Artificial Synthesis of Sound Tracks for Silent Videos

Presented by: -

Swaraj Priyadarshan Dash

B.Tech (CSE), 7th Semester

Regd. No: 1701209365

Silicon

...beyond teaching

Silicon Institute of Technology

Bhubaneswar

Silicon

Table of Contents

1 Abstract

Objects make distinctive sounds when they are hit or scratched. These reveal aspects of an object's material properties, as well as the actions that produced them. The task is of predicting what sound an object makes when struck within a visual scene. The algorithm will produce a sound for the hit that is realistic enough to fool human viewers. This "Turing Test for sound" represents much more than just a clever computer trick. Researchers envision future versions of similar algorithms being used to automatically produce sound effects for movies and TV shows, as well as to help robots better understand objects' properties. The important performance criterion of the synthesized sound track is to be time synchronized with the input video. This provides for a realistic and believable portrayal of the synthesized sound, enough to fool a person. The basic algorithm uses a recurrent neural network to predict sound features from videos and then produces a waveform from these features with an example-based synthesis procedure. Deep Learning techniques are generally used in these kinds of problems which involve teaching computers to sift through huge amounts of data to find patterns on their own. Deep learning approaches are useful because they provide freedom to hand-design algorithms and supervise their progress. This research area is an extension of Automatic Image Captioning projects, that combine the two sensory perceptions, in the form of images and audio. The idea, however, is the same, that is to find context, and build a model to learn and reproduce. The scope for this research idea is huge: it can be used to generate audio for video clips without audio, or unclear audios. It can also be used as the Foley Artist, that is responsible for creating an overlay soundtrack that helps movie scenes come alive for the audience. The topic idea is inspired by independent research done by researchers from MIT's Computer Science and Artificial Intelligence Laboratory (CSAIL) and Microsoft Research Fellowship.

2 Introduction

Adding sound effects in post production using the art of Foley has been an intricate part of movie and television soundtracks since the 1930s. The technique is named after Jack Foley, a sound editor at Universal Studios. Mr. Foley was the first to make sound effects for live radio broadcasts with the tools and items he had around him. Now almost every motion picture and television show contain Foley tracks.

Movies would seem hollow and distant without the controlled layer of a realistic Foley soundtrack. To construct the augmented sound, the Foley artist uses special sound stages

surrounded by a variety of props such as car fenders, chairs, plates, glasses as well as post production sound studios to record the sound effects without the ambient background sounds. This requires electronics such as monitors, camcorders, mikes, manual record volume controller, and an external audio mixer.

Foley artists have to closely observe the screen while performing diverse movements (e.g. breaking objects, running forcefully on rough surfaces, pushing each other, scrubbing different props) to ensure their sound effects are appropriate. The process of Foley sound synthesis therefore adds significant time and cost to the creation of a motion picture. Furthermore, the process of artificially synthesizing sounds synchronized to video multi-media streams is a problem that exists in realms other than that of the Motion Picture industry.

Figure 1. Foley Recording Studio

Research has shown that the end-user experience of multimedia in general is enhanced when more than one source of sensory input is synchronized into the multimedia stream. We generate the augmented sounds for injection into a synchronized video file by first utilizing a deep neural network.

To train our network, we find motivation from the multimodal coherence ability of the human brain to synchronize audio and visual signals. Recent works explored the relationship between auditory and visual modalities through computational models as way of localizing and separating sound source, material and action recognition, generating natural sounds and learning audio features for video analysis. In this seminar, we propose a deep sound synthesis network for the first time that performs as an automatic Foley, generating augmented and enhanced sound effects as an overlay on video files that may or may not have associated sound files.

Figure 2. Foley Recording for movie at Pinewood Studio

3 Methodology

3.1 Sound Feature Extraction

We first compute the features of all the audio files using spectrogram analysis; a visual way of representing the strength of a signal over time at different frequencies present. Human hearing is based on a form of real-time spectrogram encoded by the cochlea of the inner ear, we convert the audio signal into a 2D representation (spectrogram) for extracting the audio feature. Our spectrograms provide an intensity plot (in dB) of the Short-Time Fourier Transform (STFT) magnitude of the windowed data segments along with the window width w by computing as follows:

$$\text{spectrogram}(t, w) = |STFT(t, w)|^2$$

The windows are usually allowed to overlap by 25%-50% in time. We use 950 frames since our sampling frequency is 44100 Hz. A Hanning window is used to compute STFT, and each audio feature vector Sn(t) has a width of 1 and a height of 129. The brighter the color, the more energy is present in the audio at that frequency.

