Understanding Dog Behavior through Visual and Aural Sensing Using Deep Learning

Amy Lin
Princeton High School

1. Introduction

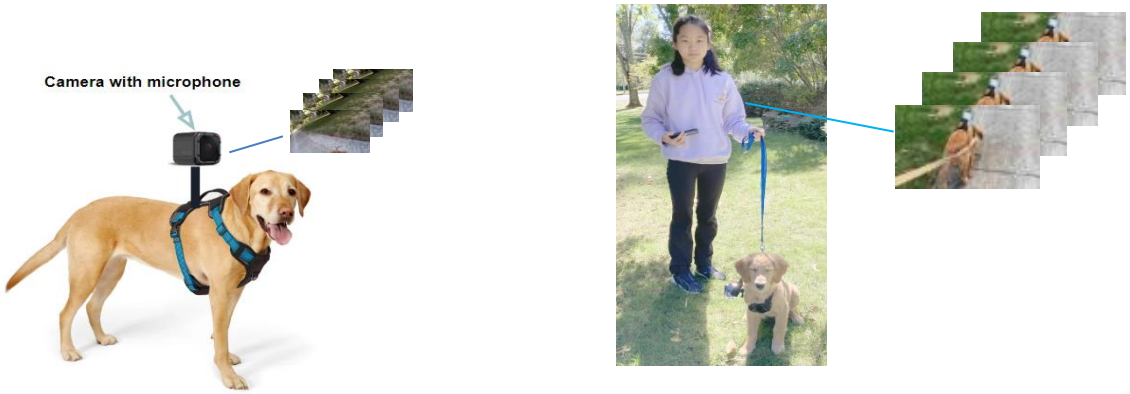
There have been many developments in artificial intelligence (AI) technologies in the past decade that have revolutionized the way people live and shaped society. However, AI relies on large amounts of data and can make mistakes that seem trivial to humans. For example, self-driving cars can misrecognize obstacles on the road which can lead to fatal accidents. In contrast, a dog knows to stay away from realistic street art of a ditch even though it has never seen it before. A better understanding of natural intelligence is critical to advancing the next generation of AI. There have been efforts to study animal intelligence as the first step to understand natural intelligence, which is a complex and challenging problem.

In this project, we are interested in studying how a dog behaves and reacts to different stimuli in its surroundings. Understanding dog behavior could provide useful insights in understanding animal intelligence as well as human intelligence. Dogs have five senses: vision, hearing, smell, touch, and taste. Compared to a human, their sense of hearing is 4 times as powerful and their sense of smell is about 100,000 times as powerful [5]. Dogs also have a wider angle of vision than humans, but cannot always see objects in focus. Additionally, the average dog has the intelligence of roughly a 2.5 year old human baby, but is even more adept at reading and understanding people than chimpanzees and human babies [6]. Studies have also shown the brightest dogs seem capable of learning hundreds of words. Dogs are capable of expressing emotions such as happiness, anger, fear, and jealousy, and are also capable of experiencing depression and anxiety, just like humans [3].

Existing studies on modeling dog behavior and reaction to visual stimuli have been reported for the purpose of developing robotic canine companions. A recent study by Ehsani et al. used a deep learning model to predict a dog's behavior using only visual information perceived by the dog. Their model was able to predict the dog's reaction and estimate walkable surfaces [1]. A study by Marks et al. on understanding animal behavior showed that animals are capable of displaying behaviors linked to stress, fear, curiosity, stress, anxiety, discomfort, and more, and subtle changes in behavior can be demonstrated through the animals' interactions with others [3]. Gregory Berns et al. studied dogs' brain activity in response to human hand signals using fMRI scans [4]. The study showed that dogs tend to pay close attention to human signals and display brain activity only when they saw familiar hand signals for rewards. Other work performed by researchers used cameras and wearable devices to monitor a dog's behavior [12, 13].

Previous work has shown interesting results. Most of them rely on visual information. In this work, we plan to study how a dog reacts multi-modality stimuli including visual and auditory information. By collecting video and audio data from a dog's perspective, we create a database of egocentric visual and audio stimuli that represents what a dog sees and hears. We use a deep learning approach and propose an extended Convolutional Neural Network (eCNN) model to learn the association between the dog's reaction and the visual and audio stimuli perceived by the dog.

This work can be applied to create effective ways of training dogs for various services. Currently, dog training is done from a human trainer's perspective. However, it would be more effective to train dogs from their own perspective. With the known association between a dog's surroundings and its reactions, we can design environments customized to the dog's reactions and perform interventions according to their natural behavior. Finally, as dogs display many human-like qualities, we hope to gain insights into human intelligence by understanding the behavior and intelligence of dogs.



(a) Camera with microphone mounted on dog harness

(b) Hand-held camera captures videos of dog movements and surroundings

Figure 1. Data collection (Left: dog view; Right: human view)

2. Methodology

We conducted this research by analyzing visual and auditory information from the environment perceived by a dog.

