國立臺灣大學生農學院生物機電工程學系

學士論文

Department of Biomechatronics Engineering

College of Bioresources and Agriculture

National Taiwan University

Bachelor Thesis

應用慣性測量單元與機器學習於犬類行為偵測

Dog behavior detection using IMU and machine learning method

侯博瀚

Po-Han Hou

指導教授：林達德 博士

Advisor: Ta-Te Lin, Ph.D.

中華民國 110 年 7 月

July 2021

Abstract

Human have always wanted to understand more about animal behaviors. Among all of the animals, dogs are one of the most intimate companion with human. Through their behaviors, we can get a lot of information, such as health condition or mental condition. Therefore, this research aims to develop an automatic detection system to detect five different behaviors, standing, walking, running, eating, resting, of a dog by IMU (inertial measurement unit) data and machine learning method. The IMU will be fixed on the dog's neck and the data sent back will be labeled according to a validation video. A classification model will be trained and tested with these data. In this research, I used Random Forest, Deep Neural Network, K-nearest Neighbor and Support Vector Machine to train the model. The accuracy of the models was tested by a hold out set, and the Deep Neural Network model reached the accuracy and f1 score of 0.95.

Table of Contents

List of Figure

List of Tables

Chapter 1. Preface

In recent years, the health condition of pets has become an issue that many people care about. Many study indicates that their health condition are closely related to their behaviors. Therefore, understanding pets’ behaviors is an important part to evaluate their healthiness. Dogs are one of the most popular pets. By observing their behaviors, we can gain more insight into their health condition and how to take care of them. Dog behavior information is also needed for dogs to do specific jobs, e.g. rescue dogs, guard dogs, in order to train them more effectively. Hence, human developed many ways to study dog behaviors. The most straight forward way is direct observation. However, this method is not only time consuming, but may also have negative effect on their behaviors and healthiness. Thus, this research tries to develop a system that can detect and classify dog behaviors automatically using IMU (inertial measurement unit) and deep learning method.

Chapter 2. Literature Reviews

Dog behaviors has always been an indication of its mood or health condition. According to Daniel Mills et.al, when a dog keeps walking in a spot, it is likely to be an expression of excitement [1]. When a dog is in high stress, it may result in restlessness behavior and decreased appetite or water intake. Therefore, by observing the behaviors of dogs, we can be aware of their conditions and take better care of those dogs.

There are many approach to observe dogs' behaviors. One of the approach is direct human observation. However, this method is not ideal mainly because of two reasons. First of all, continuous observation by a human and interfering with the normal life of pets by continuous observation can have a negative impact on a pet's behavior and health. Secondly, the observation process is time consuming and labor-intensive [2]. Furthermore, human observation may be limited by the environment, due to difficulties when observing something to far away or places that are too dark. People's opinion on how a behavior should be classified may also vary, which may affect the consistency of the data.

Marius Baba et al. used images from surveillance camera to train a classification model which is used to discriminate normal behaviors and dangerous behaviors of wild dogs [3]. They used the Support Vector Machine algorithm and their model reached the accuracy of 99%. Nevertheless, this method requires a lot of surveillance camera to cover the whole area the dogs live. Moreover, this method is affected by weathers such as rainy or snowy days, which may result in difficulties to acquire clear images. On the whole, using image to detect dog behaviors may cost a lot of money and hard to put into practical use.

Patricia Pons et al. proposed a depth based tracking method to develop an automatic behavior detection system for cats [4]. They used Microsoft Kinect® v1.0 sensor to record video streams of depth and color information from the cats' movements. The depth based data was used to train a classifier model with different algorithms. A combined model has been built using a stacking approach and considering the three best base learning algorithms: rule induction, support vector machine and decision tree, with an average accuracy of 83.18% for all postures. Although non-wearable tracking sensors may have less interference on animal's movements, these methods are strongly limited to a certain range which the tracking sensor can cover. The Microsoft Kinect® v1.0 sensor can only cover an area of approximately 200 cm long and 270 cm wide, which might be too small for active pets like dogs.

Due to the disadvantages of the methods above, this research decides to use an IMU (inertial measurement unit) to acquire motion data from the dog. Using IMU has many advantages including: (1) IMU is small in size with light weight, so that it can be attached on the dog without interfering their motion; (2) IMU is relatively cheap; (3) The sampling frequency can be up to several thousand Hz and can acquire data for a long period, so it is easier to get more data. On the other hand, the Global positioning system(GPS), which is another method to get motion data, only has the sampling frequency of 5-20 Hz; (4) IMU can work independently without external aid, unlike Global Navigation Satellite System (GNSS) or Global System for Mobile Communications (GSM) chips which needs satellite signal in order to measure motion data [5].

Gregory J. Jenkins et al. used a six-degree accelerometer and gyroscope IMU for gait analysis [6]. They used a detection algorithm to process the data and detect 1259 strides. A validation video was used to evaluate the result, and only one stride was missed. The accuracy is very high, but this research only focus on gait posture.

