Exploring Ocean Animal Trajectory Pattern via Deep Learning

Thesis by

Su Wang

In Partial Fulfillment of the Requirements

For the Degree of

Masters of Science

King Abdullah University of Science and Technology, Thuwal,
Kingdom of Saudi Arabia

March, 2016
The thesis of Su Wang is approved by the examination committee

Committee Chairperson: Prof. Xiangliang Zhang

Committee Member: Prof. Xin Gao

Committee Member: Prof. Mikhail Moshkov
ABSTRACT

Exploring Ocean Animal Trajectory Pattern via
Deep Learning

Su Wang

We trained a combined deep convolutional neural network to predict seals’ age (3 categories) and gender (2 categories). The entire dataset contains 110 seals with around 489 thousand location records. Most records are continuous and measured in a certain step. We created five convolutional layers for feature representation and established two fully connected structure as age’s and gender’s classifier, respectively. Each classifier consists of three fully connected layers. Treating seals’ latitude and longitude as input, entire deep learning network, which includes 780,000 neurons and 2,097,000 parameters, can reach to 70.72% accuracy rate for predicting seals’ age and simultaneously achieve 79.95% for gender estimation.
ACKNOWLEDGEMENTS

Many thanks to Prof. Xiangliang Zhang for her supervision and support in entire master thesis. Also, I would like to appreciate Dr. Chengbin Peng for his valuable suggestions on model optimization.
# TABLE OF CONTENTS

Examination Committee Approval 2  
Copyright 3  
Abstract 4  
Acknowledgements 5  
List of Figures 8  
List of Tables 9  

1 Introduction 10  
2 Related Work 14  
2.1 Animal Trajectory Analysis 14  
2.2 Deep Learning 15  

3 Data Description 17  
3.1 Statistics information about the raw data 17  
3.2 Preliminary Study 18  

4 Deep Network for Animal Trajectory Analysis 22  
4.1 Neural Network Structure 22  
4.2 Convolutional Neural Network 25  
4.2.1 Convolutional Layer 25  
4.2.2 ReLU Layer 26  
4.2.3 Pooling Layer 26  
4.2.4 Fully Connected Layer 27  
4.3 Overall Architecture 28  
4.4 Introduction of Caffe 30
# LIST OF FIGURES

3.1 Main area where trajectory located ................................................. 19
3.2 Trip statistics ............................................................................. 19
3.3 The histogram of distance per step ................................................. 19
3.4 Typical trajectory of a male seal ................................................... 19
3.5 Typical trajectory of a female seal .................................................. 19

4.1 Neuron: NN’s basic unit ................................................................. 23
4.2 Artificial Neural Network ............................................................... 23
4.3 Overall CNN Architecture ............................................................. 29

5.1 Example of input .......................................................................... 33
5.2 The Loss of age classification ......................................................... 34
5.3 The Loss of gender classification .................................................... 36
5.4 Comparison of Loss on dropouts .................................................... 38
5.5 Comparison of accuracy rates on dropouts ...................................... 38
5.6 The output of fully connected layer ................................................. 40
5.7 Kernels visualization of conv1 ......................................................... 40
5.8 The output of conv1 ................................................................. 41
LIST OF TABLES

3.1 Age classification via classical algorithms .................................................. 20
3.2 Gender classification via classical algorithms ............................................. 20

5.1 Accuracy for age classification ............................................................... 35
5.2 Accuracy for gender classification .......................................................... 35
5.3 Comparison of the age and gender classification ....................................... 36
5.4 Accuracy for age and gender classification ............................................. 42
Chapter 1

Introduction

As a hot machine learning technique, the deep neural network has been popularly applied to the aspect of artificial intelligence. One of the most representative work is the usage of Convolutional Neural Network (CNN) by Hinton in 2012 [1]. It dramatically improves the performance of image recognition in the competition of ImageNet 2012. Hinton’s model substantially exceeded the state-of-the-art at that time. As a result, deep neural network got plenty of attention from researchers.

