against a blurry complex background.

plied to this output was learned to just produced the mean. This was indeed the case.

Upon investigation, we found the reason our CNN and LSTM layer would output roughly the same thing throughout the video, even in frames with and without the drumstick, was not due to an error in model construction. Due to the computational limitations of our project, we blurred and downsampled the drumstick dataset's frames to reduce their resolution by a factor of 16. Since the drumstick is a relatively slim object, it became almost unnoticeable to even the trained human eye in the lower resolution images, thus making all of the image-frames appear similar.

5.2. Impacts Dataset Results

As mentioned previously, we produced a novel dataset called the impacts dataset. This custom dataset contains clips of a water bottle hitting a table, a trash can being kicked, etc. and was created to broaden the scope of visually indicated sounds our model could predict the volume of.

Figure 4 shows the predicted volume and actual volume (in decibels) on the same plot for two randomly selected videos from this dataset. As seen in the figure, our model predicts an inflection in the volume at the correct time, and predicts a relatively low volume for the remainder of the time in the videos correctly. However, our model did not accurately predict the volume of the inflection in these videos. We found the observations discussed in this paragraph held for our model in general.

We tried modifying our architecture and our training approach in the following ways to improve our results:

- Using the Adam vs the SGD optimizer
- Training both the CNN and LSTM together, vs setting the CNN to use the pre-trained ImageNet classification weights and only training the LSTM.
- Increasing and decreasing the number of features and layers in the CNN and LSTM in our model.

We evaluated these different approaches quantitatively by examining the validation loss after a certain amount of epochs, and qualitatively by producing a plot similar to figures 3 and 4 with the predicted and actual volumes for a video plotted. We found that the choice of the optimizer did not make any noticeable difference in our model's performance. In addition, we found that adjusting the number of layers in our LSTM between 1 to 3 did not affect our model's performance, and neither did scaling the number of features in our CNN up or down by a factor of 2.

However, we did find that using the default ImageNet weights for our CNN and only training the LSTM did perform better than training both the CNN and LSTM from scratch. Our model's validation loss started at a lower value and continued to fall with the former approach, whereas with the latter approach our model's validation loss started high and very quickly stagnated. We believe this may have been caused by exploding or vanishing gradients as the Resnet101 CNN in our model is several layers away from the output. Thus, it may have taken several hundreds of epochs to train the CNN from scratch, which we did not attempt due to computational limitations. The results shown in Figure 4 were achieved with the former approach.

Despite trying several different approaches to improve our model's performance, we found our model still struggled with accurately estimating the volume of sound inflections. This may partially be attributed to computational limitations. We found that our model's validation loss consistently fell, even when we stopped training around 25 epochs, and performance may have further improved if we continued training. Figure 5 shows the training and validation loss as the number of epochs our model is trained increases. In addition, we also had to lower the resolution of the videos in the impacts dataset substantially. While it was still possible to gain a coarse understanding from the lower-quality dataset, a higher resolution and higher frame rate may be needed for our model to better ascertain the instantaneous speed during the impact and the severity of impact as well, which is needed to predict the volume of those sounds.

A future study with access to greater computational resources could be very helpful in testing these potential ideas to improve our model. In addition, it could train our model on an even larger impacts dataset to broaden our model's learned domain even further.

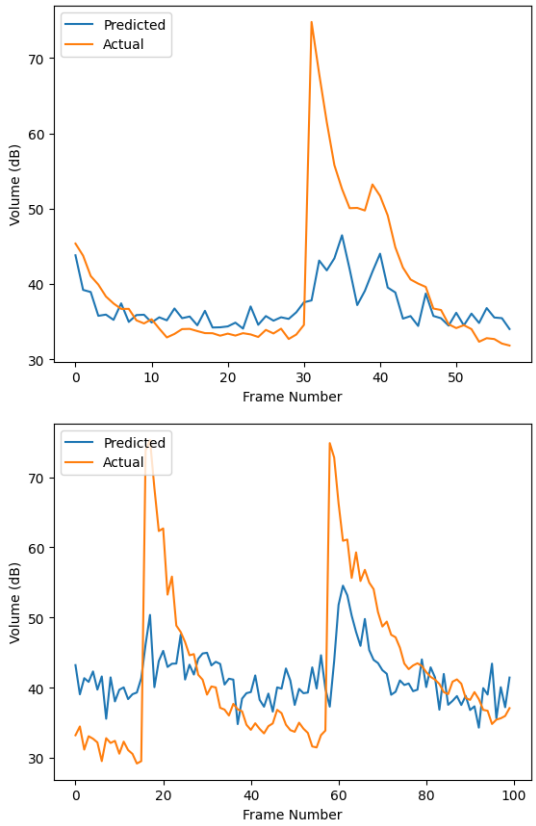

Figure 4. Displays graph of running our CNN-LSTM model on the Impacts dataset. Our model was successful in predicting the localities of inflection points, but more work is needed to accurately predict the sound intensities at those points.