Linda Gerencsér et al. [5] and Satyabrata Aich et al. [2] both used IMU to obtain data for their dog behavior model. Yet, only accelerometer and gyroscope data are used for development of the model. Michael Winters et al. used IMU data to train a supervised model and an unsupervised model [7]. However, the accuracy was only 82.06% for the supervised model and 74.25% for the supervised model. Also, only the Random Forest machine learning method was used, so there is lack of comparison between different models.

Chapter 3. Materials and Methods

The IMU will be attached to a dog, and the motion data sent back will be labeled by a validation video recorded at the same time. The data will be preprocessed and used to train and test the model.

3.1 Inertial Measurement Unit

Inertial measurement unit (IMU) is a sensor that detects motion, usually consisted of an accelerometer and a gyroscope, both with three axes (x, y, z). Some may include a magnetometer also with three axes. It works by detecting linear acceleration using the accelerometer and rotational rate using the gyroscope. The magnetometer is commonly used as head reference. The sensor used in this research is LPMS-B2, a 9 axes IMU, with accelerometer, gyroscope and magnetometer. It uses Bluetooth to connect with data receiving device within the range of 20 meters. The size is 39*39*8 mm with the weight of 12g. The sampling rate can be up to 400Hz and the battery can last more than 6 hours. Additionally, LPMS-B2 provides library in C++ language for users to develop their own application in order to acquire or compute statistics the user needs.

3.2 Subject and environment

The subject of this research is a 10 years old male Chihuahua. By feeding and playing with him, five activities eating, standing, resting, walking and running were ensured to be performed by the dog. The front yard, an empty place with better vision, was chosen to collect the data in order to record the validation video.

3.3 IMU data

The IMU was fixed on the top of the collar around the dog’s neck, as figure 1 shows. A C++ program was written to collect 9 axes data of the IMU through Bluetooth, modifying the sampling rate to 33Hz, and the collecting time was set to 1 hour. Every sample was marked with a timestamp to label the behaviors according to the validation video. 294435 samples were collected in total and showed in figure 2.

Figure 1. Position of IMU on the dog

accx	accy	accz	gyrox	gyroy	gyroz	magx	magy	magz	timestamp	classification
-0.73594	0.767439	0.162755	-5.82082	-4.51201	-3.83357	0.847705	57.9363	-76.3812	19:38:57	resting
-0.71249	0.713654	0.172607	7.25918	-0.78951	-3.09857	0.380006	56.1239	-77.0243	19:38:57	resting
-0.73828	0.773926	0.168213	-8.63082	-4.02701	-1.66357	0.204619	57.1763	-77.0535	19:38:58	resting
-0.67801	0.728638	0.130402	-5.53332	-10.3795	-0.85857	0.409237	55.4224	-76.615	19:38:58	resting
-0.66046	0.682007	0.154999	-4.51832	-1.36701	5.49393	0.935399	57.2932	-76.7319	19:38:58	resting
-0.70056	0.746429	0.159302	-3.24082	-0.91201	3.51643	1.4031	57.4686	-76.6735	19:38:58	resting
-0.6918	0.666748	0.151855	-0.72082	-4.27201	2.06393	0.496931	57.0593	-76.7319	19:38:58	resting
-0.6904	0.629578	0.153656	5.19418	0.487988	7.61143	0.789243	56.884	-76.2643	19:38:58	resting
-0.68784	0.695313	0.174469	12.7892	12.773	3.72643	1.25694	56.6501	-76.5858	19:38:58	resting
-0.673	0.772919	0.176575	-6.39082	7.06799	-8.01608	0.613856	56.8547	-76.5566	19:38:58	resting
-0.70422	0.689484	0.160309	0.20668	-6.63451	-6.44107	0.935399	57.2932	-76.995	19:38:58	resting
-0.71121	0.734131	0.17453	5.24668	4.07549	0.453929	0.96463	56.387	-76.0889	19:38:58	resting
-0.68726	0.74176	0.168457	8.18668	3.77799	-5.61857	0.584624	57.0009	-76.4396	19:38:58	resting
-0.72565	0.757172	0.158539	-1.75332	-4.62201	-1.31357	0.643087	56.7963	-77.0827	19:38:58	resting
-0.72073	0.749634	0.154938	-3.64332	-8.94452	-1.90857	0.613856	56.7378	-76.9658	19:38:58	resting
-0.66727	0.668518	0.158081	-3.59082	-2.31201	6.66643	0.760012	57.0886	-76.1181	19:38:58	resting
-0.68356	0.723358	0.156982	-4.55332	-0.29951	2.93893	0.760012	55.8901	-76.4396	19:38:58	resting
-0.67981	0.649139	0.14209	4.58168	-0.82451	5.35393	0.672318	56.6793	-76.7612	19:38:58	resting
-0.69309	0.692566	0.174988	9.34168	7.03299	2.67643	0.321543	56.8547	-76.9073	19:38:58	resting
-0.70227	0.733826	0.165741	-1.17582	5.51049	-3.67607	1.14002	56.767	-76.6443	19:38:58	resting
-0.70212	0.712097	0.174103	-3.08332	-1.87451	-1.90857	0.584624	56.884	-76.6735	19:38:58	resting
-0.71082	0.717926	0.164703	-0.87832	-0.08951	-4.23607	0.526162	56.4163	-76.6735	19:38:58	resting
-0.69714	0.728912	0.170441	5.75418	4.39049	-1.89107	0.906168	56.767	-76.8196	19:38:58	resting
-0.68774	0.734161	0.16745	10.3392	1.97549	-5.40857	0.263081	57.0886	-76.8489	19:38:58	resting
-0.68677	0.703949	0.142609	4.17918	-4.60451	-3.55357	0.584624	56.387	-76.9366	19:38:58	resting

