ACOUSTIC INSIGHTS INTO BIRD SOUNDS: UTILIZING DUAL ACOUSTIC CAMERAS FOR 3D POSITIONING ANALYSIS USING NEURAL NETWORKS

Sarah Shitrit
ID 207028614
Tel Aviv University
sarahshitrit@mail.tau.ac.il

Adi Green
ID 313472417
Tel Aviv University
adishafir@mail.tau.ac.il

April 24, 2025

ABSTRACT

In the quest to understand avian behavior through auditory signals, this study employs Dual Acoustic Cameras for 3D Positioning. Positioned at distinct angles, these cameras capture bird calls in a three-dimensional space. The acoustic data from two unique vantage points is used to pinpoint avian vocalization coordinates. A specially designed neural network, preceded by extensive efforts in data preprocessing, processes the acoustic footage, outputting 3D coordinates. Initial efforts with a CNN+LSTM architecture faced computational hurdles, leading to exploration of a more resource-friendly approach: a Convolutional Neural Network (CNN) coupled with fully connected layers. The results highlight the potential of acoustic cameras and neural networks in decoding the spatial aspect of bird calls, shedding light on avian acoustic behavior. This work sets a precedent for using machine learning in ecological and behavioral studies. Furthermore, experiments with data augmentation and dropout techniques underscore their role in enhancing data quantity and model robustness, optimizing avian behavior analysis through acoustic cameras and neural networks.

1 INTRODUCTION

This project is a collaboration between the Deep Learning course led by Raja Giryes and the Neuroecology Lab led by Professor Yossi Yovel and it's team.

Many animals use sound signals to coordinate movement. Understanding bird behavior and movement in three dimensions can provide valuable insights into their habitat utilization, migratory patterns, and nesting behaviors. Such knowledge can guide conservation efforts, especially for endangered or threatened species.

We aim to analyze footage from two acoustic cameras placed at different angles. These special cameras provide data on the source of sounds, in our case, bird calls. By comparing the recordings from the two distinct perspectives, we intend to determine the birds' three-dimensional positions. For our experiments, we used a stationary speaker that played bird noises in intervals. The videos which were recorded on the cameras were the inputs to the neural network we've designed. The network outputs the 3D location of the bird.

In our pursuit of an effective model for multi-class image labeling, initial memory constraints led to the transition from a CNN+LSTM architecture to a Video-to-Frame + Basic CNN + Fully Connected Layers approach. Despite facing stabilization challenges in the Large CNN + Fully Connected Layers experiment, the introduction of regularization techniques significantly improved overall performance. Augmenting the training dataset proved impactful, resulting in a noteworthy reduction in Mean Squared Error (MSE) through strategic dropout usage. Despite the insufficient amount of data and computational challenges, the results achieved through the CNN and Fully Connected Layers approach were deemed satisfying.

2 RELATED WORK

The use of audio-visual data to locate and track sound sources has been an active area of research in recent years. The fusion of audio and visual cues allows for a more robust and accurate localization compared to using either modality alone. Detection and tracking of drones using advanced acoustic cameras was performed by Joel Bassat et al. (2015) They presents a system to acoustically detect and track small moving objects, such as drones or ground robots, using acoustic cameras. The system is completely passive and composed of a 120-element microphone array and a video camera. [1] They used classic methods. In the field of neural networks there is Ephrat et al. (2018), who employed an audio-visual model for speech separation and demonstrated its ability to discern individual speakers even in noisy conditions.[2] Other notable works include that of Afouras et al. (2018), where they presented a deep learning method to combine lip reading with audio data to improve speech recognition performance[3]. While the majority of existing methods focus on a combination of audio-visual data for tasks like speech separation and recognition, the direct application for 3D localization remains an area of keen interest and ongoing exploration.

Our approach aligns closely with the methods commonly employed in the other studies mentioned before, that make use of neural networks. We initially explored the utilization of architectures such as LSTM and Convolutional Neural Networks, as they have been mentioned in relevant articles in the field. However, we encountered a significant challenge during our experimentation phase: limited computational resources. Architectures like Transformers and LSTM, which are often used in these articles, place high demands on computational power. As we progressed through our experiments, we came to the realization that such complex networks weren't essential for our specific task. In fact, we found that the temporal dimension, which these networks excel at modeling, played a less critical role in our context. Consequently, we were able to streamline our approach and work without the resource-intensive networks typically advantageous for handling sequential data.