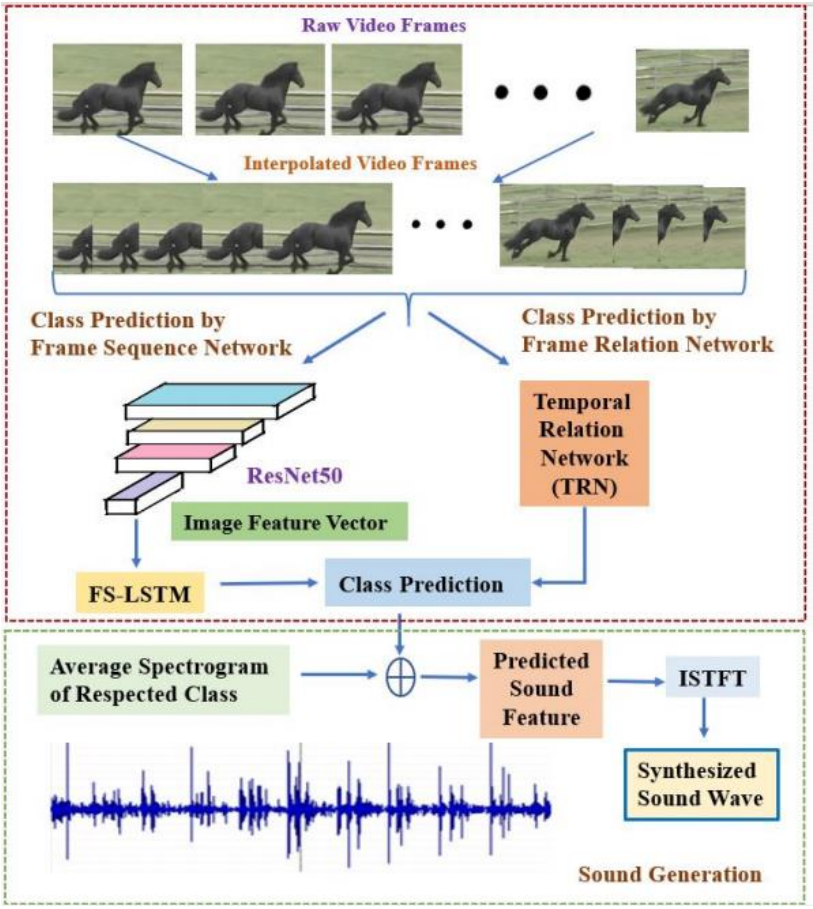


Figure 3. Automatic Foley Generation Architecture

3.2 Sound Class Prediction from Video

We present two different approaches for predicting the sound class from input video frames: i) Frame-Sequence Network (interpolation technique followed by combined convolutional neural network (CNN) and Fast-Slow LSTM (FSLSTM) network and ii) Frame-Relation Network (combination of CNN and temporal relational network (TRN)).

3.3 Approach 1: Frame Sequence Network

In this approach, we increase the video frame rate capturing detail motion information of video frames by using interpolation technique. Next, we extract the image features from each interpolated video frame by applying CNN (ResNet-50). Finally, we predict the sound class associated with the video clip by recurrent network (FS-LSTM) using the image features obtained from the bottleneck convolutional layer of the residual network.

3.3.1 Video Preprocessing using Interpolation Technique

To map image features with audio features, we face the problem of equalizing the video frame numbers with audio sample numbers. In our dataset we find, the total number of audio samples is about 13 times higher than the total video frame number. In earlier works each image feature vector was replicated k times (where k is the ratio of audio and video sampling rates) to compensate for the difference between the two sampling rates.

However, this replication technique sometime fails to store the precise motion information of videos correctly, thus we lose some action relationships between consecutive frames. We solve this problem exploiting the technique of generating intermediate video frames by using interpolation algorithm during the video pre-processing step before moving towards the image feature extraction step.

Video frame interpolation technique can estimate dense motion, typically optical flow between two input frames, then interpolate one or more intermediate frames guided by the motion. Because of this interpolation, we successfully increase the video's frame rate as required without changing the video length and losing the inter-frame motion information.

3.3.2 Generating Image Feature Vector Using CNN

We aim to compute image feature vectors with the ResNet-50 convolutional neural network (CNN) model. For this, first we read a sequence of interpolated video frames *I1, I2 ,…, In* from training videos as visual inputs excluding corresponding sound tracks. To avoid the complexity found in the two stream approach while obtaining accurate flow estimates in case of fast, non-rigid motion videos, we create space-time images (SP1, SP2, ..., SPn) for each frame by joining three grayscaled images of the previous, current, and next frames.