Figure 2. Raw data

3.4 Validation video

A Raspberry Pi 3 and a Pi camera V2 is used to record the validation video. A recording device was built as figure 3 shows, and placed at one edge of the front yard with a wide angle to cover as much area as possible. A python program was written to show the timestamp on the video, and the frame rate was set to 24fps, which is enough to determine the behavior clearly.

Figure 3. Recording device

3.5 Data processing

Five different behaviors, resting, eating, standing, walking and running was labeled according to the timestamp and the validation video. Only behaviors that lasted more than 1 second was labeled. There was a total of 102508 labeled samples, with 38549 resting samples, 17587 eating samples, 32100 standing samples and 11941 walking samples, 2330 running samples as figure 4 shows.

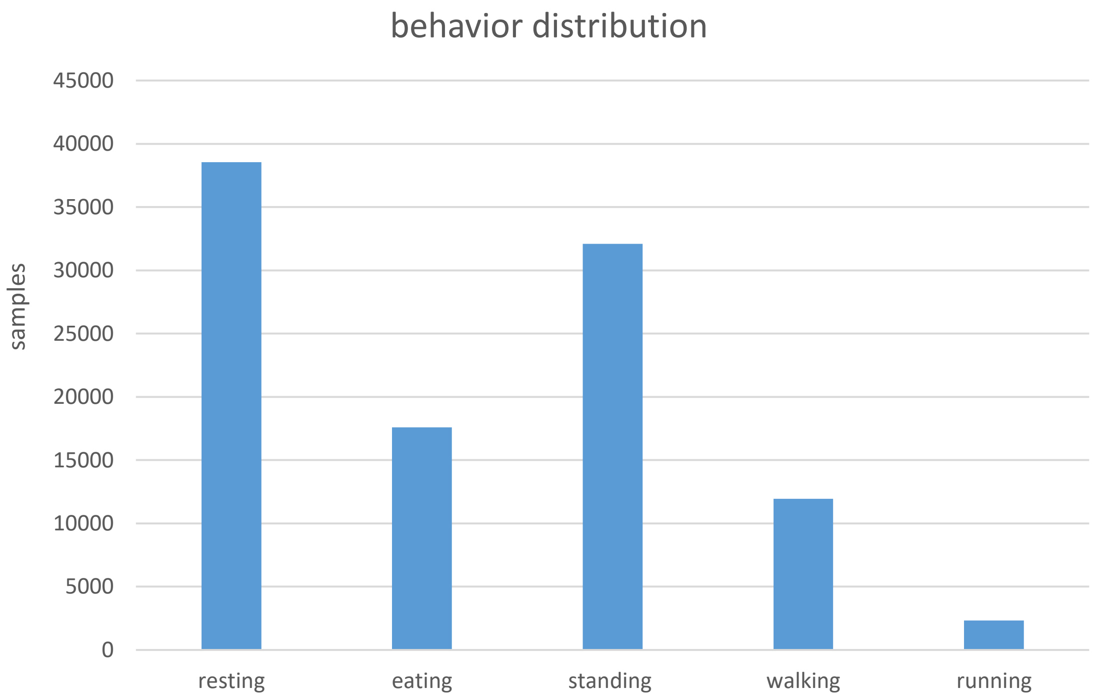

Figure 4. behavior distribution

After labeling, all samples were divided into windows of fixed size in temporal order. The rolling window is half the size of a group. Size of 8, 16, 33 and 64 were all experimented to find the most appropriate window size. If all samples in the window has the same label, the mean, standard deviation, maximum and minimum of 9 axes will be calculated for each window and the whole window will be given a behavior label. Every labeled window will be an input instance for the model with the total of 36 features. Part of the processed data is showed in figure 5.