3 DATA

3.1 Data Collection and Dataset Description

At the beginning of the project, data collection was carried out by us, with Marie providing guidance, within Professor Yossi's laboratory. The dataset comprises of 10-second-long videos captured from two distinct acoustic cameras, each filming an object from varying angles. While one camera remained stationary throughout the filming, the other camera intermittently adjusted its angle. Please refer to Figure 1 for a visual representation of the camera settings. From now on we will address camera 1 (the stationary one) as Ronen's camera and Camera 2 as the Lab's camera.

Figure 1: Camera Settings for Data Acquisition. The further camera on the right is Ronen's, that remained stationary during data collection.

We used a speaker that played a recording of bird sounds. In order to localize the speaker a white cross that was used as a ruler was drawn to the floor as shown in Figure 2. The middle of the cross was at a distance of 3 meters from Ronen's camera. Scale marks with a distance of 10 cm between them were drawn on the cross to cover a total distance of 3 meters from the center to each side. A wooden block of a height of 40 cm was also used in order the vary the measurements along the dimension that is perpendicular to the floor.

Figure 2: The ruler that was drawn to the floor.

The data consists of videos in .webm format (an example of a pre-processed frame is shown in Figure 3). A total of 140 videos were captured from each camera, resulting in a combined total of 280 videos. Each video has a frame rate of 25 frames per second. The acoustic camera changes the color of areas from which sounds come from according to the legend that can be seen in Figure 3.

The data acquisitions process was composed of 3 trials:

1. The cameras were positioned at 90 degrees angle between them. The speaker moved along the ruler between videos in steps of 20 cm. The speaker moved along the diagonals of the cross between videos in steps of 50 cm. The same process was repeated using a wooden block of height of 40 cm above ground, the steps were of 50 cm.
2. The cameras were positioned at 135 degrees angle between them. The speaker moved along the ruler between videos in steps of 50 cm. The speaker moved along the diagonals of the cross between videos in steps of 50 cm.
3. The cameras were positioned at 45 degrees angle between them. The speaker moved along the ruler between videos in steps of 50 cm. The speaker moved along the diagonals of the cross between videos in steps of 50 cm.

An illustration is shown in Figure 3.

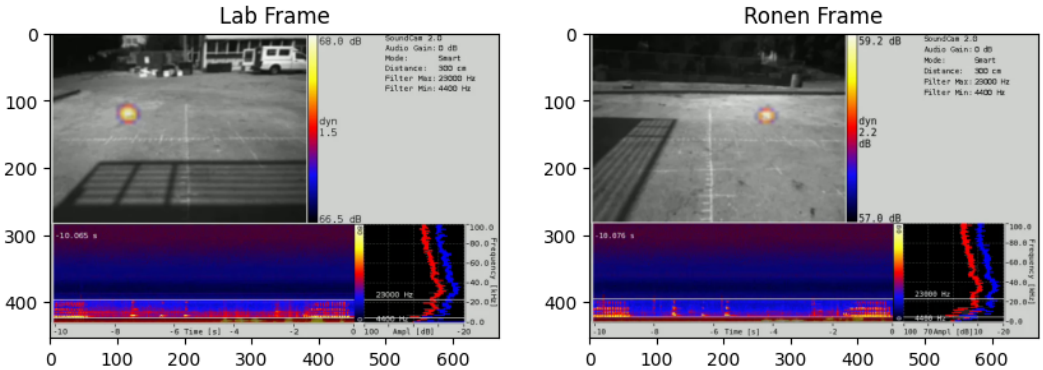


Figure 3: Examples of pre-processed frames from the two cameras. One frame is from camera 1 (Lab), and the other is from camera 2 (Ronen).

Figure 4: Left to right: **Trial A:** 90 degrees between the cameras. **Trial B:** 135 degrees between the cameras. **Trial C:** 45 degrees between the cameras.

3.2 Pre-Processing

In order to synchronize and enhance the quality of the collected videos, a series of preprocessing steps were carried out. Matching videos, captured from the two acoustic cameras at varying angles, underwent a synchronization process. Initially, audio was extracted from each video and converted to mono, ensuring consistency. A bandpass filter was then applied to isolate specific bird noises, focusing on the frequency range between 1000 Hz and 2500 Hz. This range was chosen by trial and error to get the best synchronization. Cross-correlation analysis was employed to determine any temporal offset between the two audio streams. Depending on the offset, the videos were trimmed to align their temporal frames accurately. If a temporal shift was identified, the video with the lag was adjusted to match the timing of the other video.