With rapid development during recent years, deep learning approach has been playing significant roles in many sub-aspects of artificial intelligence, such as image recognition, speech recognition, natural language processing (NLP), machine translation and so on. Regarding image recognition, Alex Net [1] which essentially is a multilayer convolutional neural network is the most representative work. It not only achieved a breakthrough precision in one of the most challenging image competition but also initiated the utilization of deep neural network for computer vision issues. Related work proves deep network is a stable and robust system that resists natural noise such as illumination, rotation and image background. Contributing to this property, the sub-direction of image recognition, facial recognition also achieved dramatic improvement [2][3]. In addition, deep learning refreshed the state-of-the-art records in speech recognition [4][5] and machine translation [6]. And in industry, Google exhibited a novel artificial intelligence system, AlphaGo [7], which combined Monte
Carlo simulation with two deep networks. Beating Go World Champion, AlphaGo made researchers with an indelible impression. The great success of AlphaGo reveals that deep network technique has excellent prospects that adapt to more area.

One significant characteristic of the deep system is learning representation. After constructing a deep network, computational devices can automatically detect effective feature maps on raw data. More specifically, it is convolutional units, one important composition of deep network that undertakes feature detection. These units, (also call them kernels or filters,) usually locate at multiple convolutional layers and be response to extract different feature. These features represent various physical meaning. For example, some units focus on geometry pattern while some extract color information. Eventually, these learned convolutional pattern that get high weight scores during the step of back propagation [8][9] are retained and be comprised of the final feature map. Compared with traditional methods, deep learning reflects a powerful capacity of automatically accomplishing feature detection. On the other hand, deep learning exists limitations as well. One limitation is that training such a complex and delicate system requires a vast number of raw data and consumes a long training time and a large number of computing resources. That’s why deep learning did not become popular before 2006. However, in this big data era, data becomes rich, sufficient and easy to acquire. Also due to the rapid development of graphics processing units (GPUs), parallel computation dramatically reduces time cost. As a result, deep learning technique has been an unprecedented growth since 2006.

Although deep learning approach was popular in many aspects, it hasn’t been applied for animal movement analysis. Currently, most approaches studying animal movement and biological trajectory mainly rely on probabilistic representation and mathematical modeling. For example, R. M. Kareiva proposed correlated random walks [10], treating animal ocean path as an entirely random walk. However, random walks are based on two theoretical assumptions, that’s the direction of next step
obeys a particular distribution (e.g. uniform distribution), and the length of next step follows another distribution (e.g. Gaussian distribution). However, our experiment demonstrated that seals’ trajectories have no apparent probabilistic pattern, regarding their angle and speed. The result of our preliminary trajectory analysis indicated that seals’ speed and angle highly depend on their real-time situations, e.g. hunting and avoiding predators usually lead to rapid changes, no matter speed or angle, while a stable and long-time linear moving are likely for the purpose of migration. Hence, the speed or angle distribution regarding seal’s complete trajectory is hard to estimate. The second approach is random process analysis. One outstanding example is the State-Space Model [11]. This model took a wide variety of patterns into account, as a result effectively improved the performance. But due to its complexity and poor model generalization, this model had weak performance on other animal’s data set. In addition, Deborah Austin made use of seals’ trajectories to implement gender classification via correlated walk model [12]. However, since the landform and sampling period vary a lot, this model did not work well on our data.

In this paper, we focus on how to apply deep learning approach on ocean animal trajectory. The feasibility of using deep network relies on two aspects. First, raw dataset provides sufficient records of seals’ trajectories. Totally, there are around 489 thousand geographic records. Each record contains seal’s real-time latitude and longitude. All records retain inter-category consistency during the entire observation period, as they are continuous and measured in a fixed time. Therefore, we firmly believe that these trajectories are sufficient and provide potential intra-category patterns. Second, deep neural network has an adequate capacity of feature detection, particularly of geometric pattern. Our preliminary experiment indicates that seal’s trajectory is likely to exist some geometric patterns that vary between different categories. Due to individual physiological state, female seals usually stay in habitat and feed for its child, while the male is more likely to appear sports state that keeps
moving with long distance for the purpose of patrol its territory or migration. Without a set of complex probability distribution hypothesis, a deep neural network is a suitable approach that automatically detects useful patterns.