accx_avg	accy_avg	accz_avg	gyrox_av	gyroy_av	gyroz_av	magx_av	magy_av	magz_av	accx_std	accy_std	accz_std	gyrox_std	gyroy_std	gyroz_std	magx_std	magy_std	magz_std	accx_max	accy_max
-0.8678	-0.4167	-0.2379	10.5534	1.8935	3.3606	-6.4761	-17.685	-103.45	0.07465	0.05989	0.0522	36.6186	19.7838	20.9117	1.0628	1.98604	2.80796	1.0274	0.56162
-0.8968	-0.3988	-0.1829	7.73005	1.10843	2.7002	-8.4501	-23.396	-91.206	0.09275	0.05709	0.0642	26.4555	25.4699	15.3879	0.85613	1.2918	3.71224	1.05609	0.50891
-0.9039	-0.3797	-0.014	10.0947	-2.1067	-4.5525	-12.99	-24.805	-75.848	0.08766	0.13202	0.11023	36.314	25.8775	30.4183	3.9102	1.60405	8.3997	1.25375	0.57559
-0.9416	-0.2897	-0.0971	2.34579	0.58972	0.33278	-14.839	-16.596	-101.82	0.10966	0.09249	0.071	39.4228	30.1998	23.7978	3.77488	3.4145	4.18911	1.17407	0.50256
-0.9484	-0.3132	-0.0453	-2.3904	-0.6484	1.28379	-14.813	-17.02	-101.65	0.07651	0.07089	0.04938	20.8979	18.408	19.4464	1.64886	1.24089	1.3222	1.23636	0.5271
-0.9221	-0.3727	-0.0385	3.15763	-0.1693	-4.1989	-14.136	-14.863	-105.25	0.07687	0.08949	0.06096	24.135	21.3135	19.6357	2.1121	1.47368	1.51813	1.08328	0.55014
-0.8835	-0.4424	-0.1059	-8.9866	0.42292	-0.0323	-11.543	-10.616	-109.71	0.11324	0.0628	0.08153	26.0977	22.6727	15.1509	0.77227	4.69453	4.08433	1.18735	0.56787
-0.8599	0.49565	-0.0137	7.69232	-6.9129	7.42082	-42.068	0.65359	-91.869	0.09468	0.08233	0.04005	21.2875	11.7427	32.5504	2.68234	2.45351	5.03636	1.09036	0.67374
-0.8802	-0.4475	-0.0056	19.4772	5.79488	-2.9425	-7.9998	-22.774	-86.983	0.09574	0.09592	0.08608	38.4961	30.5172	24.7827	2.45038	2.74359	8.93181	1.06317	0.61105
-0.9119	-0.3898	0.01503	-1.0848	-0.2839	3.41278	-10.525	-24.309	-74.378	0.16284	0.06092	0.09796	28.3848	55.5134	22.1423	1.11091	0.66631	3.1381	1.35593	0.54446
-0.9195	-0.3666	-0.016	1.43635	-1.1102	2.60368	-11.571	-24.638	-76.352	0.0874	0.06178	0.06582	32.9616	38.1556	20.2037	1.48016	0.74261	2.35126	1.10483	0.49899
-0.9695	0.12071	0.1261	6.3954	-2.2992	16.4675	-30.994	-14.828	-73.263	0.06762	0.20181	0.0735	24.9477	24.0847	14.7309	7.09748	5.93341	2.57685	1.16116	0.46018
-0.9318	-0.2676	-0.0232	-3.5949	7.21483	-17.489	-15.969	-21.242	-85.953	0.07666	0.21049	0.06907	45.4697	24.532	42.6556	8.38735	4.47003	6.38094	1.08728	0.48999
-0.9232	-0.3629	-0.016	1.74012	-0.3312	-2.3644	-11.119	-24.156	-81.741	0.09467	0.07829	0.07862	32.9106	31.1178	19.6086	1.88648	0.76338	1.57148	1.17459	0.69846
-0.9385	-0.3078	0.03008	-12.181	1.09667	-3.1992	-13.52	-22.421	-84.068	0.08167	0.07603	0.09675	41.6291	28.3048	24.145	2.24784	0.79603	6.47177	1.16653	0.45905
-0.9205	-0.3791	-0.0523	6.12744	-2.1414	0.47305	-10.922	-22.056	-89.175	0.1137	0.11078	0.08461	50.4228	35.976	26.903	3.84507	1.11131	3.04591	1.11853	0.60422
-0.936	-0.3228	0.07199	6.0465	-3.4069	8.7243	-13.86	-21.944	-86.023	0.10994	0.11733	0.05084	32.2581	24.5691	18.1972	4.46354	0.97251	1.60833	1.12225	0.57538
-0.877	0.47083	0.08549	-5.0305	2.33588	-5.9423	-43.336	-1.6698	-84.456	0.06508	0.07963	0.04123	18.4258	12.3693	16.1042	1.59567	2.01479	2.29907	1.03741	0.61389
-0.9689	-0.1145	0.1008	14.2097	3.24861	-10.188	-23.082	-19.367	-79.761	0.14328	0.19553	0.07754	49.8336	34.0812	43.487	7.14347	4.7035	7.68958	1.42725	0.46372
-0.9377	0.29622	0.08981	-10.169	4.39952	6.47446	-37.917	-8.9708	-78.392	0.11729	0.14674	0.04118	27.8045	15.3242	40.5689	4.91768	4.8262	4.59064	1.23752	0.49176
-0.9397	-0.2123	-0.0256	-28.228	-4.7374	-3.4899	-18.223	-16.333	-76.317	0.09659	0.13988	0.22289	65.6512	57.1448	33.8692	4.4902	4.34336	22.2833	1.2099	0.59171