After the videos were synchronized, they underwent a sequence of preprocessing steps within the framework of the VideoDataset class. These steps included loading and cropping the videos to dimensions of 285x385 pixels, ensuring that only relevant content was presented while excluding spectograms and other irrelevant information, as illustrated in Figure 4. The frames were then converted from BGR to RGB color space, followed by by replacing each video with one frame which is the average of all frames in the video. This average representation served to reduce data volume within a batch, contributing to more efficient processing. The idea behind it was that the noise is random and unrelated to the label, thus averaging will make areas that are marked with color due to sound of birds and not due to random sounds more prominent by having higher color values. An example for final results from these steps can be found in Figure 5 and a general overview of the process can be visualized in Figure 6.

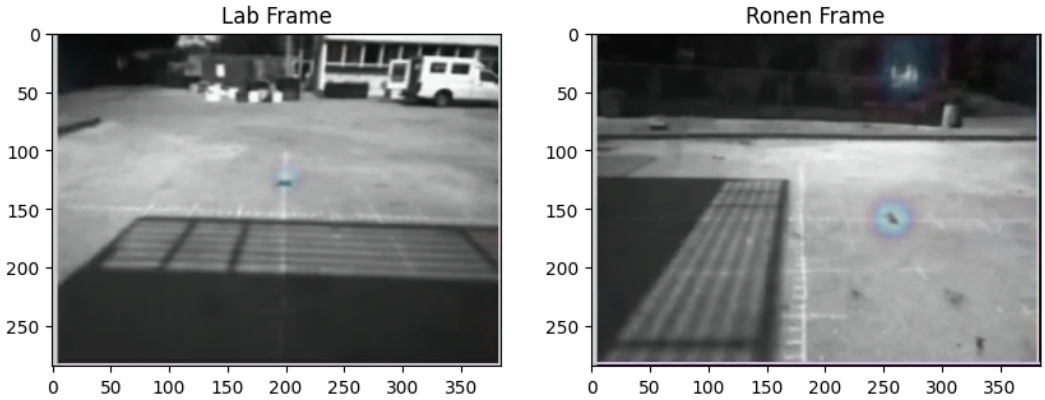


Figure 5: Examples of processed and averaged frames from the two cameras.

In conclusion, through these preprocessing actions, the videos were aligned, relevant content was extracted, and a concise average visual summary was derived, setting the stage for effective utilization of the processed videos in subsequent machine learning tasks.

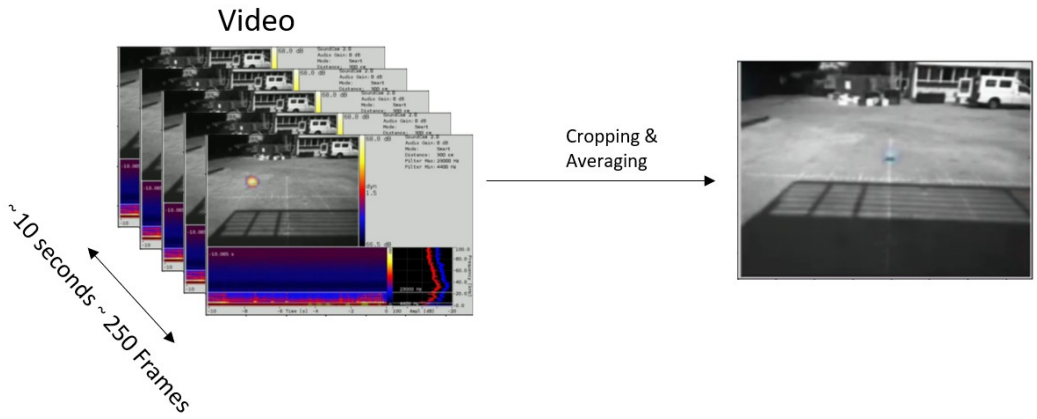


Figure 6: The preprocessing steps included video synchronization between 2 videos using audio. Following this, the frames from each video were cropped, converted to RGB, and averaged to reduce data volume. This figure depicts a general overview of the process from video to a single frame.

3.3 Labels

The labels include the (x, y, z) representation of the sound source in meters measured from the center of the cross. The directions are specified in Figure 7. These labels were given according to the camera that remained fixed in place (Ronen's). The distribution of these labels can be observed in Figure 8. For training and testing, a 80-20 train-test split was applied, as depicted in the figure. The test distribution mirrors the train distribution due to the limited amount of available data.