The thesis is organized as follows. Next chapter introduces related work on animal trajectory analysis and deep learning. We would like to introduce several tricks popularly applied to model optimization, meanwhile to discuss their performance on our task. Chapter 3 discusses the background of raw data and preliminary study results. Chapter 4 mainly presents the methodology of convolutional neural network for analyzing animal trajectory data and the deep learning software. Chapter 5 shows experimental results. Chapter 6 gives conclusion and discusses future work.
Chapter 2

Related Work

2.1 Animal Trajectory Analysis

There is a large number of work on applications of machine learning algorithms to animal trajectory analysis. In the chapter of Introduction, we have introduced correlated random walks [10], which regards animal ocean path as an entirely random walk. We also mentioned State-Space Model [11], which essentially is one kind of random process analysis. Moreover, we discussed correlated walk model [12], which was closely related to our work. Because [12] implemented gender classification on seals’ trajectories. However, as we mentioned previously, this model does not work well on our data.

More recently, researchers have proposed other methods for animal trajectory analysis. Langrock [13] introduced hidden semi-Markov models (semi-HMMs), which consider some extensions of HMMs for animal movement models and some flexible state transition models, taking individual random effects into account. This model can provide valuable insights on animal’s behavioral states. However, the HMM-based approach also exists two limitations. First, it requires regularly timed observations of animal trajectory. If the time interval between successive observations varies, some critical assumptions of HMMs, e.g. each hidden state only depends on its previous state and all hidden state space which consists of N possible values is modeled as
a categorical distribution, would make no sense. In addition, HMMs have limited capacity that discovers meaningful patterns among several categories. In our paper, one of the primary goals is to look for some discriminative patterns that explain the difference between the different gender or the different age. Therefore, HMMs approach does not match our requirement.

2.2 Deep Learning

Deep learning approaches have been playing significant roles in many aspects. We have introduced Alex Net [1] which achieved a breakthrough accuracy in the competition of ImageNet 2012. For facial recognition, Wang [2][3] learned a set of deep hidden identity feature through deep learning. These features were extracted from various face regions to form complementary and overcomplete representations. As a result, facial verification accuracy achieved dramatic improvement.

Some detailed approaches of training CNN are worth to discussing here. Ioffe [14] proposed a phenomenon as internal covariate shift, where the distribution of each layer's inputs changes during forward passing. As a consequence, stochastic gradient descent required lower learning rates and more iterations, such that overall training time substantially increases. To solve this problem, Ioffe discussed batch normalization is an efficient and necessary way during the training phase. Our experiment also supports Ioffe’s conclusion. Without batch normalization, our model requires nearly ten times iterations than the model applies batch normalization. Meanwhile, the training phase indicates that the learning rate of a non-batch-normalization model must be a small value (e.g. 0.00001 or less). Otherwise, this model cannot get the optimal solution. Therefore, in this paper, we applied batch normalization in normalization layers, not only to resist internal covariate shift, but also to speed up the training phase. Additionally, Hinton [15] proposed a novel approach to resist overfit-
ting, called “Drop-out”. It is a novel form of regularization methods essentially. Our experiment demonstrates that drop-out makes the good effort on resisting overfitting. The introduction of drop-out is in chapter 4.2.4.
Chapter 3

Data Description

The data we analyze are 110 seals’ trajectory with more than 489 thousand location records\textsuperscript{1}. These trajectories data is a set of samples of location information from the continuous movement with the corresponding timestamp. Due to the unpredictable environment, individual path doesn’t have the same time range in general and the time interval between two sample points is not necessary the same. Our goal is to extract geometry pattern from trajectories and then make use of learned features to classify their gender and age. Consider that these seals’ trajectories have different length, starting point as well as starting time, it’s difficult to compare these trajectories directly and achieve efficient classification. Thus, we try to use deep learning network to learn from these trajectories, further to discover the potential pattern of seals’ movement.

3.1 Statistics information about the raw data

We have 31 female seals’ data and 79 male’s data. Among these females seals, 2 of them are subadults and 29 of them are adults. Among these male seals, none

\textsuperscript{1}Data was sourced from the Integrated Marine Observing System (IMOS) - IMOS is a national collaborative research infrastructure, supported by Australian Government. Other organizations are also involved in data collection, such as Australian Antarctic Division, Macquarie University and the University of Tasmania. We acknowledge the data custodians, owners and primary users for sharing the data. They are Rob Harcourt, Mark Hindell, Dan Costa and Michele Thums and Clive McMahon, Victor Eguluz and Carlos Duarte.
of them are adult, 66 of them are subadults, and 13 of them are juveniles. Totally speaking, there are two categories of gender task, 31 female and 79 male respectively. For age recognition, the entire data set is split into three groups, including 29 adults, 68 subadults, and 13 juveniles.