2.1 Data Collection

Data collection focuses on collecting the visual and audio stimuli in the dog's environment. A family pet, which is a Golden Retriever, was brought to different environments including parks and neighborhood streets. As Figure 1 shows, a GoPro camera is attached to the harness that the dog wears to capture what the dog sees and hears (Fig. 1(a)). The video stream captured by this GoPro camera is referred to as the dog-view video. At the same time, a hand-held camera is used by a person to record

the dog's movements and its surroundings. The video stream captured by the hand-held camera is referred to as the human-view video. When recording the videos, both cameras were turned on at roughly the same time. An audible start signal, e.g. "take one", was used to synchronize the recordings made by the two cameras. A total of 22 pairs of videos were collected.

2.2 Data Preparation

Three types of data, image frames, audio signals, and dog actions, were extracted from the dog-view and human-view videos. Image information was extracted from the dog-view videos. When the dog was walking or running, the dog-view camera often recorded extra noise from the leash. Therefore, we used audio signals extracted from the hand-held camera for its better audio quality to replace audio signals from the dog-view camera. Dog actions were manually labeled using the images from the human-view videos.

(a) Dog view image

(b) Human view image

Figure 2. Audio signals, dog view images, and human view images are aligned by time.

To establish the correspondence between the dog's actions and its visual and audio stimuli, we established the time correspondence between the image frames from the dog's view and the image frames from the human's view. Assume FPS_D and FPS_H are the frame rates (i.e. frames per second) of the dog-view camera and the human-view camera respectively. Assume T_{D0} and T_{H0} are the *start times* when the start signals of the dog-view camera and the human-view camera respectively. Given the dog-view image frame fr_D, the corresponding human-view image frame fr_H is

$$fr_H = \left(\frac{fr_D}{FPS_D} - T_{D0} + T_{H0}\right) \cdot FPS_H \quad (1)$$

Once the correspondence between the dog-view and human-view image frames are established, the correspondence between the dog's actions and the visual and audio stimuli is known.

As Figure 3 shows, there are 4 types of actions defined: *Sit, Stand, Walk,* and *Smell*. When the dog starts an action, the dog tends to continue the action for a period of time. Therefore, the image frames were only labeled when the dog changed its actions, and the same action is assigned to the subsequent image frames until a new action is presented.

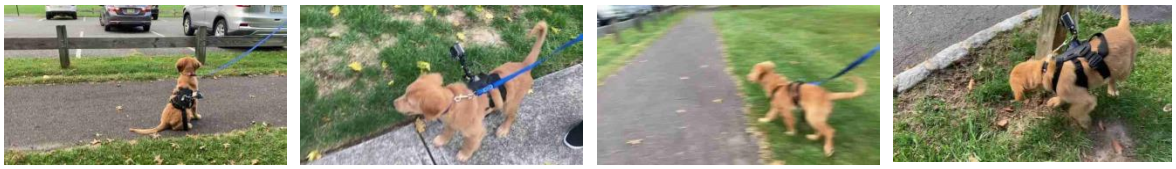


Figure 3. Images showing 4 actions of the dog: *Sit, Stand, Walk,* and *Smell*.

2.3 Data Analysis

Given the visual and audio stimuli a dog senses and its corresponding actions, we want to understand how the dog reacts to what it sees and hears. For example, what makes a sitting dog start walking? We propose an extended Convolutional Neural Network (eCNN) to learn the association between visual and audio stimuli and the corresponding dog actions. Compared to the CNN model, which only takes images as input variables, the eCNN model is able to explore data from multiple modalities, including images, motion, and audio. Our problem is formulated as a multi-class classification problem, where we use image, audio, and motion information to classify the dog's action into one of 4 classes: *Sit, Stand, Walk,* and *Smell*, shown in Figure 3.

2.3.1 Input Features

Image frames of size 672x378 pixels were extracted from the dog-view videos to capture what the dog sees. Each image is composed of three color channels: red, green, and blue. For faster computation, all image frames were resized to 151x85 pixels without losing useful visual information. The frame rate of the dog-view camera is 30.05 FPS and consecutive image frames are highly similar. Therefore, we performed down sampling over time and selected one in every 5 frames to include in the dataset.

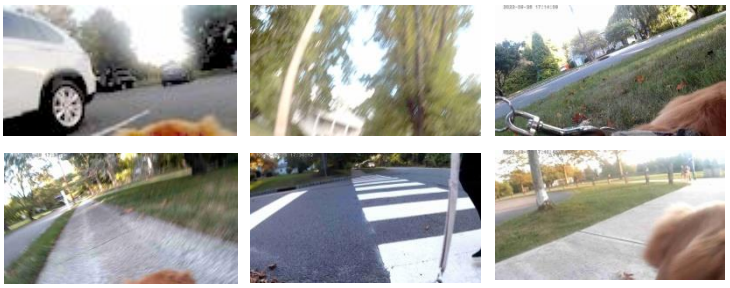


Figure 4. Sample images captured by the dog-view camera in the dataset.

The audio signals extracted from the human-view videos capture what the dog hears. We want to analyze the frequency content of the audio signal. Short-Time Fourier Transform (STFT) was used to decompose the audio signal into individual frequency components. STFT is defined as

$$S(m,\omega) = \sum_{n=-\infty}^{\infty} x(n)w(n-mH)e^{-i\omega n} \quad (2)$$

where $x(n)$ represents the audio signal at time n, m is the index of the moving window, H is the hop length, $w(n)$ is the windowing function, and ω is the frequency. Since the STFT $S(m,\omega)$ is a complex function, we take the magnitude $|S(m,\omega)|$ as the input feature to the eCNN model. Figure 4 shows an example spectrogram of an audio signal produced by STFT.