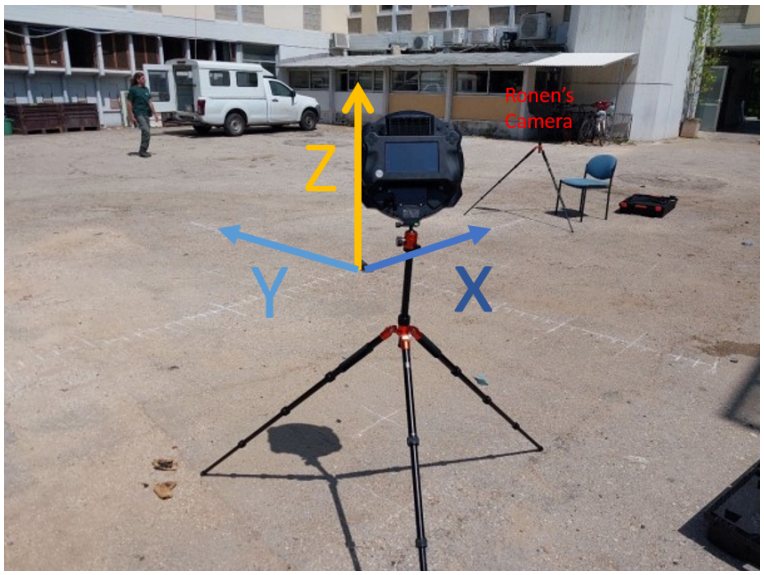


Figure 7: Direction of axes in relation to Ronen's camera.

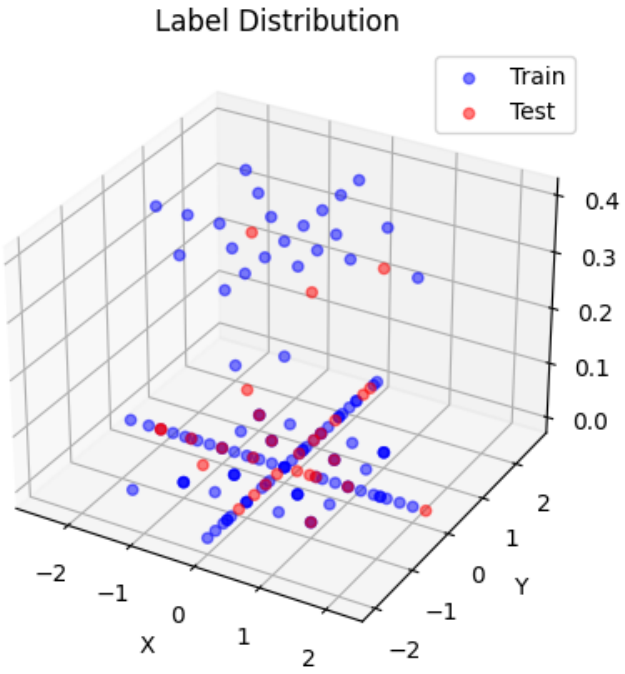


Figure 8: Labels Distribution. The test distribution mirrors the train distribution due to the limited amount of available data.

4 METHODS

4.1 Convolutional Neural Networks (CNN)

Convolutional neural networks (CNN) are widely used for image labeling tasks and also for depth estimation [4]. Advancements in deep learning have witnessed a surge in the use of architectures with multiple hidden layers, outpacing the performance of traditional methods across various domains, particularly in pattern recognition. Among these architectures, Convolutional Neural Networks (CNNs) stand out. A CNN comprises several layers, which include the convolutional layer, non-linearity (or activation) layer, pooling layer, and fully-connected layer. Notably, only the convolutional and fully-connected layers contain trainable parameters, while the pooling and non-linearity layers do not. CNNs have demonstrated remarkable efficacy in a range of machine learning tasks, especially those involving image data . This is evident in their success with major image classification challenges like ImageNet, as well as their applications in computer vision and even natural language processing (NLP), where the results have been nothing short of astounding [5].

In the context of our research, we are confronted with a challenging task of regression aimed at precisely estimating the spatial coordinates of sound-emitting objects. This task essentially transforms into an image classification problem because videos are essentially sequences of individual images, each capturing a distinct moment in time. What sets our challenge apart is the potential infinity of labels we must deal with. This infinite possibility arises from the continuous nature of the spatial coordinates we aim to determine, originating from the center of a reference point and allowing for an infinite array of possible positions within a three-dimensional space.

CNNs are particularly well-suited to our scenario, not only for their image analysis capabilities but also for their adaptability. We remain conscious of our resource constraints, especially concerning computational power and the volume of data at our disposal. CNNs offer the flexibility to employ relatively less complex network architectures, thereby achieving a balanced compromise between computational feasibility and the pursuit of high-precision results. This pragmatic approach empowers us to efficiently tackle our distinct and computationally intensive task of predicting the continuous three-dimensional locations of sound-emitting objects within our video data.