We have 489 thousand records, on average 4400 points for each seal. The area these trajectory located is shown in Figure 3.1. The distribution of daily moving distance appears in Figure 3.2. The location information is collected each hour, and no missing data. Detailed speed analysis is shown in Figure 3.3, where X-axis is one step distance KM/hour, and Y-axis represents corresponding frequency. Since all data are sampled at a certain step, each step length is equivalent to seals’ speed. As Figure 3.3 shown, seals keep their speed around 0 to 10 KM/h. Sometimes they may accelerate to a high speed. The fastest speed reaches to 52 KM/h.

Two typical trajectories are given standing for different genders. The typical path of the male is shown in Figure 3.4. It’s almost linear and goes in only one direction. The typical female path is completely non-linear and has many destinations during the travel, which is shown in Figure 3.5. However, when observing entire data set, we find that a small part of seals do not support above conclusion. These violated trajectories are likely to affect task’s accuracy rates negatively. More details will be discussed in Chapter 5.

3.2 Preliminary Study

A preliminary study is conducted for gender and age classification, respectively. We define a set of features for describing a trajectory, such as Pareto random walk for speed, von Mise distribution for angle change between two successive steps and other geographic information, e.g. mean of speed and variance, overall moving distance, the length between the original point and the end point and so on. These features
Figure 3.1: Main area where trajectory located

Trip Statistics

Figure 3.2: Trip statistics

Distribution of Distance

Figure 3.3: The histogram of distance per step

Figure 3.4: Typical trajectory of a male seal

Figure 3.5: Typical trajectory of a female seal
Table 3.1: Age classification via classical algorithms

<table>
<thead>
<tr>
<th></th>
<th>ACC (%)</th>
<th>Precision</th>
<th>Recall</th>
<th>F-score</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVM</td>
<td>77.5</td>
<td>0.82</td>
<td>0.87</td>
<td>0.84</td>
</tr>
<tr>
<td>Naive Bayes</td>
<td>74.7</td>
<td>0.75</td>
<td>0.89</td>
<td>0.80</td>
</tr>
<tr>
<td>KNN</td>
<td>79.2</td>
<td>0.86</td>
<td>0.85</td>
<td>0.85</td>
</tr>
<tr>
<td>DT</td>
<td>71.1</td>
<td>0.81</td>
<td>0.90</td>
<td>0.85</td>
</tr>
</tbody>
</table>

Table 3.2: Gender classification via classical algorithms

<table>
<thead>
<tr>
<th></th>
<th>ACC (%)</th>
<th>Precision</th>
<th>Recall</th>
<th>F-score</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVM</td>
<td>77.5</td>
<td>0.82</td>
<td>0.87</td>
<td>0.84</td>
</tr>
<tr>
<td>Naive Bayes</td>
<td>74.7</td>
<td>0.75</td>
<td>0.89</td>
<td>0.80</td>
</tr>
<tr>
<td>KNN</td>
<td>79.2</td>
<td>0.86</td>
<td>0.85</td>
<td>0.85</td>
</tr>
<tr>
<td>DT</td>
<td>71.1</td>
<td>0.81</td>
<td>0.90</td>
<td>0.85</td>
</tr>
</tbody>
</table>

are based on our observation and understanding of trajectories. They are expected to represent each trajectory best as a vector in 12-dimension space. Several widely used classification methods are employed, such as Support Vector Machine (SVM), K-NN, Naive Bayes and Decision Tree. Overall evaluation of these methods is conducted in a 5-fold cross-validation. Entire results, including accuracy rate, precision, recall as well as f-score are given in Table 3.1 and Table 3.2.
SVM is the first model we applied, and it can reach to 70.0% accuracy for the age and 77.5% for the gender. The second method we utilized is Naive Bayes. Our preliminary study indicates that the correlation coefficient among 12-dimensional feature is quite low, closing to zero. Hence, we reasonable assume that entire feature is independent of each other. Theoretically, Naive Bayes would make sense in this situation. Our experimental results indicate that it indeed works well on gender task, as it got 74.7% accuracy. However, Naive Bayes provides a weak prediction on the age task, whose accuracy is only 51.2%. As we know, the quality of feature strongly determines how much accuracy the task can get. In this case, the 12-dimensional feature we extracted is likely to provide meaningful representation on the difference between male and female. However, it is weak to distinguish the subadults and the juveniles (as most of the subadults and all juveniles are male). Therefore, Naive Bayes degrades in age task. In our thesis, we attempt to use a deep network to learn representations from the raw trajectories inputs. We expect the feature learn from deep network has stronger representation to classify the subadults and the juveniles.