In our experiment, we used the hop length of 1/90th second and a window size of 1 second. Figure 5 shows the resulting spectrogram (right) of a recorded audio signal (left) produced by STFT.

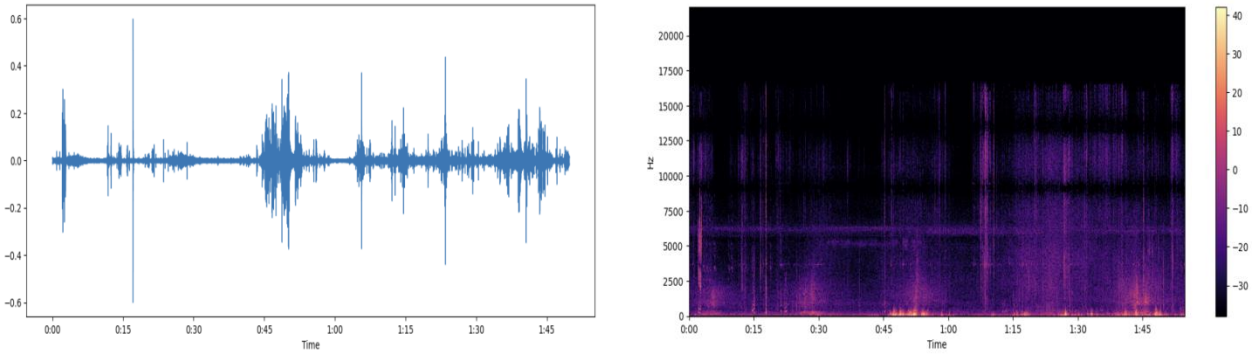


Figure 5. Spectrogram of audio signal produced by STFT. (Left: audio signal; Right: spectrogram)

To capture the sequential nature of the visual data, we included motion information as the third type of input feature. To compute image motion, we use the template matching method [11] to estimate the motion vectors of image blocks, i.e. how an image block moves from one frame to the next. Define I_t as the image frame at time t. As Figure 6 shows, for an image block centered around pixel (x_0, y_0), template matching finds the new block location $(x_0 + mx, y_0 + my)$ in the next image frame I_{t+1} such that the difference between the two image blocks measured by sum of squared error is minimized. A motion vector $mv(x_0, y_0) = (dx, dy)$ is found as

$$(dx,dy) = argmin_{\substack{kx=\{-D,\dots D\}\\ ky=(-D,\dots D\}}} \sum_{i,j} I_{t+1}(x_0+i+kx, y_0+j+ky) - I_t(x_0+i, y_0+j) \quad (3)$$

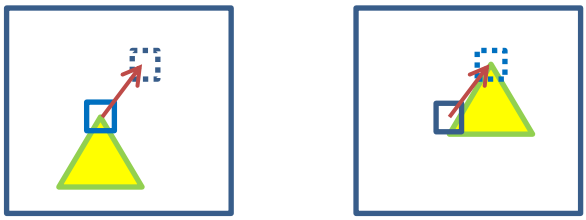


Figure 6. Motion vector of an image block.

Figure 7. Image pyramid with 3 levels.

To make the method more computationally efficient, we first constructed image pyramids, sequences of resized images at different resolutions $\{I_t^l: l = 0, 1, \dots, L-1\}$ (Figure 7) with a scaling factor 2. Template matching is first performed on the lowest resolution image. Assume at level l a motion vector (dx^l, dy^l) is found for an image block centered around pixel (x_0, y_0). Then the motion vector at level $l-1$, (dx^{l-1}, dy^{l-1}), at pixel location $(2x_0, 2y_0)$ is found as

$$\left(dx^{l-1}, dy^{l-1}\right)$$

$$= argmin_{\substack{kx=\{2dx^l-D,\dots,2dx^l+D\} \\ ky=\{2dy^l-D,\dots,2dy^l+D\}}} \sum_{i,j} I_{t+1}(2x_0 + i + kx, 2y_0 + j + ky) - I_t(2x_0 + i, 2y_0 + j) \qquad (4)$$

In general, image motion can be caused by both camera movements and objects moving in the scene. Since we are interested in object motion, we want to remove the motion caused by camera movements. We use the global average of motion vectors to represent the motion caused by camera movements and subtract the global motion from the motion vectors.

$$\overline{mv}(x, y) = \left(dx(x, y) - dx_g, dx(x, y) - dy_g\right)$$

$$dx_g = \frac{1}{N}\sum_{x,y} dx(x,y), \quad dy_g = \frac{1}{N}\sum_{x,y} dy(x,y) \tag{5}$$

In our experiment, we constructed image pyramids with 4 levels. Image blocks of 5x5 pixels were used in template matching. For computational efficiency, centers of image blocks are 32 pixels apart in the original resolution. A resulting motion field is shown in Figure 8.