Figure 5. processed data

3.6 Classification Model

This research used four different algorithms, including Random Forest, Deep Neural Network (DNN), K-nearest Neighbor (KNN) and Support Vector Machine (SVM) to train the classification model. Among the four models, DNN has the highest accuracy, so this research will focus on this deep learning method.

3.6.1 Deep learning

Deep learning is a class of machine learning algorithms that uses multiple layers to progressively compute more features from the input. Deep-learning architectures such as deep neural network, recurrent neural network, convolutional neural networks have been applied to many fields like computer vision, speech recognition and motion pattern recognition. They are capable of producing results comparable or even better than human performance.

3.6.2 Deep Neural Network

DNN is a kind of deep learning method which uses a mathematic model that imitates the structure and function of brain neural network. It repeatedly conducts different level of computation and training to find the most optimized and most effective model. The structure includes an input layer, multiple hidden layers and an output layer as figure 6 shows

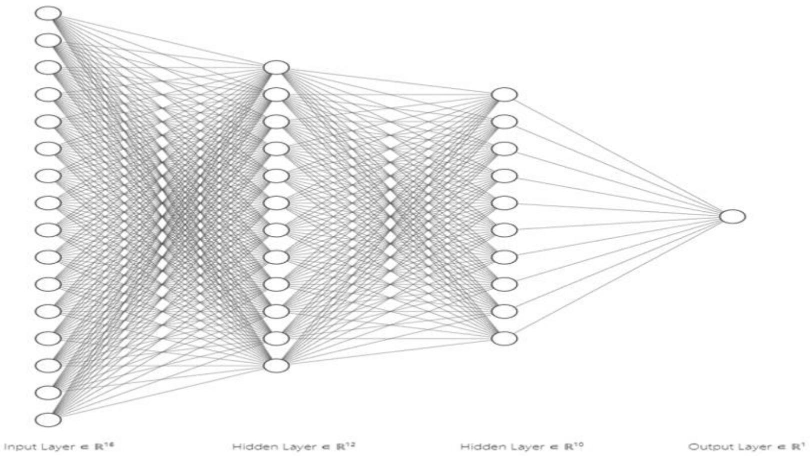

Figure 6. Deep neural network

The neurons in the network will be assigned a weight value when passed to another layer. Therefore, the neurons in the hidden layers and the output layer are the weighted sum of all neurons from the previous layer. In order to solve non-linear classification problems, the weighted sum will be multiplied with a non-linear function, which is called the activation function. The activation function used in this research are ReLU and Sigmoid which are defined as formula 1-1 and formula 1-2, respectively.

$$\mathrm{f(z)} = \max(z, 0) \tag{1-1}$$

$$\mathrm{f(z)} = \frac{1}{1 + e^{-x}} \tag{1-2}$$

ReLU function neglects negative input value and limits the output value to between

0 and ∞. Sigmoid function limits the output value to between 0 and 1.

DNN will use an optimizer to change the weight value continuously according to the loss from the loss function. A gradient descent algorithm is generally used to optimize and keep finding lower loss values. It involves a procedure of taking repeated steps in the opposite direction of the gradient of the function at the current point in order to find the local minimum. The final model will be generated after the loss converges.

In this research, the DNN model is constructed with Keras library in Python language. All value of the features was normalized to the range of 0 and 1 to obtain better results. The data was divided into 70% of training data and 30% of training data. The model has an input layer of 36 neurons matching the number of features and a output layer of 5 neurons matching the number of classification targets. There are three hidden layers each with 100 neurons and other parameters are showed in table 1.

Table 1. DNN parameters

	Value
Activation	ReLU for hidden layer Sigmoid for output layer
Optimizer	Adam
Loss	Sparse Categorical Cross Entropy
Batch Size	32
Epoch	100

Chapter 4. Results and discussions

There were 5635 data after pre-processing, 3945 data to train the model and a hold out set of 1690 data to test the model. The hold out set validation method is chosen because it is more similar to real life situation, since the model aims to predict new data that has never been observed before. Accuracy and F1 score, which are defined as formula 2-1 and 2-2, will be used as evaluation metrics.

$$\text{Accuracy} = \frac{\mathrm{TP+TN}}{TP+FP+FN+TN} \tag{2-1}$$

$$\mathrm{F_1\ score} = \frac{2TP}{2TP+FP+FN} \tag{2-2}$$

True Positive (TP) means the model predicts the behavior when it actually happens. False Positive (FP) means the model didn’t predict the behavior when it actually happens. False negative (FN) means the model predicts the behavior when it didn’t happen. True negative (TN) means the model didn’t predict the behavior when it really didn’t happen.