The third approach is K-NN, which reaches to 67.0% accuracy on the age and 79.2% on the gender. KNN has the best performance on the gender task. However, as a type of so-called “lazy learning”, K-NN is extremely sensitive to the local structure of the input/feature. We try to extract some feature that has the substantial difference between adults, subadults, and juveniles. But actually, it is tough to find such perfect feature. Thus, feature extraction usually becomes a bottleneck to limit the performance of K-NN.

The last method we attempt is Decision Tree. We observed that entire data samples have no substantial tendency that males seals became a cluster, and females seals became the other cluster. All points of males are split into several parts, either for all females. Therefore, Decision Tree does not work well on two tasks. It reaches 61.0% on age classification and 71.1% on gender classification.
Chapter 4

Deep Network for Animal Trajectory Analysis

4.1 Neural Network Strucutre

Traditional Neural Network contains three kinds of layers, an input layer, hidden layer, and output layer. Input and output layer are single-layer while hidden layer can have a multilayer structure. The number of hidden layers depends on the size of data set. Commonly, a simple NN contains 2 – 4 hidden layers, while a complex network may reach to more than 10 hidden layers. Each layer includes plenty of nodes, which regards as the fundamental unit of the neural network, we called it, Neuron.

As Fig 4.1 and Fig 4.2 shown, each neuron connects with other neurons from adjacent layers. Its input comes from the output of the previous layer (except input layer, where the input is raw data). As the same rule, the output of one neuron will be fed to next layer as input. As a fundamental unit, neuron executes a set of linear and nonlinear computation as follows:

\[ y^{(t)} = f(\sum_{i=0}^{m} (w_{ji} \ast x^{(t)}_i)) \]  

(4.1)
Where $x^{(t)}_i$ is input vector in $t$ th iteration, and corresponding output we denote as $y^{(t)}$. $w_{ji}$ is a weight value from neuron $i$ to neuron $j$. And function $f()$, namely activation function, implements nonlinear computation. Some useful activation functions contain sigmoid function, tanh function and so on. Usually, we call this network as artificial neural network (ANN) and its structure as Full Connected Structure, to distinguish other new deep systems.

After creating a model structure, the next step is to train NN’s parameter. Raw data will go through the entire system and make output units produce final results, which will compare to the desired output results by taking a loss function. Here we choose Softmax function to determine the labels of the data sample. Assuming the data set contains $K$ categories, the definition of Softmax function is:

$$
    h_\omega(x^{(i)}) = \begin{bmatrix}
    p(y^{(i)} = 1|x^{(i)}; \omega) \\
    p(y^{(i)} = 2|x^{(i)}; \omega) \\
    \vdots \\
    p(y^{(i)} = k|x^{(i)}; \omega)
    \end{bmatrix} = \frac{1}{\sum_{j=1}^{k} \exp(\omega_j^T x^{(i)})} \begin{bmatrix}
    \exp(\omega_1^T x^{(i)}) \\
    \exp(\omega_2^T x^{(i)}) \\
    \vdots \\
    \exp(\omega_k^T x^{(i)})
    \end{bmatrix} \tag{4.2}
$$

Where $x^{(i)}$ is $i$ th sample vector and $y^{(i)}$ is its predicted label. The predicted probability for the $k$ th class denotes as $p(y^{(i)} = k|x^{(i)}; \omega)$. It is calculated by the composition
of K inner products of $\omega$ and $x$. And exponential normalization ensures the sum of overall K predicted probabilities is 1. Given Softmax function, corresponding loss function, namely “Softmaxloss”, is designed as follows:

$$J(\omega) = -\frac{1}{m} \left[ \sum_{i=1}^{m} \sum_{j=1}^{k} I\{y^{(i)} = j\} \log \frac{\exp(\omega_j^T x^{(i)})}{\sum_{l=1}^{k} \exp(\omega_l^T x^{(i)})} \right]$$  \hspace{1cm} (4.3)