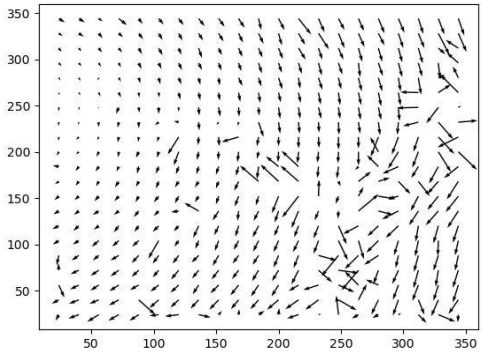


Figure 8. Motion field.

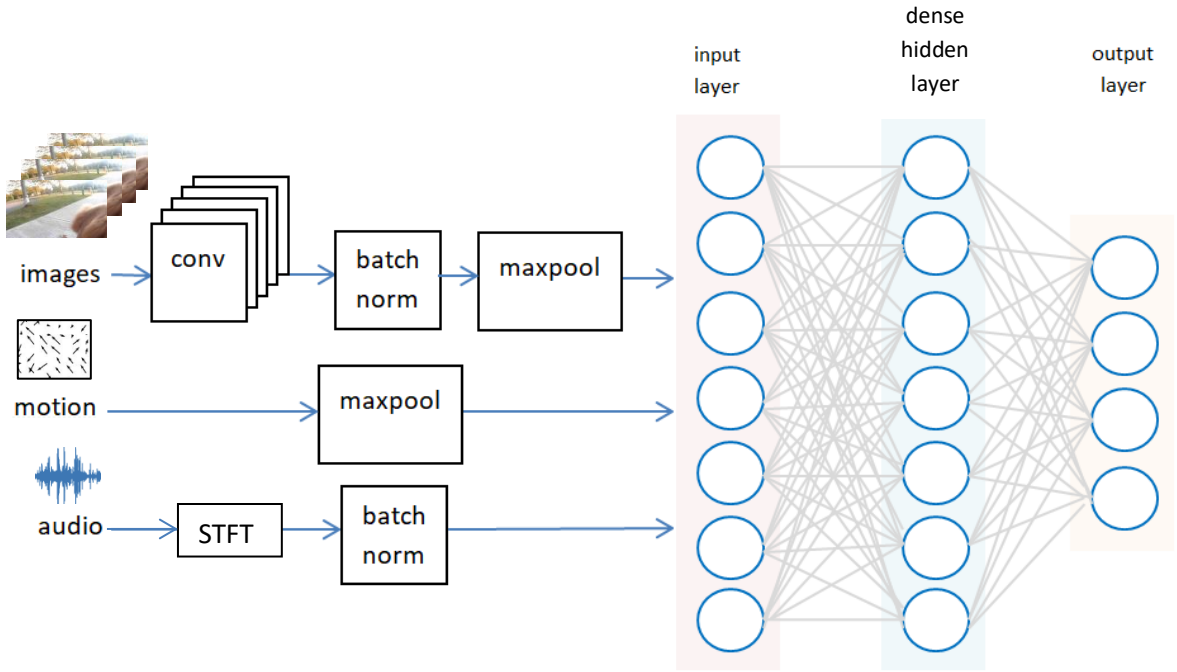


Figure 9. Extended Convolutional Neural Network (eCNN) model.

2.3.2 Extended Convolutional Neural Network (eCNN)

To utilize images, motion, and audio, we propose an extended CNN (eCNN) model to take multi-modal data as input, as shown in Figure 9. In the eCNN model, the first input type is an image. The image goes through a convolution layer, which is composed of 32 filters with kernel size 7x7 and stride 1. Batch normalization is then applied to the convolved image over the color channels, and a max pooling operation with a pool size of 3x3 is performed. The resulting output is flattened and fed into the input layer of the eCNN model. The second input type is a motion field. The magnitudes of the motion vectors go through a max pooling layer with a pool size of 9x3 before they are fed into the input layer. The third input type is an audio signal. First we apply STFT to the audio signal to get a vector of the frequency domain representation. Batch normalization is performed on its magnitude spectrum and the resulting output is fed into the input layer. The input layer feeds the three types of input data into a dense hidden layer with 30 nodes and a sigmoid activation function. The dense layer is connected to an output layer of 4 nodes with a softmax activation function, corresponding to the one-hot encoding of each of the 4 classes: *Sit, Stand, Walk,* and *Smell*. TensorFlow Keras was used to implement the eCNN. The Adam algorithm was selected for optimization. We also used a dropout of 20% on the input and output of the dense layer to reduce overfitting.

2.3.3 Train and Test

The dataset of images, motion, audio, and ground truth labels were randomly split into training, validation, and testing sets. 70% of the data which included 3505 samples were used for training. 10% of the data which included 458 samples were used for validation. The remaining 20% of the data which included 954 samples were used for testing. When training the eCNN model, to prevent the optimization algorithm from getting stuck in local minima, we adopted mini-batches at two levels. First, the training samples were divided into a number of batches, referred to as "hyper-batches". Second, when each hyper-batch was used for training, mini-batches within the hyper-batch were used for optimization in TensorFlow. Similarly, training was run over epochs at two levels. First, when each hyper-batch was used for training, TensorFlow ran optimization over multiple epochs. Second, we ran training on all hyper-batches over multiple hyper-epochs. Model performance over the number of hyper-batches and hyper-epochs was evaluated.