Figure 4. Space-time Image Generation: Concatenation of three consecutive 224×224 resized grayscale frames of horse racing video after applying interpolation technique

Finally, we compute the input feature vector Vt for each frame t, by concatenating the CNN features for the spacetime image (SPt) at frame t and the RGB image from the first frame of each video:

$$Vt = [f(SPt);\ f(I1)]$$

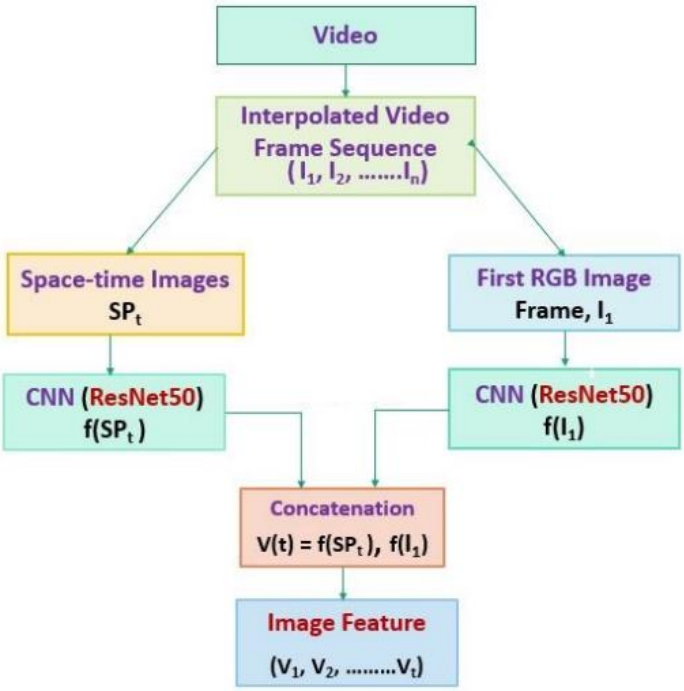


Figure 5. Image Feature Extraction in Frame-Sequence Network applying Convolutional Neural Network (CNN)

3.3.3 Sound Class Prediction Using FS-LSTM

We use the image features (Vt) found from our CNN as input to a special recurrent neural network (RNN) named as Fast-Slow LSTM (FS-LSTM). Here, LSTM cells act as building units to map the video frames to audio frames and compute the predicted sound features at each time instance. The FS-LSTM network has two hierarchical layers.

The lower layer contains n number of sequential LSTM cells (L1,L2,...,LN) considered Fast cells, whereas the upper layer consists of only one LSTM cell (U) considered the Slow cell. U takes input from L1 and gives its state to L2. In the lower layer, the image feature vector Vt is fed to the first LSTM cell (L1) as input and the final predicted sound class matrix is computed as the output of the last LSTM cell (LN).

For arbitrary LSTM cells L1, L2,...,LN and U, the FS-LSTM architecture (Fig.8) can be expressed in the following set of equations:

$$H\,L1\,t\; =\; \beta\,L1\,(H\,Ln\,t-1\,, Vt)$$

$$HU\,t\; =\; \beta\,U\,(HU\,t-1\,, HL1\,t\,)$$

$$H\,L2\,t\; =\; \beta\,L2\,(H\,L1\,t\,, HU\,t\,)$$

$$H\,Li\,t\; =\; \beta\,Li\,(H\,Li-1\,t\,);\; 3\;\leq\; i\;\leq\; N$$

Here, β represents the update function of a single LSTM cell. The encoding process is performed by revising the value of hidden vector (Ht) with updated image feature vector Fig. 8. Fast-Slow LSTM network with i Fast cells. (Vt). Finally, the output sound class prediction matrix (sc1) is calculated from affine transformation of H LN t:

$$sc1\; =\; W\;HLN\,t\; +\; B$$

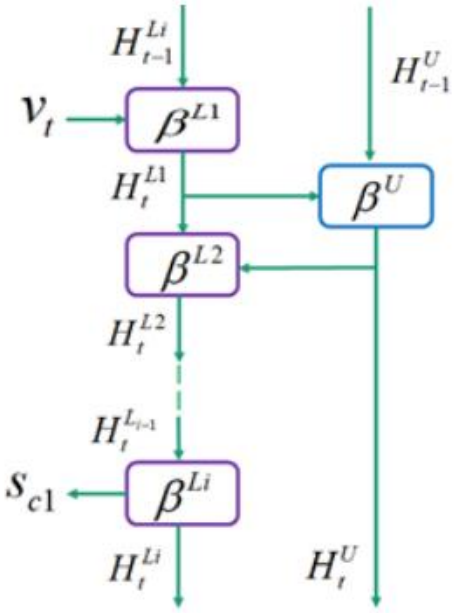


Figure 6. Fast-Slow LSTM network with i Fast cells

3.4 Approach 2: Frame-Relation Network

We aim to capture the detail transformations and actions of the objects, present in the movie scenes more accurately with less computation time. So, to learn the model about the complex behaviors of a visual scene with less amount of video frames, we apply an interpretable network combining CNN and MultiScale Temporal Relation Network (TRN). The network compiles the temporal relations among the frames at different time scales by using the following equation:

$$MRQ(v) = R2(v) + R3(v) + \ldots + RQ(v)$$

Here, R2, R3,...,RQ are the relation functions that capture the temporal relations between 2, 3,...,Q number of ordered frames of video v with respectively. We compute the relation over time among up to 8 frames by setting Q equals to 8. If the video v contains r number of selected frames as v = [f1, f2, ..., fr], we can define relation functions as follows:

$$R2(v) = h\varphi(X\,j)$$

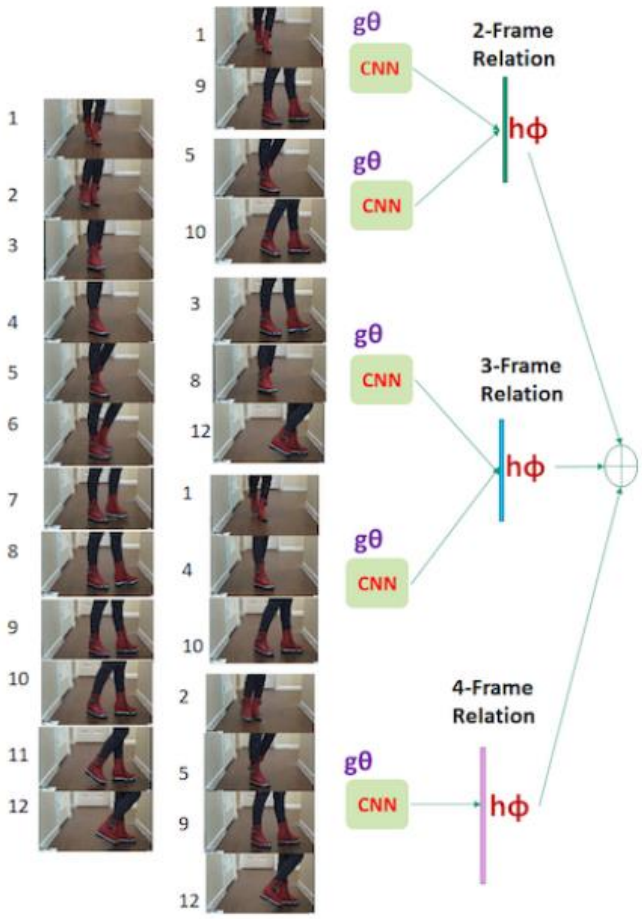


Figure 6. Multi-Scale Temporal Relation Network (TRN)

3.5 Sound Synthesis

We apply the same sound synthesis method on both of the sound class prediction approaches. We take the average of all spectrograms of each sound class in our training set, that we combine with the predicted sound class matrix sc computed from the frame sequence and the frame relation network separately.

For the Kth sound class, we find the predicted sound feature, s 0 (K) from its average spectrogram (AK):

$$s\,0\,t\,(K) \;=\; sc \;+\; AK$$

We reduce the difference between the actual the and predicted sound feature for every timestep, calculating the robust loss L through prediction of the square root of the sound features:

$$L(\gamma) = log(\alpha + \gamma\, 2\,)$$

$$s\, 0\, t\, (K) = sc + AK$$

For sound synthesis from predicted sound feature vectors (~s 0 t), we perform the Inverse Short Time Fourier Transform (ISTFT) method with Hanning window because of its less computational complexity than parametric synthesis approach or example based synthesis method. The inverse STFT (ISTFT) can regenerate the time-domain signal exactly from a given STFT without noise. We calculate ISTFT by inverting the STFT using the popular overlap-add (OLA) method for better denoising efficiency and synthesis quality as shown below:

$$y(t) = \frac{1}{2\pi} \int\, Y\, (\tau, w) e + jwtd\tau\, dw$$

For the phase reconstruction from spectrogram, we apply the iterative Griffin-Lim algorithm for 16 iterations while performing ISTFT. The proposed automatic Foley Generation Model details are summarized in Algorithm as follows:

Algorithm 1 Automatic Foley Generation

Input: Silent video frames (v_i) and training audio tracks (A_j).
Output: Generated audio tracks ($Aud_{generated}$).

for $j \leftarrow 1, 2...N$ **do**
  $S_j \leftarrow Spectrogram(A_j)$
end for
$Avg \leftarrow Average(S)_{eachclass}$
if $FrameSequenceNetwork$ **then**
  for $i \leftarrow 1, 2...N$ **do**
    $I_t \leftarrow interpolate(v_i)$
    $SP_t \leftarrow spacetime(I_t)$
  end for
  $V_t \leftarrow concatenate(CNN(SP_t), CNN(I_1))$
  $Model \leftarrow FSLSTM(V_t)$
  $Prob \leftarrow Model.predict(V_{test})$
end if
if $FrameRelationNetwork$ **then**
  for $i \leftarrow 1, 2...N$ **do**
    $I_t \leftarrow interpolate(v_i)$
  end for
  $V_t \leftarrow CNN(I_t)$
  $Prob \leftarrow TRN(V_t)$
end if
$S' \leftarrow Prob + Avg$
$Aud_{generated} \leftarrow ISTFT(S')$