Where the function $I(\cdot)$ defines as follows:

$$I\{y^{(i)} = k\} = \begin{cases} 
1 & \text{if true label of } y^{(i)} \text{ is } k, \\
0 & \text{others.} \end{cases}$$  \hspace{1cm} (4.4)

The concept of “Softmaxloss” is straightforward. Given a data sample $x^{(i)}$ with the true label $k$, loss function only focuses on the predicted probability of $k$th class. For the predicted probabilities of other classes, since the results of function $I(\cdot)$ are 0, their loss are equal to 0.

Essentially, the NN’s training phase is to finely tune model’s parameter, such that its loss would decrease to a global minimum. Some practical optimization methods have been proposed, e.g. D.E. Rumelhart proposed Back Propagation (BP) \cite{8} \cite{9} that iteratively applies Stochastic Gradient Descent (SGD) method to search for the global minimum. In this paper, we adopt Stochastic Gradient Descent and Back Propagation to train our model. Based on the equation 4.3, the gradient of Softmaxloss is:

$$\nabla_{\omega} J(\omega) = -\frac{1}{m} \sum_{i=1}^{m} [x^{(i)}(I\{y^{(i)} = j\} - p(y^{(i)} = j|x^{(i)}; \omega))]$$  \hspace{1cm} (4.5)

To some degree, NN highly simulates the neural circuits of the human brain, constructing a cascaded structure similar to human brain structure. However, ANN did not implement high achievement for a long time. One drawback of NN is the incomplete feature representation. In other words, ANN’s parameter lacks physical
meaning explanation in some extends. Moreover, due to NN’s complex structure, its interpretability is weak. In many situations, we have to do pre-processing on raw data, extracting feature matrix before classifying via NN. To understand the meaning of parameters of NN, also to improve performance, these pre-processing are necessary and important.

4.2 Convolutional Neural Network

In 2012 Imagenet competition, Hinton proposed a novel neural network called Alex Net[1]. This model achieved top recognition accuracy rates and much better than the second place. Alex Net is essentially a convolutional neural network (CNN), which has been turned out to be good at extracting local structural feature. Therefore, CNN is proper to apply for image recognition, speech recognition. Different with ANN, CNN has more complex structure. CNN splits hidden layer into more specific layers. These layers mainly contain convolutional layer, ReLU layer, pooling layer and fully connected layer.

4.2.1 Convolutional Layer

As a material composition of CNN, convolutional layer almost took on all task that extracts raw data’s local structural feature. The basic unit of a convolutional layer is a set of learnable convolutional filters, usually we call it as “kernel”, or name entire filter set as a filters bank. The size of one filter is small, typically 3 pixels * 3 pixels or 5 * 5 (11 * 11 is optional in the first convolutional layer). According to the definition of convolution, each filter computes the dot product of the entries of a filter and raw image data, producing a score indicating the strength of the corresponding feature (e.g. a filter aims to extract edge information. For one local region where pixels do vary substantially, a filter will produce a high score indicating edge exists here. In
contrast, dot producing a region in which pixels are almost uniform has a low rating.)

After training phase, the network can find a set of filters with high scores as valid feature representation at some spatial position in the input.

4.2.2 ReLU Layer

Rectified Linear Units is essentially an activation function, given as follows:

\[ f(x) = \max(0, x) \] (4.6)

Traditional ANN usually utilizes \textit{sigmoid} function or \textit{tanh} function as activation function. Due to a relatively simple structure, computation cost on ANN is not so remarkable. However, since CNN is more complicated than ANN, traditional activation functions require high computation cost. As a result, training phase usually spends several weeks even several months.

Comparing with above activation functions, ReLU function is usually much faster than \textit{tanh} or \textit{sigmoid} units. As Hinton proposed on Alex Net [1], \textit{tanh} function spent four times of epochs than ReLU, when they reach to 25% training error on the CIFAR-10 dataset. It is the ReLU function that breaks computation limitation on hardware devices (either because of GPUs techniques) and makes CNN be fast and practical.