To evaluate the model's performance, we computed the overall prediction accuracy as well as a confusion matrix to assess the prediction accuracy in each class. We use $\{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\}$ to represent the set of data samples, where x_i is the input feature and y_i is the ground truth class label, i.e. $y_i \in \{sit, stand, walk, smell\}$. We use $\hat{y}_i$ to represent the predicted class label for input x_i. The accuracy of class C $(C = sit, stand, walk, smell)$ is computed as

$$accuracy(C) = \frac{|\{i: y_i = C, \hat{y}_i = C\}|}{|\{i: y_i = C\}|} \tag{6}$$

$|S|$ represents the number of samples (cardinality) of set S. The overall accuracy is defined as the average accuracy across all classes.

$$accuracy = \frac{1}{4}\big(accuracy(sit) + accuracy(stand) + accuracy(walk) + accuracy(smell)\big) \quad (7)$$

A confusion matrix is defined as a two dimensional matrix $[M_{r,c}]$, where the rows r ($r \in \{sit, stand, walk, smell\}$) represent the ground truth class labels and the columns c ($c \in \{sit, stand, walk, smell\}$) represent the predicted class labels. A value in row r and column c represents the number of data samples that have a ground truth class r and were predicted as class c.

$$M_{r,c} = |\{i: y_i = r, \hat{y}_i = c\}| \quad (8)$$

3. Results and Discussion

We trained the eCNN model on a training set of 3505 samples and a validation set of 458 samples. The model was tested on a testing set of 954 samples. The training set includes 338 samples labeled as *Sit*, 521 samples labeled as *Stand*, 1574 samples labeled as *Walk*, and 1072 samples labeled as *Smell*. The validation set includes 55 samples labeled as *Sit*, 65 samples labeled as *Stand*, 207 samples labeled as *Walk*, and 131 samples labeled as *Smell*. The testing set includes 102 samples labeled as *Sit*, 149 samples labeled as *Stand*, 427 samples labeled as *Walk*, and 276 samples labeled as *Smell*. The experiments were run with 1 hyper-batch and 4 hyper-batches over 40 hyper-epochs. The best overall accuracy on the validation set was used to select the best number of hyper-epochs and number of hyper-batches. Training was run on a Windows machine with 24.0 GB RAM and 2.3 GHz AMD Ryzen 5 processor. One round of training took approximately 7 hours.

Table 1 shows the performance of the model over 40 hyper-epochs and 4 hyper-batches selected by validation. On the validation dataset, the model achieved an overall accuracy of 79.47%. It correctly predicted 88.00% of samples labeled as *Sit*, 72.73% of samples labeled as *Stand*, 78.95% of samples labeled as *Walk*, and 78.20% of samples labeled as *Smell*. On the testing dataset, the model achieved an overall accuracy of 79.02%. It correctly predicted 84.21% of samples labeled as *Sit*, 78.87% of samples labeled as *Stand*, 78.66% of samples labeled as *Walk*, and 74.33% of samples labeled as *Smell*. The performance of the eCNN model on the validation and testing sets were very similar, showing no obvious overfitting. The model achieved highest accuracy for class *Sit*, but performance was relatively similar over all classes.

The confusion matrix on the testing set is as follows:

Prediction / Ground truth	sit	stand	walk	smell
sit	80	5	9	1
stand	3	112	23	4
walk	15	26	328	48
smell	4	6	67	223

Table 1. Experimental results.

	Training	Validation	Testing
Number of Samples	3505	458	954
Number of Samples: class *Sit*	338	55	102
Number of Samples: class *Stand*	521	65	149
Number of Samples: class *Walk*	1574	207	427
Number of Samples: class *Smell*	1072	131	276
Overall Accuracy	94.34%	79.47%	79.02%
Accuracy of class *Sit*	99.11%	88.00%	84.21%
Accuracy of class *Stand*	95.59%	72.73%	78.87%
Accuracy of class *Walk*	96.19%	78.95%	78.66%
Accuracy of class *Smell*	86.47%	78.20%	74.33%

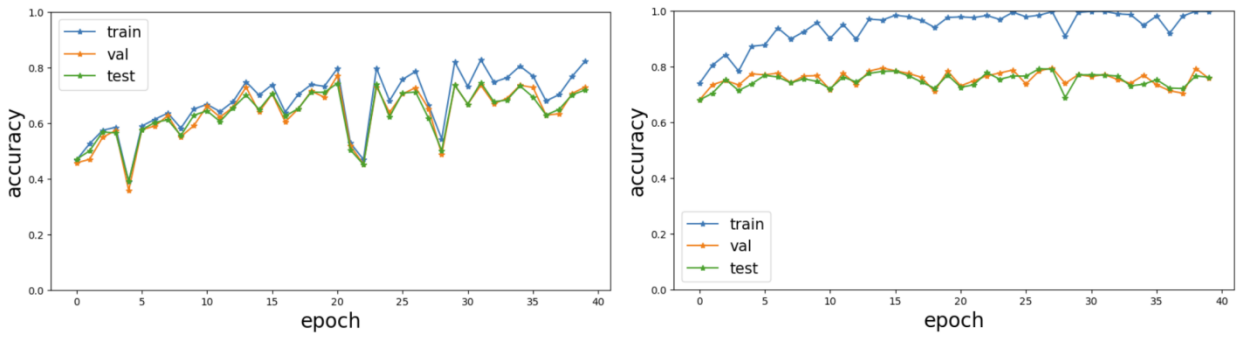


Figure 10. Overall performance of model over different hyper-epochs (Left: 1 batch used; Right: 4 batches used. Blue: training; Orange: validation; Green: test)