The accuracy and F1 score of every behavior will be averaged to obtain the final accuracy and F1 score. The final test accuracy and F1 score for four different models, Random Forest, Deep Neural Network (DNN), K-nearest Neighbor (KNN) and Support Vector Machine (SVM), are showed in table 2.

Table 2. Test results of different models

	Accuracy	F1 score
DNN	0.959	0.958
SVM	0.756	0.763
Random Forest	0.906	0.906
KNN	0.856	0.853

The DNN model has the best result, while KNN and Random Forest has decent accuracy and F1 score, both above 0.85. The SVM model didn't work well with much lower accuracy and F1 score compared to other models.

Data sets with distinct group size was used to train the neural network to find the most appropriate size. The result is showed in figure 7. The size of 33, which was chosen initially, still has the best accuracy and F1 score. However, sizes of 8,16,64 has similar results, with about 3% slightly lower accuracy and F1 score. Since the sampling rate of the IMU is 33Hz, a group size of 33 samples means a 1 second interval is chosen to compute features. Therefore, from this comparison, it can be known that useful features can be extracted from an interval of 0.25 seconds to 2 seconds to distinguish different behaviors. A longer time interval may contain more information, but it will also decrease the number of data. Thus, the 1 second interval is an ideal length with enough information to predict the behavior and also generates adequate data.

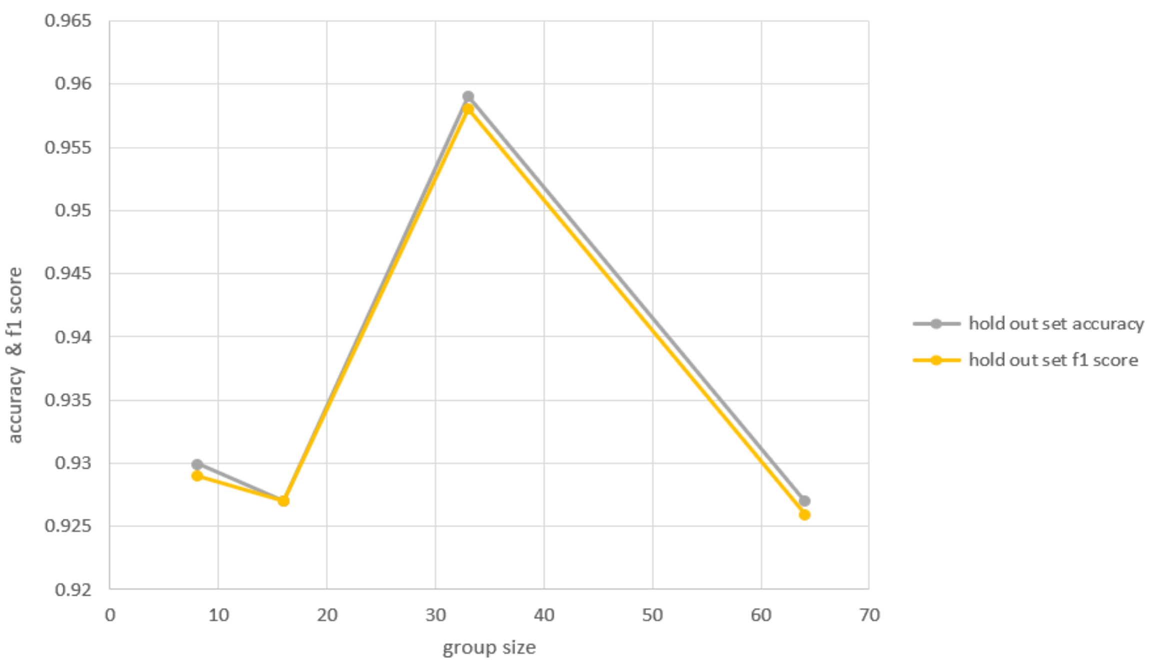

Figure 7. Group size comparison

The confusion matrix of the test data is showed in figure 8. The behaviors are presented in numbers from 0 to 4. Number 0 denotes resting, number 1 denotes eating, number 2 denotes standing, number 3 denotes walking and number 4 denotes running.