4.2 Long Short-Term Memory (LSTM)

CNNs are extensively applied for image classification problems but they scarcly captures the characterization of time sequence. RNNs are capable of processing sequential information such as in speech recognition [6] and video [7]. Standing for Long Short-Term Memory, LSTM represents an advanced architecture within the realm of Recurrent Neural Networks (RNNs). It was specifically developed to counteract the challenges of vanishing and exploding gradients that traditional RNNs often encounter. Distinct from feedforward neural networks, RNNs possess cyclic connections, rendering them uniquely adept at handling sequential data. Their prowess has been showcased in tasks like sequence labeling and prediction, evident in applications such as handwriting recognition, language modeling, and acoustic frame phonetic labeling.

For each element in the input sequence, each layer computes the following function:

$\mathrm{i}_t = \sigma\left(W_{ii}x_t + b_{ii} + W_{hi}h_{t-1} + b_{hi}\right)$
$f_t = \sigma\left(W_{if}x_t + b_{if} + W_{hf}h_{t-1} + b_{hf}\right)$
$g_t = \tanh\left(W_{ig}x_t + b_{ig} + W_{hg}h_{t-1} + b_{hg}\right)$
$o_t = \sigma\left(W_{io}x_t + b_{io} + W_{ho}h_{t-1} + b_{ho}\right)$
$c_t = f_t \odot c_{t-1} + i_t \odot g_t$
$h_t = o_t \odot \tanh\left(c_t\right)$ According to the implementation used by pytorch for LSTM [6]. An illustration is shown in Figure 9.

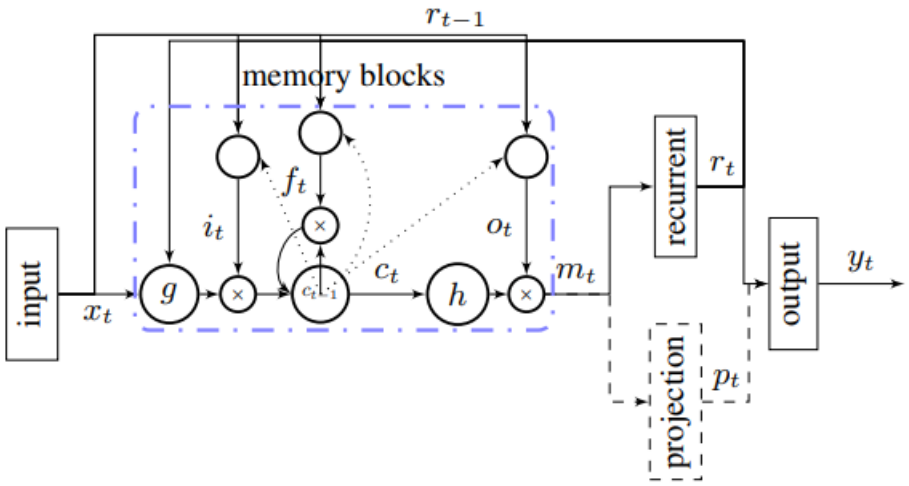


Figure 9: A diagram demonstrating the structure of a LSTM as in the pytorch's implementation we've used.

In the context of video data, where each frame serves as a temporal precursor to the next, LSTM's ability to retain and selectively update information over extended sequences becomes paramount. This makes LSTM networks valuable for tasks such as video activity recognition and predicting motion patterns, as they excel in discerning patterns and dependencies across successive frames, providing a robust framework for comprehensive video analysis.

4.3 Combination of CNN and LSTM

All kinds of attempts have been made to combine the two methods, CNN and LSTM, especially in the context of video. However, most of these attempts focused on methods for classifying or describing the video rather than regression[8][9]. We hypothesize that the optimal results can be achieved by combining both methods, leveraging the time series features of the video and the sophisticated feature extraction capabilities of the CNN.

5 EXPERIMENTS

5.1 CNN + LSTM

First we tried to use a CNN+LSTM architecture as we found evidence in the literature that it might be a good method for regression of 3 dimensional coordinates as discussed under the "methods" section. We later on encountered technical difficulties as the model required massive memory which we were not able to provide. Therefore we had to use less sophisticated models and find a way to decrease the size of our input data.

At this experiment we used a pre-trained ResNet101 for feature extraction. ResNet101 is a variant of the ResNet (Residual Network) architecture, specifically designed with 101 layers. Introduced by Microsoft Research in 2015, ResNet addresses the vanishing gradient problem in deep neural networks by introducing "skip connections" or "residual connections". These connections allow activation to bypass one or more layers, effectively enabling the network to learn identity functions and ensuring that adding more layers does not harm performance. The primary advantage of ResNet101, and ResNets in general, is their ability to train very deep networks without significant degradation in accuracy. The depth of the network allows for the extraction of intricate features from input data, making ResNet101 particularly effective for complex tasks like image classification. The output of the ResNet was a feature vector of length 300.