4.2.3 Pooling Layer

As we know, image, as one kind of digital data storage and visualization method, usually contains plenty of redundant information. When human takes eyes on a picture, our brain typically observes the local representative pattern (e.g. color difference, texture feature). The same with CNN, when raw data goes through the convolutional layer and produces new neuron volume, this new feature cube is too huge (e.g. Alex
Net outputs $55 \times 55 \times 96$ neurons after first convolutional layer). Therefore, pooling layer is responsible for further extracting significant information and compressing neural volume simultaneously.

Typical pooling operator contains max pooling and average pooling. For 5 pixels $\times$ 5 pixels pooling unit, what max pooling does is to scan a $5 \times 5$ local region of neuron volume, and represent this area via its maximum value. As a result, 25 parameters compress to a single parameter, actually avoiding entire CNN being complicated and redundant. Similar to Max pooling, Average pooling detects the mean of a local region, which has a good effect on speech recognition.

Traditionally, a pooling unit does not overlap with its adjacent units. Here we introduce the concept of “strike”. The definition of a strike is the distance of two central points in adjacent units. For example, for $5 \times 5$ unit, non-overlapping pooling set strike being 5. If the strike is configured being less than 5, corresponding pooling units would be overlapped with adjacent units. We call such pooling method where the strike is less than the length of a pooling unit as overlapping pooling. Alex Net [1] adequately demonstrated that overlapping pooling has an excellent property that can slightly resist overfitting. Therefore, our model utilizes overlapping pooling of size $3 \times 3$ with strike 2.

### 4.2.4 Fully Connected Layer

The concept of the fully connected layer is similar to ANN. Since convolutional layer, ReLu layer and pooling layer manage to detecting significant patterns, the remaining work is to train a classifier and accomplish classification. Thus, the fully connected structure can be regarded as a classifier. For entire CNN, the first half part, which consists of multiple convolutional layers, ReLu layers and pooling layers, is a feature representation phase. And the left half part, namely multiple fully connected layers, is a classification phase.
In particular, here we discuss an efficient training technique, called “drop-out” [15]. As we know, training a deep learning network is an incredibly time-consuming work, such that some known trikes for resisting overfitting now are unavailable. To solve this issue, Hinton proposed Drop-out technique in 2012. The primary concept of Drop-out is the weight of neurons at fully connected layers are set to be zero with a fixed probability, typically is 0.5. According to this rule, half neurons at fully connected layers, whose weight becomes zero, won’t contribute to loss computation during the forward pass, also do not update the corresponding parameters at the step of back propagation. For each forward and back pass, raw data are trained by a random partial deep network. For the entire training phase, drop-out plays a role that integrates multiple partial deep systems. Drop-out technique is similar to the concept of boosting that combines multiple weak classifiers. As a result, Drop-out contributes CNN to effectively overcoming overfitting. Also, it was reported to be able to improve slightly accuracy rate. In experimental section, some related results illustrate that Drop-out does make a real effort on resisting overfitting.

4.3 Overall Architecture

The overall architecture of our convolutional neural network is shown in Fig 4.3. Since there are two tasks we should do, age classification and gender classification, we construct two fully connected phases as classifiers for the two task respectively. These two phases share parameters from feature representation phase.

The input of CNN is a 227 * 227 * 3 (RGB channels) image. The first convolutional layer (denoted as “conv1”) filters the input with 96 kernels. Each kernel is an 11 * 11 * 3 convolutional operator with a strike of 4. As a result, the output of conv1 totally contains 55 * 55 * 96 neurons. A ReLU layer (ReLU1) follows after conv1, then is pooling layer (pool1) and normalization layer (norm1). In pool1, we adopt
overlapped pooling that the size of each window is 3 * 3 with a strike of 2. After passing pool1, the number of neurons in one layer further reduces to 27 * 27 * 96. The next layer is the second convolutional layer (conv2). Similar to conv1, conv2 consists of 256 kernels of size 5 * 5 * 48 with a strike of 1. Overall conv2 contains 27 * 27 * 256 neurons, then is ReLU2, Pool2 and norm2.