Predicted

	0	1	2	3	4	∑
0	667	0	16	2	0	685
1	1	302	4	0	0	307
2	8	7	505	8	1	529
3	6	0	8	135	6	155
4	0	0	0	4	10	14
∑	682	309	533	149	17	1690

Figure 8. Confusion matrix

The running behavior has the lowest accuracy and is frequently misclassified to walking behavior, since the motion pattern of the two behaviors are similar. However, the maximum and minimum feature or the standard deviation feature should be able to distinguish these two behaviors due to different motion intensity. The real problem is that the running sample is inadequate, because the test subject is a ten-year old Chihuahua which can’t run often due to his age. The lack of running data probably the main reason that the model can’t classify running behavior correctly. Although there were some methods to ensure the dog performed all five behaviors, the data collector, which was myself, was not an experienced dog trainer, so the dog may not follow the orders that easily. Therefore, the data may not be uniformly distributed as intended. By adding more running data, the accuracy of running classification is likely to improve.

On the other hand, eating behavior has the highest classification accuracy. One possible reason is its unique motion, which the dog lowers its head and chews the food continuously with its jaws, separates this behavior from others. For standing and resting behaviors, they have been misclassified with each other due to their relatively static motion. Nonetheless, the mistaken portion is pretty small, which is lower than 6%.

New data can be input to the model and predictions will be generated and showed on the python program as figure 9 below. To save space, the predictions are also presented with numbers from 0 to 4. Number 0 denotes resting, number 1 denotes eating, number 2 denotes standing, number 3 denotes walking and number 4 denotes running.

```
predictions shape: [3 2 3 0 0 2 0 0 1 0 0 0 0 0 2 1 0 3 1 2 2 0 0 2 2 1 0 3 0 0 1 3 0 3 0 2 0
 2 0 2 1 1 3 0 2 0 1 0 0 0 0 1 1 2 2 2 0 0 0 0 0 0 2 0 2 3 0 1 0 0 0 3 0 2
 3 0 1 0 0 0 0 2 2 2 2 0 2 2 2 2 3 0 0 3 0 3 1 1 3 0 0 0 0 0 1 2 0 0 1 2 3
 2 3 0 2 0 0 0 0 0 0 1 2 0 0 3 0 0 2 3 1 0 0 3 2 0 0 0 2 0 2 2 2 1 1 0 0 2
 0 1 0 2 2 0 0 0 0 0 3 0 0 0 2 3 1 0 0 2 2 1 0 0 2 0 0 0 0 2 0 0 0 1 0 1 3
 0 0 0 2 0 0 0 2 2 3 0 3 1 0 0 0 1 2 0 0 0 0 3 3 2 0 0 1 3 3 0 1 0 0 3 3 2
 2 0 1 0 0 2 0 0 2 0 0 0 1 0 2 0 3 0 2 0 1 2 0 1 0 1 3 0 0 0 0 1 0 3 2 0 2
 2 0 0 1 2 3 2 0 3 1 1 2 0 0 3 0 2 0 0 2 2 2 1 2 0 1 2 2 0 0 1 0 0 0 1 0 1
 2 2 0 2 1 2 2 1 0 2 0 2 2 0 0 2 2 0 2 0 0 0 0 0 1 1 1 0 0 0 2 0 1 1 1 0 0
 0 0 0 1 3 2 0 2 2 0 0 0 0 0 3 1 1 0 0 0 0 0 0 0 1 1 3 0 0 2 0 1 1 0 1 0 2
 0 0 0 0 2 2 0 0 1 0 2 0 0 1 3 0 2 0 0 2 2 0 3 2 2 1 2 1 3 2 2 0 2 0 2 1 0
 0 1 0 0 1 2 0 2 0 0 0 0 2 3 0 1 0 2 2 2 0 3 0 0 3 2 3 2 1 0 2 2 2 0 0 2 3
 3 0 1 2 0 2 0 2 3 2 0 1 0 0 3 1 3 2 2 3 0 0 2 1 1 1 0 0 0 2 0 0 2 3 3 3 3
 2 0 3 2 3 0 2 0 0 1 2 2 1 0 1 1 1 2 0 2 2 0 1 0 0 0 3 0 1 2 0 1 2 0 2 1 0
 2 0 0 1 0 0 1 0 0 2 1 1 0 0 2 3 1 0 1 1 0 1 0 3 2 1 2 2 0 0 0 0 2 0 2 1 0
 0 1 0 3 0 0 2 3 2 2 2 1 2 1 1 0 3 2 3 3 2 0 2 1 2 0 1 2 2 1 0 0 0 0 0 2 2
 2 0 2 0 0 2 2 2 1 2 0 1 0 0 2 0 1 2 2 0 1 1 1 2 0 0 3 0 0 0 1 0 0 0 2 0 1
 1 0 0 1 1 0 0 2 0 0 0 0 3 1 0 0 1 2 0 0 2 0 2 0 0 0 2 0 0 1 0 1 1 0 1 0 0
 0 0 1 0 2 3 0 2 0 1 3 1 0 0 2 2 0 0 2 2 0 1 0 3 1 0 2 2 0 1 0 0 1 3 0 1 2
 0 0 2 2 1 0 2 0 1 1 0 3 1 2 3 0 1 2 0 2 2 3 2 0 3 2 2 2 1 0 1 3 2 0 2 0 1
 0 1 0 2 1 2 0 0 0 0 3 0 2 0 0 2 3 0 2 2 1 2 3 0 2 0 1 1 0 3 0 1 2 2 1 2 2
 2 3]
```

Figure 9. Predictions

To show that the model can predict behaviors correctly, a C++ program is written to put the prediction on the validation video according to the data timestamp. The prediction text will be labeled on the video frame and whether the behavior in the video matches the text can be verified easily. Figure 10 to Figure 14 shows particular frames of the dog performing certain behaviors. The prediction is displayed on the upper left of the video with red font.