The two 300 length feature vectors, one extracted from each video, were the input for the LSTM layers. The LSTM network has an input of 600, hidden size of 256 and 3 layers. The idea behind using an LSTM was explained in the "methods" section. Finally, the output of the LSTM was passed through 2 fully connected layers to produce the final regression result.

Two videos after preprocessing, one from each camera, were the inputs to the net (without the averaging part but ensuring we get the same amount of frames for each video). Each frame was treated separately for the feature extraction in the ResNet.

We conducted a series of experiments to overcome our resource limitations. These experiments involved iteratively adjusting the architecture by reducing the number of layers in the ResNet, varying batch sizes, and even considering gray-scale images instead of RGB. Despite these attempts, we continued to face challenges in running the architecture, leaving us uncertain about its potential advantages under our current constraints. However, it's worth noting that we believe this architecture holds great potential due to its ability to incorporate sequential video information alongside the feature extraction capabilities of the CNN.

5.2 Video to frame + Basic CNN + Fully connected layers

To optimize memory usage, we implemented a modified CNN architecture that omitted the LSTM at the end of the CNN and employed a simplified CNN structure with just one convolution layer and one pooling layer, as illustrated in Figure 10. At the conclusion of this CNN, we introduced a linear fully connected layer, responsible for outputting three continuous values representing X, Y, and Z coordinates. During this stage, we also decided to create a single averaged frame from each video as the input frame, primarily to alleviate memory requirements.

The rationale behind averaging the frames was driven by the understanding that most of the noise within the video data is stochastic and time-dependent, while the desired signal consistently corresponds to the same fixed 3D position within the scene. By employing frame averaging, we sought to attenuate noise-related components and enhance the signal stemming from the speaker, thus improving the robustness of our model.

It's important to note that we did not document final results for this preliminary network iteration, as it primarily served as a proof of concept to assess its functionality. Subsequently, we conducted more extensive experiments using a more sophisticated CNN architecture. Although the model from this initial phase may not be available for analysis, our subsequent work involved a comprehensive and well-documented investigation.

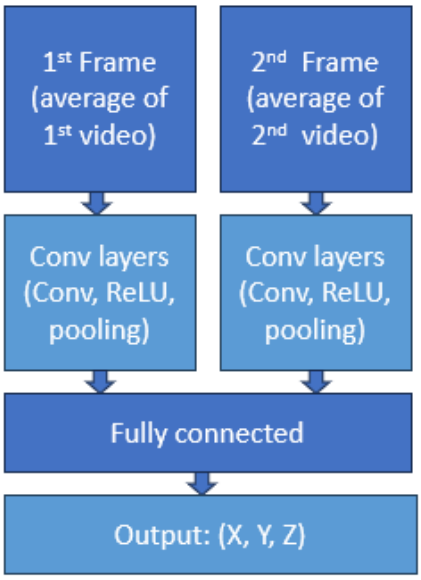


Figure 10: Architecture of a CNN with fully connected layers.

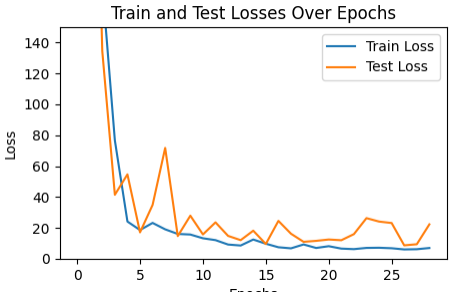
5.3 Large CNN + Fully connected layers

In our pursuit of improving model performance, we decided to enhance the depth of the CNN and additional blocks, resulting in a Siamese architecture comprised of three blocks total. Each block consists of a convolution layer, batch normalization, and dropout layer. Our input data consisted of two frames, each sized at 285x385 pixels with three channels. During the forward pass, each frame was independently processed through a block, incorporating the ReLU activation function and max-pooling with a kernel size of 2 and a stride of 2. Following the completion of the third block, the data from both frames was concatenated and passed through two fully connected layers, with dropout applied between the layers. The mean squared error (MSE) loss function was employed, with the network outputting a vector of size 3, representing the (x, y, z) coordinates.

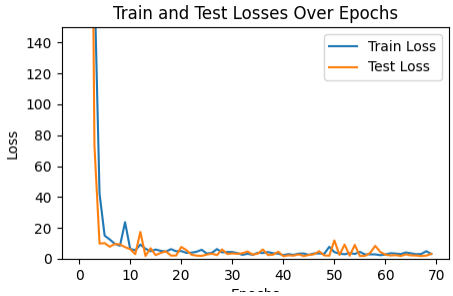
In our experimental setup, we implemented an early stopping mechanism to halt training when no improvement on the validation data was observed after a pre-defined number of epochs. This mechanism played a pivotal role in preventing overfitting, a critical concern due to the limited number of available samples. Overfitting, which can occur when a model becomes overly tailored to the training data, may compromise the model's generalization to unseen data.