4 Experimental Result

For our model evaluation, we create a video dataset particularly focusing on Foley tracks used for movies. We give a detail description of our dataset in subsection A. Next, we introduce the model parameters and implementation details in subsection B. Based on our dataset and models, we evaluate our generated sound both in qualitative and quantitative ways, described in subsection C and D respectively. Later in subsection E, we present a detailed ablation study of our methods and parameters. Finally, we discuss the results of four human evaluation survey questions on our synthesized sound in subsection F.

4.1 The Automatic Foley Dataset (AFD)

Our interest is to enable our Foley generation network to be trained with the exact natural sound produced in a particular movie scene. To do so, we need to train the system explicitly with the specific categories of audio-visual scenes that are closely related to manually generated Foley tracks for silent movie clips. We find different video datasets (e.g. GHD, VEGAS, AudioSet, UCF101) for the sound generation task. However, none of these datasets are able to serve our requirements for various reasons. For example, the GHD dataset consists of videos of human actions (such as, hitting, scratching), that are mostly focused on a material recognition task. On the other hand, large number of videos are accumulated in the datasets like AudioSet, VEGAS, and UCF101, but most of them contain either background sounds, human speech or environmental noises. Foley tracks are generated and recorded in a denoising environment inside the studio.

So we choose to create a dataset entirely focused on Foley sound tracks of movie events. Here we select 12 different categories of videos (that are frequently used for Foley generation) associated with clear sound tracks combining both indoor and outdoor scenes. We choose sound classes where we can record our own (e.g. cutting, footsteps, car passing, clock sound, breaking, etc.). For recording, we use a video camera along with a shotgun microphone system. Then we apply denoising algorithm used especially for outdoor recordings. For other popular Foley sound categories of movie clips such as, gunshots, horse running, waterfall, fire, rain, thunder sounds, we use YouTube and collect the most clear audio-video clips available with the least background noise. Altogether, our Automatic Foley Dataset (AFD) contains a total of 1000 videos from 12 different classes. The average duration of each video is 5 seconds. The twelve video classes and their associated data statistics are shown in Fig. 10 and 11 respectively.

Figure 8. The Automatic Foley Dataset (AFD) Video Classes: we chose 12 popular movie events where Foley effects are generally added in movie post-production studio

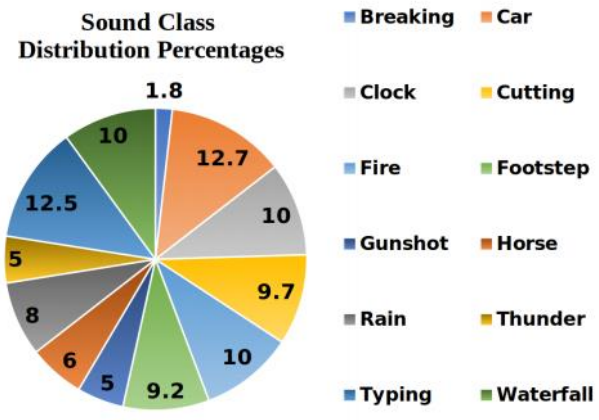


Figure 9. The Automatic Foley Dataset (AFD) Distribution: we show the data distribution percentages of 12 video classes in the pie chart

4.2 Implementation Details

We conduct the training separately, with our two approaches, using 80% of the videos, representing all 12 classes in the dataset. Our audio sampling rate is 44 kHz and video frame rate is 190 FPS (after interpolation). For video interpolation technique, we use an m-interpolation filter in the ffmpeg video editing package from Python.

In the first approach of sound classification, we get two (2048-D) image feature vectors from the output of conv5 layer of ResNet-50 network for both spacetime and first image frame from train videos. We concatenate these vectors to obtain the ultimate image feature vector of 4096-D that we pass to the FS-LSTM network as input.

We apply 4 LSTM cells in the Fast layer of FS-LSTM because of its optimum performance in the classification task. At each LSTM cell, we set an initial value of 1 for the forget-bias. We use orthogonal matrices for all weight matrices. We apply layer normalization separately to each gate. We use dropout to regularize the recurrent network. At every time step of FS-LSTM, we employ Zoneout in recurrent connections and a diverse dropout mask in nonrecurrent connections.

During training, we use minibatch gradient descent with the Adam optimizer. Our minibatch size and learning rate are 0.001 and 128 respectively. In the second approach of sound classification, we keep the training hyper-parameters for the CNN architecture the same as earlier.