To evaluate the effects of multiple hyper-epochs and hyper-batches on the performance of the model, we plotted the overall accuracy over different numbers of hyper-epochs and hyper-batches in Figure 10. With 1 batch, the training reached optimal performance at around 20 hyper-epochs. With 4 batches, the training reached optimal performance much earlier at around 5 hyper-epochs. In both tests, the model's performance on validation and testing sets were very similar, suggesting no obvious overfitting. However, with 4 batches, there is a relatively large difference between the training

performance and the validation performance. This was likely caused by the smaller number of data samples in each batch fed into the training algorithm. The performance of each class over different hyper-epochs is shown in Figure 11. Although each class had a different number of training samples, the model's performance was consistent across all classes.

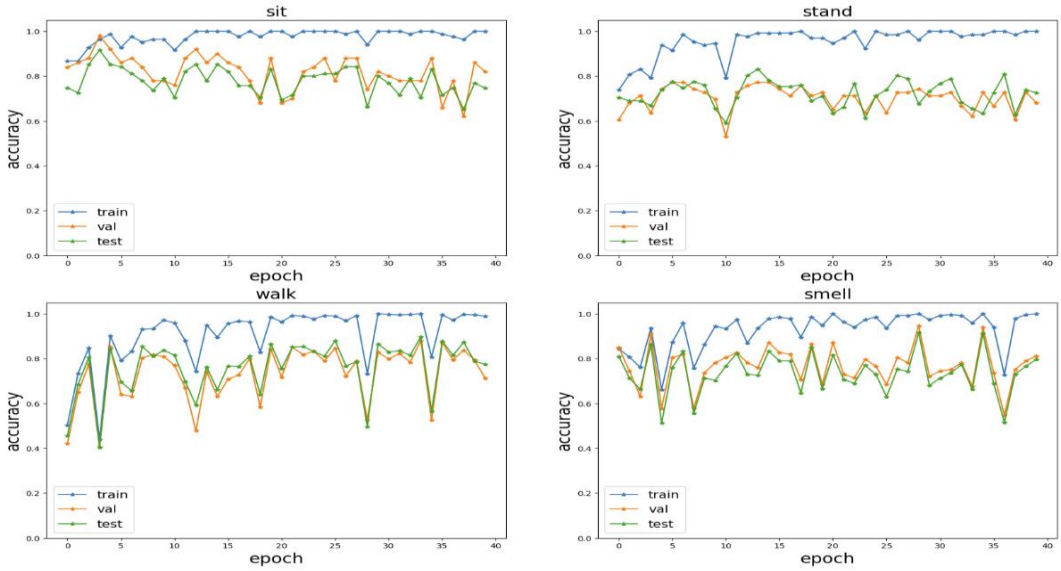


Figure 11. Performance on each class over different hyper-epochs (Upper left: *Sit*; Upper right: *Stand*, Lower left: *Walk*, Lower right: *Smell*)

Figure 12 shows the 32 convolution filters of size 7x7 that were learned by the model. They represent various color patterns. Many filters have a color difference along a diagonal, which suggests that the dog pays more attention and reacts to color differences in its field of view.

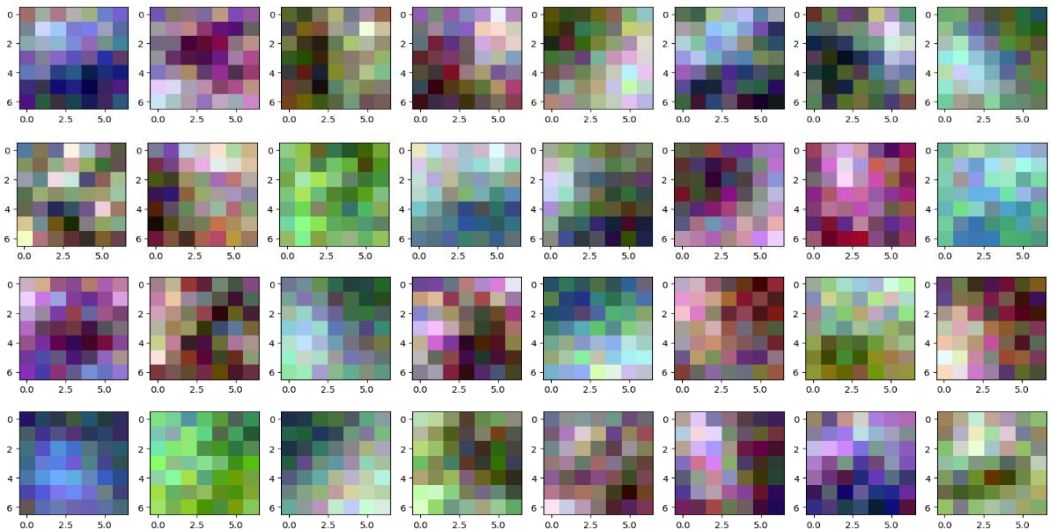


Figure 12. 32 convolutional filters with size 7x7 learned by the eCNN model.