In these experiments, we utilized two matching frames as input, each representing the average of the entire video. In the absence of any regularization, our model yielded an MSE loss of 6.112 for the training data and 8.752 for the test data. Recognizing the importance of regularization, we incorporated two techniques we learned about in our studies: dropout with a rate of 0.05 and Weight Decay (WD) with a value of 10^-4. Regularization plays a vital role in preventing the network from fitting noise in the training data and encourages it to focus on learning meaningful patterns.

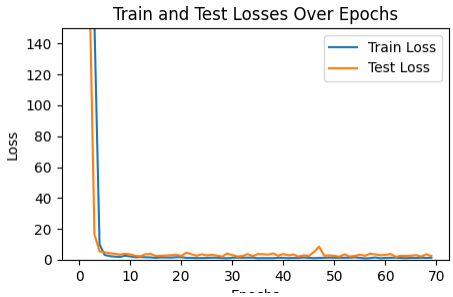
Examining the results in Figure 11, it becomes evident that introducing regularization helps with stabilization, as reflected in the training versus test data plots. Looking at the outcomes presented in Figure 12, we observe a noteworthy overall improvement in performance compared to the absence of regularization. With dropout, our model achieved an MSE loss of 1.7739 for the test data, and with weight decay, it achieved an MSE loss of 2.0421 for the test data. This underscores the value of regularization as a proactive measure to enhance the network's reliability and generalizability.



(a) No regularization



(b) With dropout=0.05



(c) With Weight Decay 10^{-4}

Figure 11: This figure shows how training progressed in three different experiments using loss-versus-epoch graphs. We've set fixed y-axis limits (ylim) for easier viewing.

Model	Data	Best Loss for test (MSE)	Loss for train (MSE)	epoch
CNN, no dropout, no regularization	Averaged to one frame per video	8.7519	6.1118	32
CNN with dropout	Averaged to one frame per video	1.7739	2.4556	41
CNN with Weight decay	Averaged to one frame per video	2.0421	1.4294	63

Figure 12: RMSE for models trained on data without augmentations.

5.3.1 Train on data-set with augmentation

To further improve the reduction of train Mean Squared Error (MSE), we adopted an augmentation strategy aimed at expanding the training dataset and thereby enhancing the model's ability to generalize. In this augmentation phase, we divided each video in the training section into five segments, averaging each segment and treating it as an individual video with the same label as the original video. This strategy effectively increased the size of our training dataset from 112 samples to 568. An ilustraition can be seen in Figure 13. It is important to note that averaging to 5 frames was not chosen arbitrarily, less than that didn't seem to improve the model error but more seemed to harm it.

Furthermore, based on our previous observations indicating the effectiveness of dropout regularization, we continued to leverage dropout in this section, but with two different dropout rates: 0.1 and 0.05. The outcomes of this are evident in the results presented in the table in Figure 14, illustrating noticeable improvements. The MSE loss for the test data was superior for a dropout rate of 0.1, resulting in an MSE of 0.5924. This augmented dataset and strategic use of dropout not only boosted the model's performance but also contributed to a more robust and generalizable solution.

It's important to note that we did not include loss-versus-epoch graphs in our presentation due to the limited number of epochs (computation limitations of the google cloud platform). While the graphs may not visually convey the full training process, it's essential to highlight that the model did converge to a significantly improved loss. The decision to not include the graphs was influenced by the relatively short training duration, which did not allow for a comprehensive display of epoch-wise changes. Despite this limitation, the achieved convergence to a better loss underscores the effectiveness of our training strategies.

Furthermore, it's crucial to emphasize that our testing process was conducted on complete, unaltered videos. Notably, data augmentation was intentionally excluded from the test dataset during evaluation, ensuring that the obtained results, averaged for accuracy, maintain the integrity and impartiality of the evaluation process. This deliberate approach guarantees that the results accurately reflect real-world scenarios.