In the TRN model, there are two layers of MLP (256 units in each) for $g\theta$ and a single layer MLP (12 units) for $h\phi$. The training for 100 epochs is completed in less than 18 hours on a Nvidia RTX 2080 Ti GPU. For testing our models, we choose 20% of random videos for all 12 categories. To test the TRN model for early action recognition purpose, we select the first 25% of frames in each video.

4.3 Qualitative Evaluation

4.3.1 Waveform and Spectrogram Analysis

For qualitative assessment, we present the synthesized sound waveform along with the corresponding spectrogram obtained from our two automatic Foley architectures with the same of original sound tracks in Fig.12. For some sound categories e.g. clock, rain, horse, we notice very similar waveform and spectrogram patterns of our generated sound to the original

sound. However, results from model 1 matches more precisely with the ground truth patterns than model 2.

We observe good alignment of synthesized and original sound for all sound categories that are less sensitive to meticulous timing (e.g. clock, fire, rain, waterfall). On the other hand, visual scenes containing random action variations with time (e.g. breaking things, cutting in kitchen, typing, gunshot in action sequences, lightning and thundering in sky) show few abrupt peaks and misalignment in the generated waveform. In video clips where sound sources are moving with distances (e.g. car racing, horse running, human footsteps), the sound amplitudes also vary with distance.

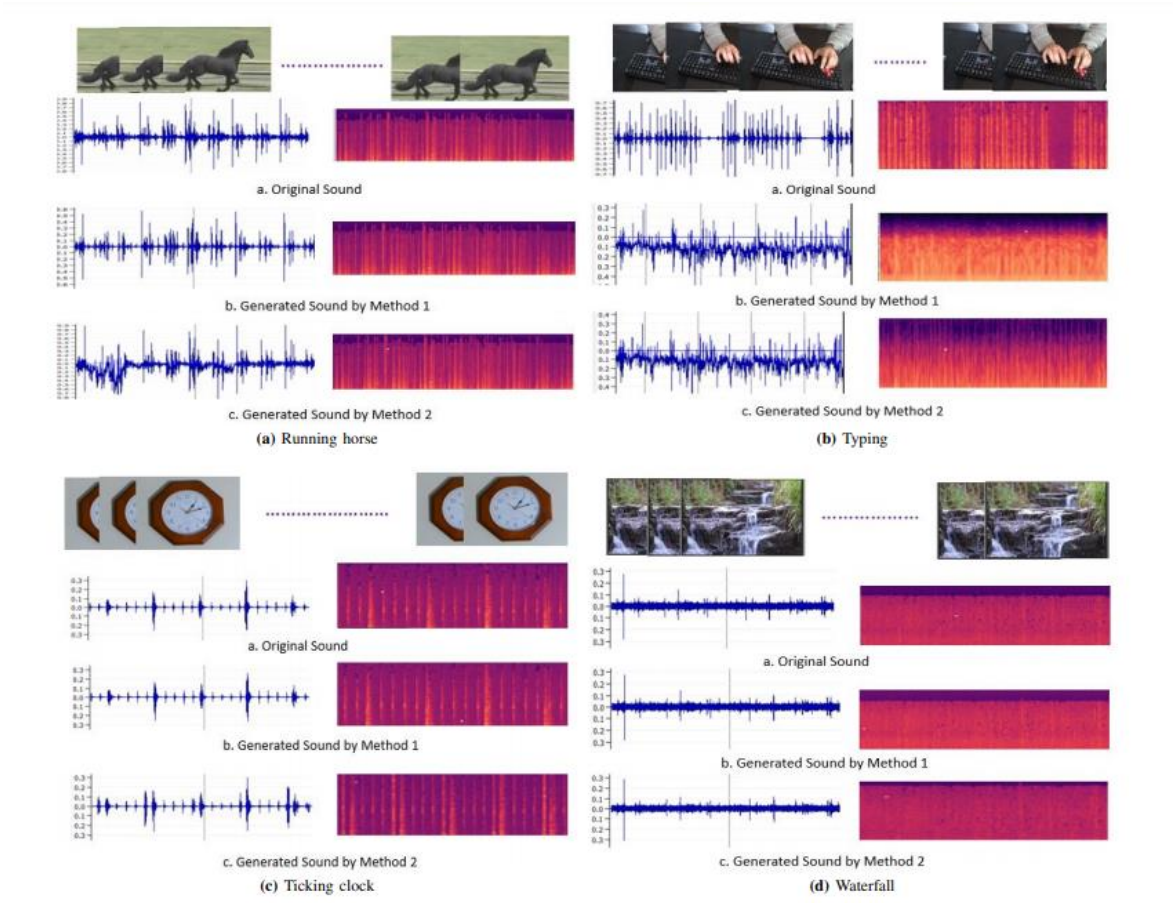


Figure 2. The waveform and spectral representation pair comparison between the original and generated sound

4.3.2 Sound Quality Matrix Analysis

Generally, the quality of a sound is assessed based on how well the sound conforms user's expectations. We choose to evaluate to what extent our synthesized sound waves are correlated with their ground truth tracks.