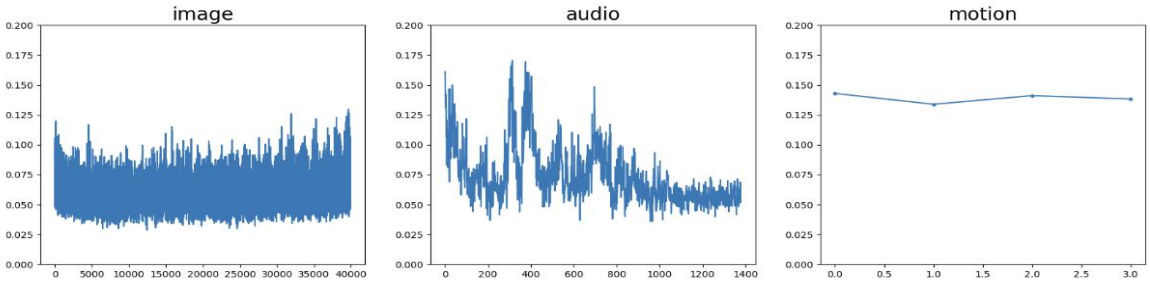


Figure 13. Average weights on image, audio, and motion features in dense layer.

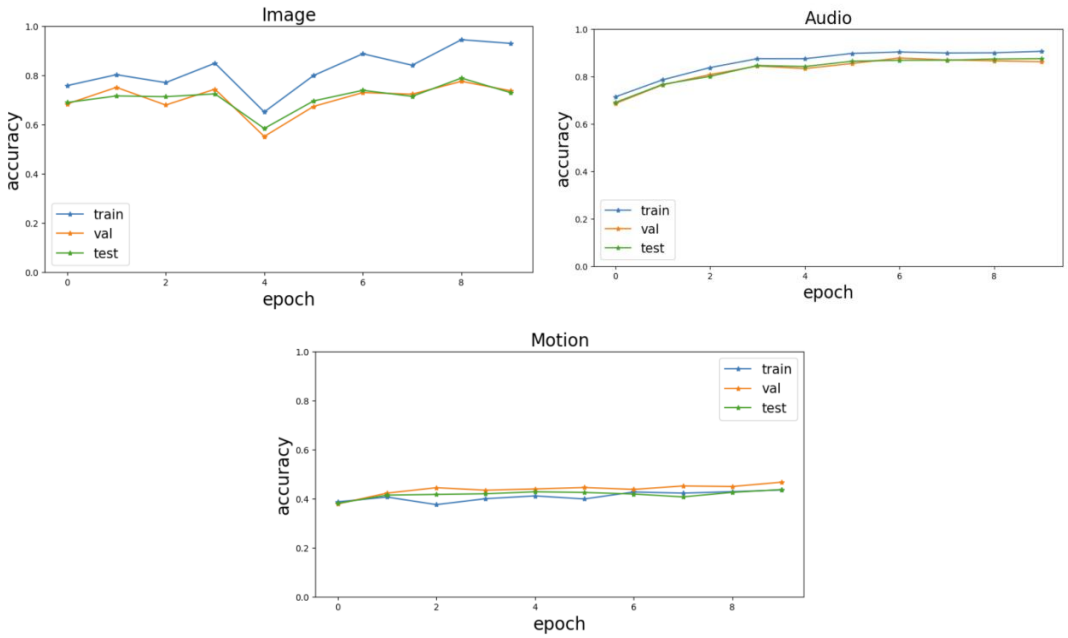


Figure 14. Overall performance of model using single-modal data (Upper-left: used only images; Upper-right: used only audio; Bottom: used only image)

To understand the role each sensing modality plays in predicting the dog's actions, we first displayed the weights on the input nodes learned by the model which correspond to image, audio, and motion features respectively. For each input node corresponding to image features, we average the magnitude of the weights of the dense layer nodes connected to that input node. The average weight is calculated for all input notes corresponding to image features and is shown in Figure 13. Similarly, the average weights for audio and motion features are calculated and shown in Figure 13 as well. We observe that for the audio features, some low frequency features are weighted more, which suggests that the dog likely reacts more to the low frequency components in what it hears.

In addition, we trained the model separately using only image, only audio, and only motion information. 10 hyper-epochs and 4 hyper-batches were used in the tests, shown in Figure 14. The optimal performance of the model trained with each type of single-modal input is shown in Table 2. The

model achieved the highest overall accuracy when using only audio as input, suggesting that audio plays a significant role in the dog's behavior. Training using only motion information had a much lower performance than the multi-modal input, which is likely due to inadequate background motion correction.