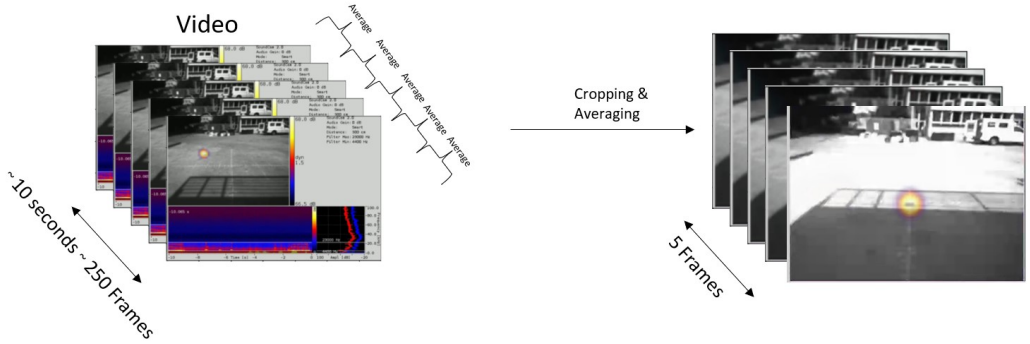


Figure 13: This figure depicts a general overview of the process from video to a multiple frames to achieve data augmentation

Model	Data	Best Loss for test (MSE)	Loss for train (MSE)	epoch
CNN with dropout of 0.05	Averaged to 5 frames per video	0.6079	0.5453	21
CNN with dropout of 0.1	Averaged to 5 frames per video	0.5924	0.5535	25

Figure 14: RMSE for models trained on data with augmentations.

6 Conclusions

In summary, our study highlights the effectiveness of combining Convolutional Neural Networks (CNN) with fully connected layers to determine the spatial locations of sound-emitting objects, such as birds. We found that deeper CNN architectures, when applied to single frames created by averaging videos, consistently produce improved results.

The introduction of regularization techniques further enhances performance. Augmentation, in conjunction with dropout, is an effective strategy for improving results, emphasizing the importance of data quantity.

To achieve even better results, expanding the dataset by collecting more data and exploring additional augmentation techniques is recommended. Furthermore, future experiments may explore the combination of CNN with Long Short-Term Memory (LSTM) networks for further improvements.

Our research has significant implications as it enables a more precise understanding of bird behavior and habitat utilization, with the potential to contribute to conservation efforts and a deeper comprehension of avian interactions with their surroundings.

References

[1] Joël Busset et al. "Detection and tracking of drones using advanced acoustic cameras". In: *SPIE* (2015). DOI: https://www.dora.lib4ri.ch/empa/islandora/object/empa%3A11762/datastream/PDF/Busset-2015-Detection_and_tracking_of_drones-%28published_version%29.pdf.

[2] Ariel Ephrat et al. "Looking to Listen at the Cocktail Party: A Speaker-Independent Audio-Visual Model for Speech Separation". In: *ACM Trans. Graph.* 37.4 (July 2018). ISSN: 0730-0301. DOI: 10.1145/3197517.3201357. URL: https://doi.org/10.1145/3197517.3201357.

[3] Triantafyllos Afouras et al. "Deep Audio-Visual Speech Recognition". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44.12 (2022), pp. 8717–8727. DOI: 10.1109/TPAMI.2018.2889052.

[4] Hamid Laga et al. "A Survey on Deep Learning Techniques for Stereo-Based Depth Estimation". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44.4 (Apr. 2022), pp. 1738–1764. DOI: 10.1109/tpami.2020.3032602. URL: https://doi.org/10.1109%2Ftpami.2020.3032602.

[5] Geoffrey E. Hinton Alex Krizhevsky Ilya Sutskever. "ImageNet Classification with Deep Convolutional Neural Networks". In: *Advances in neural information processing systems* (2012). DOI: https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.

[6] Françoise Beaufays Haşim Sak Andrew Senior. "Long Short-Term Memory Based Recurrent Neural Network Architectures for Large Vocabulary Speech Recognition ". In: *Neural and Evolutionary Computing* (). DOI: https://doi.org/10.48550/arXiv.1402.1128.

[7] Tianqi Zhao. "Deep Multimodal Learning: An Effective Method for Video Classification". In: *2019 IEEE International Conference on Web Services (ICWS)*. 2019, pp. 398–402. DOI: 10.1109/ICWS.2019.00071.

[8] Muhammad Abdullah, Mobeen Ahmad, and Dongil Han. "Facial Expression Recognition in Videos: An CNN-LSTM based Model for Video Classification". In: *2020 International Conference on Electronics, Information, and Communication (ICEIC)*. 2020, pp. 1–3. DOI: 10.1109/ICEIC49074.2020.9051332.

[9] Jeffrey Donahue et al. "Long-term recurrent convolutional networks for visual recognition and description". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 2625–2634.

7 Appendix

The code of our project can be found in this link: https://github.com/adigreen26/Deep-Learning-Acoustic-Cameras-2023.git.