For each sound category, we calculate the average normalized cross correlation values between our generated and original audio signals to know how much similarity exists between them. We present the correlation values for models in Table I.

We notice all positive cross-correlation values (above 0.5) representing an expected analogy between the ground truth and our obtained result. Apart from the four most temporal action sensitive classes (e.g. breaking, cutting, footsteps, gunshots), model 1 provides higher correlation values than method 2.

SOUND QUALITY MATRIX ANALYSIS RESULTS: AVERAGE NORMALIZED CROSS-CORRELATION VALUE OBTAINED FROM COMPARING THE ORIGINAL AND GENERATED AUDIO SIGNALS FOR MODEL 1 AND 2 IN ALL SOUND CLASSES

Sound Class	Avg. Normalized Correlation Value	
	Model 1	Model 2
Break	0.76	0.93
Car	0.68	0.65
Clock	0.92	0.73
Cutting	0.58	0.89
Fire	0.88	0.72
Footstep	0.68	0.95
Gunshot	0.65	0.82
Horse	0.87	0.63
Rain	0.90	0.81
Thunder	0.71	0.58
Typing	0.69	0.60
Waterfall	0.86	0.69
Average	**0.77**	**0.75**

Figure 3. Sound Quality Matrix

4.4 Quantitative Evaluation

It is a difficult task to obtain absolute numerical evaluation of generated sound waveforms. In this section we measure the audio-visual coherence ability of our models in predicting sound from given visual inputs. We also provide the calculated loss and accuracy details while training and testing our models.

Sound Class Prediction: To visualize the sound class prediction accuracy from video frames, we present two normalized confusion matrices for model 1 and 2 in Fig. 14. It is worth notifying that Model 1 correctly classifies a majority of the audio samples of the given test videos of all categories, except for breaking (as it is trained with least number of train samples). However, our second model can successfully identify the breaking class from test videos as the TRN network is trained to recognize the action present in a visual scene using the least amount of video frames. However, it miss-classifies some audios in the rain and waterfall cases as well as in cutting and footstep test videos.

Loss and Accuracy Calculation: We calculate the average log loss and accuracy during training and testing of our models and display the results in Table IV. For both models, lower training losses are calculated as compared to the test case. Here, Model 1 gives a smaller log loss than Model 2 resulting in increased average accuracy throughout training and testing the network.

4.5 Human Evaluation

To evaluate the quality of our synthesized sounds, we conduct a survey among local college students. In the survey, the students are presented a video with two audio samples, the original sound and the synthesised sound. We then ask students to select the option they prefer using four questions:

- Select the original (more realistic) sample.
- Select the most suitable sample (most reasonable audio initiated from the video clip).
- Select the sample with minimum noise.
- Select the most synchronized sample.

We assess the performance of our produced sound tracks for each of the categories. We evaluate both of our approaches 12 through a comparison task with the ground truth in the first query. From here, we like to gauge how realistic are our synthesized Foley tracks. To

evaluate this, we create a survey of a video option containing its original sound track and the same video in another option containing the generated sound tracks, for each of our AutoFoley categories. We observe when people make a wrong choice between the generated sound and the original one. In this survey 73.71% of the respondents chose the synthesised sound over the original sound with our first model, and 65.95% with the second model.

The remaining three qualitative questions are used to compare the performance between the two models. In these queries, we set the same two videos of all classes in two options associated with synthesized sound tracks from Models 1 and 2 respectively. We evaluate which method is preferred by respondents after observing the audio-video pairs. The survey results show that Model 2 (Frame Relation Network + ISTFT) outperforms Model 1 (Frame Sequence Network + ISTFT) for visual scenes associated with random action changes (i.e. breaking, cutting, footsteps, gunshot sound classes). For the rest of the categories, respondents chose the Model 1 generated sound tracks over Model 2. The detailed human evaluation results (selection percentages of survey queries) for each individual class are presented in Tables VIII and IX for Models 1 and 2 separately.