Table 2. Optimal performance of model trained with only image, only audio, and only motion information.

	Image only	Audio only	Motion only
Overall Accuracy	78.81%	86.74%	43.66%

4. Conclusion and Future Work

In this work, we proposed a research framework to understand dog behavior. We collected video and audio data from a dog's egocentric view and, through deep learning, learned the association between the dog's reaction and the visual and audio stimuli perceived by the dog. We proposed an extended Convolutional Neural Network (eCNN) to utilize multi-modality features of images, audio, and motion information. The model achieved promising results with an overall prediction accuracy of 79.02%. We observed that the dog reacts strongly to various color patterns and color contrasts in its field of view. It also reacts more to some low frequency components in what it hears. These findings can offer useful information when designing effective ways to train dogs for various services, such as companionship and rescue work.

In the future, we plan to extend our work in the following directions. First, we plan to add infrared sensors to study if and how dogs react to temperature. Second, we plan to test sequence models such as Recurrent Neural Networks for potential performance improvements. Third, we plan to extend data collection to study how a dog reacts to unfamiliar situations, human voices, other dogs barking, music, and much more. Lastly, we plan to extend the study to different dogs and understand the general and individual behavior of dogs.

References

[1] K. Ehsani, H. Bagherinezhad , J. Redmon, R. Mottaghi, A. Farhadi, Who let the dogs out? modeling dog behavior from visual data, *CVPR*, pp. 4051-4060, 2018.

[2] J. Hsu, "Dog cam" trains computer vision software for robot dogs, *IEEE Spectrum*, Apr 18, 2018.

[3] M. Marks, Q. Jin, O. Sturman, L. von Ziegler, S. Kollmorgen, W. von der Behrens, V. Mante, J. Bohacek, M. Fatih Yanik, Deep-learning-based identification, tracking, pose estimation and behaviour classification of interacting primates and mice in complex environments. *Nat Mach Intell* 4, pp. 331–340, 2022.

[4] J. Bryner, Brain scans reveal dogs' thoughts, *Scientific American*, May 8, 2022.

[5] Canine senses, Paws Chicago (https://www.pawschicago.org/news-resources/all-about-dogs/doggy-basics/canine-senses).

[6] J. Stromberg, Why scientists believe dogs are smarter than we give them credit, *Vox*, Jan 22, 2016.

[7] P. Agrawal, J. Carreira, and J. Malik, Learning to see by moving, *International Conference on Computer Vision*, pp. 37-45, 2015.

[8] A. Fathi, A. Farhadi and J. M. Rehg, Understanding egocentric activities, *International Conference on Computer Vision*, pp. 407-414, 2011.

[9] Y Y. J. Lee, J. Ghosh and K. Grauman, Discovering important people and objects for egocentric video summarization, *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1346-1353, 2012.

[10] S. L. Pintea, J. C. van Gemert, and A. W. M. Smeulders, Dej´ a` vu: Motion prediction in static images, *European Conference on Computer Vision*, pp 172–187, 2014.

[11] R. Gonzalez and R. Woods, Digital Image Processing, 3rd Edition, *Pearson Prentice Hall*, pp. 414-428, 1992.

[12] J. Kim and N. Moon, Dog Behavior Recognition Based on Multimodal Data from a Camera and Wearable Device, *Applied Sciences,* 12(6):3199, 2022.

[13] A. Hussain, S. Ali, Abdullah, H.-C. Kim, Activity Detection for the Wellbeing of Dogs Using Wearable Sensors Based on Deep Learning*, IEEE Access*, vol. 10, pp. 53153-53163, 2022.

[14] C.T. Siwak, H.L. Murphey, B.A. Muggenburg, N.W. Milgram, Age-dependent decline in locomotor activity in dogs is environment specific, *Physiology & Behav*ior, 75(1-2), pp. 65-70, 2002.

[15] A. Quaranta, M. Siniscalchi, and G. Vallortigara, Asymmetric tail-wagging response by dogs to different emotive stimuli, *Current Biology*, vol. 17, no. 6, pp. 199-201, 2007.

[16] C.J. Völter, L. Lonardo, M.G.G.M. Steinmann, C.F. Ramos, K. Gerwisch, M.T. Schranz, I. Dobernig, L. Huber. Unwilling or unable? Using three-dimensional tracking to evaluate dogs' reactions to differing human intentions, *Proceedings of the Royal Society*, 290(1991), 2023.

[17] Y. Lecun, L. Bottou, Y. Bengio and P. Haffner, Gradient-based learning applied to document recognition, *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278-2324, Nov. 1998.

[18] J. Martinez, M. J. Black, J. Romero, On human motion prediction using recurrent neural networks, *IEEE Computer Vision and Pattern Recognition Conference*, pp. 2891-2900, 2017.

[19] J. Liu, A. Shahroudy, D. Xu, and G. Wang, Spatio-temporal lstm with trust gates for 3d human action recognition, *European Conference on Computer Vision*, pp 816–833, 2016.

[20] S. Venugopalan, M. Rohrbach, J. Donahue, R. Mooney, T. Darrell, K. Saenko, Sequence to Sequence – Video to Text, *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4534-4542, 2015.