Figure 10. Validation video (eating)

Figure 11. Validation video (resting)

Figure 12. Validation video (standing)

Figure 13. Validation video (walking)

Figure 14. Validation video (running)

Chapter 5. Conclusions

In this research, IMU data and machine learning method were used to predict dog behaviors automatically. The automatic detection system developed in this research is able to monitor pet dogs efficiently by wearing the IMU sensor on the dog's collar. The IMU method has low cost and can be implemented easily. Features extracted from the IMU data can classify different dog behaviors accurately. The average accuracy and F1 score of deep neural network model reached 0.95. To highlight performance of specific behaviors, we use recall as metric for individual behaviors. Three of them, including eating, resting, and standing, reached the recall of 0.95 or higher. Walking behavior has decent recall at 0.90. The running behavior has the lowest precision of 0.71 due to scarcity of running data. However, this issue can be handled by changing to a more energetic subject or by more professional dog training methods. On the whole, the result shows that deep learning method can be used to develop a precise classification model which can be put into practical use.

There is still some improvement that can be done in the future. More activities can be added to the detection list, such as jumping or barking. The method in this study can also be tested on different breeds of dogs and used to develop an automatic system that can detect behaviors correctly from all kinds of dogs. With some further work, this system can be expected to help monitoring dog behaviors in real life.

References

[1] Mills, D., Karagiannis, C., & Zulch, H. (2014). Stress—its effects on health and behavior: a guide for practitioners. *Veterinary Clinics: Small Animal Practice*, *44*(3), 525-541.

[2] Aich, S., Chakraborty, S., Sim, J. S., Jang, D. J., & Kim, H. C. (2019). The design of an automated system for the analysis of the activity and emotional patterns of dogs with wearable sensors using machine learning. *Applied Sciences*, *9*(22), 4938.

[3] Baba, M., Pescaru, D., Gui, V., & Jian, I. (2016, October). Stray dogs behavior detection in urban area video surveillance streams. In *2016 12th IEEE International Symposium on Electronics and Telecommunications (ISETC)* (pp. 313-316). IEEE.

[4] Pons, P., Jaen, J., & Catala, A. (2017). Assessing machine learning classifiers for the detection of animals' behavior using depth-based tracking. *Expert Systems with Applications*, *86*, 235-246.

[5] Gerencsér, L., Vásárhelyi, G., Nagy, M., Vicsek, T., & Miklósi, A. (2013). Identification of behaviour in freely moving dogs (Canis familiaris) using inertial sensors. *PloS one*, *8*(10), e77814.

[6] Jenkins, G. J., Hakim, C. H., Yang, N. N., Yao, G., & Duan, D. (2018). Automatic characterization of stride parameters in canines with a single wearable inertial sensor. *Plos one*, *13*(6), e0198893.

[7] Winters, M., Brugarolas, R., Majikes, J., Mealin, S., Yuschak, S., Sherman, B. L., ... & Roberts, D. (2015, November). Knowledge engineering for unsupervised canine posture detection from IMU data. In *Proceedings of the 12th International Conference on Advances in Computer Entertainment Technology* (pp. 1-8).

[8] Le Roux, S. P., Marias, J., Wolhuter, R., & Niesler, T. (2017). Animal-borne behaviour classification for sheep (Dohne Merino) and Rhinoceros (Ceratotherium

simum and Diceros bicornis). *Animal Biotelemetry*, *5*(1), 1-13.

[9] Kaler, J., Mitsch, J., Vázquez-Diosdado, J. A., Bollard, N., Dottorini, T., & Ellis, K. A. (2020). Automated detection of lameness in sheep using machine learning approaches: novel insights into behavioural differences among lame and non-lame sheep. *Royal Society open science*, *7*(1), 190824.

[10] Zhang, T., Karg, M., Lin, J. F. S., Kulic, D., & Venture, G. (2013, August). Imu based single stride identification of humans. In *2013 IEEE RO-MAN* (pp. 220-225). IEEE.

[11] Baghdadi, A. (2019). *Application of Inertial Measurement Unit (IMU) in Advanced Human Health and Safety Surveillance: A Data Fusion and Machine Learning Approach* (Doctoral dissertation, State University of New York at Buffalo).