TABLE VIII

HUMAN EVALUATION RESULTS: SELECTION PERCENTAGE OF EACH SOUND CATEGORY FOR THE FIRST AND SECOND HUMAN SURVEY QUESTIONS

Class	Query 1		Query 2	
	Method 1	Method 2	Method 1	Method 2
Break	52.40%	56.10%	32.30%	67.70%
Car	71.53%	65.40%	55.76%	44.24%
Clock	90.91%	73.80%	70.90%	29.10%
Cutting	50.17%	45.35%	62.89%	37.11%
Fire	85.43%	75.40%	57.70%	42.30%
Footstep	61.72%	50.57%	72.13%	27.87%
Gunshot	68.33%	61.84%	64.60%	35.40%
Horse	89.38%	78.20%	53.80%	46.20%
Rain	88.62%	75.80%	50.00%	50.00%
Thunder	76.25%	72.17%	59.35%	40.65%
Typing	64.27%	62.39%	66.20%	33.80%
Waterfall	85.55%	74.40%	66.78%	33.22%
Average	**73.71%**	**65.95%**	**59.37%**	**40.63%**

Figure 4. Human Evaluation Results

5 Conclusion

In this seminar, we address a novel problem of adding Foley effect sound tracks to video clips of movies by using an efficient deep learning solution. Here, we have proposed two deep neural models, Frame Sequence and Frame Relation Network (associated with a less complex sound synthesis approach). These models are trained to predict the sound features from only visual inputs and artificially synthesize the required sound track. We have also introduced a nascent dataset for this particular task that contains audio-video pairs of the most popular Foley scene categories.

Our models show increased computational efficiency in learning the intricate transition relations and temporal dependencies of visual inputs from a low number of video frames. Though prior works do not directly match with our work, we conduct extensive qualitative, numerical and ablation analysis to demonstrate the usefulness of the proposed models and tools. We have achieved higher accuracy in sound retrieval experiment results (over 63%) for both prediction models than state-of-the-art researches, i.e. sound generation from visuals.

Lastly, our human survey result shows greater than 73% of respondents considered our generated sound as original. Since the task of adding automatic Foley to silent video with deep learning is an interesting and novel task, the next steps in this research are to expand on the training dataset, allowing for the generated sound output to more closely approximate the original sound. The computational efficiency of the proposed model will also be improved, with the goal of being able to process live video in real-time. The time synchronization problem will also be examined and optimized in future research.

6 References

[1] Z. Yuan, G. Ghinea, and G. Muntean, *"Beyond multimedia adaptation: Quality of experience-aware multi-sensorial media delivery,"* IEEE Transactions on Multimedia, vol. 17, no. 1, pp. 104–117, Jan 2015.

[2] J. Cortes. (2019*) The foley artist. [Online].* Available: https: //cogswell.edu/blog/the-foley-artist/

[3] A. Owens and A. A. Efros, *"Audio-visual scene analysis with selfsupervised multisensory features,"* in Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 631–648.

[4] N. Takahashi, M. Gygli, and L. Van Gool, *"Aenet: Learning deep audio features for video analysis,"* IEEE Transactions on Multimedia, vol. 20, no. 3, pp. 513–524, 2017. [8] Z. Liu, R. A. Yeh, X. Tang

[5] R. Arandjelovic and A. Zisserman, *"Look, listen and learn,"* in Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 609–617.

[6] K. Soomro, A. R. Zamir, and M. Shah, *"Ucf101: A dataset of 101 human actions classes from videos in the wild,"* arXiv preprint arXiv:1212.0402, 2012.

[7] W. W. Gaver, *"What in the world do we hear?: An ecological approach to auditory event perception,"* Ecological psychology, vol. 5, no. 1, pp. 1–29, 1993.

[8] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, *"Large-scale video classification with convolutional neural networks,"* in Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, 2014, pp. 1725–1732.

[9] W. W. Gaver, *"What in the world do we hear?: An ecological approach to auditory event perception,"* Ecological psychology, vol. 5, no. 1, pp. 1–29, 1993.

7 Appendix

Artificial Synthesis of Sound Tracks for Silent Videos

A guide by Swaraj Dash

Introduction

Adding sound effects in post production using the art of Foley has been an intricate part of movie and television soundtracks since the 1930s. The technique is named after Jack Foley, a sound editor at Universal Studios. Mr. Foley was the first to make sound effects for live radio broadcasts with the tools and items he had around him. Now almost every motion picture and television show contain Foley tracks.

1. Methodology

- **Feature Extraction**
  Highlight what's new, unusual, or surprising.
- **Class Prediction**
  Give people a reason to care.
- **Approach 1**
- **Approch 2**

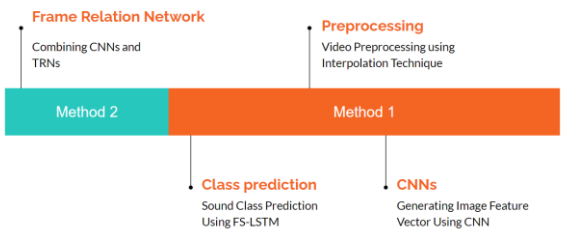
What is Feature Extraction?

Compute the features of all the audio files using spectrogram analysis; a visual way of representing the strength of a signal over time at different frequencies present

Note

Human hearing is based on a form of real-time spectrogram encoded by the cochlea of the inner ear, we convert the audio signal into a 2D representation (spectrogram) for extracting the audio feature

Building the Models



Evaluation