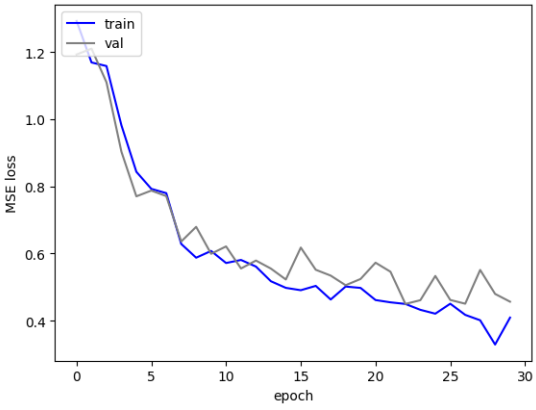

Figure 5. Training and validation loss of our model for training on the impacts dataset, with a normalized output. As illustrated, both the training and validation error have a decreasing trend, showing that with more epochs, it's likely that the model can continue to learn the data better.

5.3. Additional Discussion

Broadening the Scope By constructing a custom impacts dataset, we broadened the scope of video scenes when predicting volume. While we've improved on prior work by expanding the possibility of training more general video scenes, we've also shown that the drumstick dataset performed poorly despite being niche. Since the videos were affected more greatly when blurring and downsampling the drumstick dataset, our impacts dataset improved the model's learning greatly as seen with the losses shown in Figure 5 by recording higher-quality videos with more prominent visually indicated sounds. By combining prior work, an interesting area of research is to consider sources of visually indicated sounds that are more distorted or harder to recognize.

Limitations As discussed in detail in 5.2, despite producing promising results, our computational resources limited the extent the model could be trained for. Since the goal of our impacts dataset broadened the scope to include various video scenes, the model required more time already in the learning process. The GPU we utilized was only allowed to train our model for 10 hours at a time, but access to more robust computational resources could greatly help our model in learning general video scenes even better.

6. Conclusion

In this project, we proposed the problem of predicting volume from visually indicated sounds for video scenes in general. We created the impacts dataset that contained videos of various scenes showing clear indications of movements that produced sounds. The impacts dataset took a step beyond prior work by broadening the scope of videos. Through our constructed CNN-LSTM architecture, we developed an algorithm to predict the volume from our impacts dataset. The evaluation of the quality of our approach was done both qualitatively and quantitatively. We believe our results not only contribute new insights to visually indicated sounds, but also expose the limitations that are hard to overcome in this area of study.

Through our experiments, we've shown cases where our CNN-LSTM model can produce poor results when predicting volume, especially for the drumstick dataset. By investigating the data, we realized that by blurring and downsampling the frames, the diminished resolution caused sources of sound to be unrecognizable. Since this made all the images appear similar, it was hard for the model to learn what was causing significant sudden increases in volume. A possible consideration for further research is to study videos of lower resolution to predict volume. While a challenging task, it would be a useful and intriguing application to predict sound from silent films from before the 1930s which had much lower-quality videos.

Since our impacts dataset improved upon lower-resolution videos, a significant tradeoff was performance. For more modern applications such as foley sounds in the movie industry, improvements in camera technology are producing videos of higher resolution. With higher resolution, our model would require more robust computational resources. In addition, different model architectures can be explored. While a CNN-LSTM architecture can allow the model to learn about changes in each image frame over time, a more robust architecture could help represent more complex factors in videos. All in all, we believe this project has furthered the progress made in the field of sound generation from visual media and opens up many more avenues of exploration.

7. Individual Contributions

The authors of this project collaborated on all aspects of this project, including in designing the architecture, constructing the dataset, and writing the report. Aryan specifically implemented the CNN-LSTM architecture, and wrote the abstract and sections 1, 2, 4, and 5. Roderick performed the data processing, trained the model, constructed the visuals, and wrote sections 3, 5.3, and 6.

8. Acknowledgements

We would like to thank the 6.8301 staff for their continual feedback and support in this project. We received feedback on both communication and technical aspects of this project, which helped us fine tune and create the final paper we have now.

References

[1] J. H. McDermott W. T. Freeman A. Owens, J. Wu and A. Torralba. Ambient sound provides supervision for visual learning. *European conference on computer vision*, pages 801–816, 2016. 1, 2

[2] J. McDermott A. Torralba E. H. Adelson A. Owens, P. Isola and W. T. Freeman. Visually indicated sounds. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2405–2413, 2016. 1, 2, 3

[3] Andrea Vedaldi Honglie Chen, Weidi Xie and Andrew Zisserman. Vggsound: A large-scale audio-visual dataset. *VGG, Department of Engineering Science, University of Oxford, UK*, pages 1–5, 2020. 2