Several important details have to mention here. Starting from conv2, the strike of all remaining convolutional layers is set to 1, not 4. The definition of the strike is introduced at Chapter 4.2.3 “Pooling Layer”. In our model, only the strike in conv1 is set to 4, while strikes in other convolutional layers are fixed to 1. The reason for doing it is to simplify our model, as a 227 * 227 * 3 image has too many pixels as well as plenty of redundant information. A strike with a large value can eliminate internal noise information. On the other hand, to retain active feature, other convolutional layers should choose a strike of small value (1 as default). Moreover, due to the purpose of resisting overfitting, overall CNN utilizes overlapping pooling. The size of each pooling window is 3 * 3 with a strike of 2. Thus, pool2 has 13 * 13 * 256 neurons. The layer after pool2 is conv3, conv4. They do not bind with pooling and normalization layers. Conv3 has 384 kernels of size 3 * 3 * 256. Conv4 has
384 kernels of size 3 * 3 * 192. The last two layers of feature representation phase are conv5 and pool5, respectively. Conv5 has 256 kernels of size 3 * 3 * 192. Pool5 follows conv5 and contains totally 6 * 6 * 256 neurons, which are final output of entire feature representation phase. To sum, overall feature representation phase contains five convolutional layers with three pooling layers (pool1, pool2, pool5) as well as two normalization layers (norm1, norm2). For each trajectory image, a 9,216-dimensional vector is generated at the end of feature representation phase.

Next, we construct two classification phases, one for age and the other for gender. Both of them have the same structure. Specifically, each phase consists of three fully connected layers. The first layer contains 4096 neurons while the second layer has 512 neurons and the third layer is output layer. The phase of age has three output nodes representing adult, subadult and juveniles, respectively. The phase of gender outputs two nodes representing male or female. Each fully connected layer shares weights with its adjacent layers.

4.4 Introduction of Caffe

Caffe [16] is a deep learning framework developed by the Berkeley Vision and Learning Center (BVLC) and by community contributors. Dr. Yangqing Jia created the project during his Ph.D. at UC Berkeley. As an impressive open resource, Caffe implements deep learning models and provides various demos that help users to master deep network. Also, Caffe offers two alternations of hardware mode at the step of training. One mode is graphics processing units (GPUs), supporting NVIDIA chips and CUDA compiling environment. Also, Caffe provides CPUs interface. However, CPUs spend much more time than working on GPUs mode, so it is not a recommended way. Last but not the most important, Caffe supports plenty of interface on Python and MATLAB. It is very convenient for researchers observing each layer’s
input and output. In this paper, entire experimental work was implemented via Caffe.
Chapter 5

Experiment Result

5.1 Data Augmentation

The entire dataset contains 110 seals’ trajectory totally with 489 thousand location records. Considering that the input of CNN is in the form of images, here we generate 110 images, to conveniently import trajectories to CNN. Each image represents an individual path. One example is given in Fig 5.1. All images have a uniform size of 256 * 256 pixels. The range of a picture describes overall scope of one seal’s activities. White pixels indicate seals’ location records while black pixel is background.

The first way used for data augmentation is image rotation and horizontal reflection. Specifically, we manually rotate images with 10 different angles, which are $\frac{\pi}{12}$, $\frac{\pi}{6}$, $\frac{\pi}{4}$, $\frac{\pi}{2}$, $\frac{3\pi}{4}$ for clockwise and counterclockwise. Totally, we extend the number of images by a factor of 12 (1 raw image + 10 rotating images + 1 horizontal reflection). The second way of data augmentation is random crops. As the size of CNN input layer is smaller than that of raw data, we have to extract randomly 227 * 227 patches from each original image. Here we make use of Caffe [16] internal function to complete image crops. Eventually, the size of the training set increases by a factor of 1200. And for each testing image, we just extract five patches (4 corner patches and one center patch).

Except rotation, random crops and horizontal reflection, other optional ways for
First, a practical color jittering approach is HSV transform, e.g., raising all pixels’ saturation (S component of HSV) and value (V component of HSV) to power between 0.25 and 4, or adding a tiny offset to the hue (H component of HSV). However, our original image only contains two colors, black and white, so HSV transform does not enhance image quality. Thus, we do not adopt the color jittering approach in this paper. The second optional method is fancy PCA. The goal of fancy PCA is to “approximately capture an important property of nature images”. As [1] mentioned, using fancy PCA can reduce the top-1 error rate by over 1%. However, this method does not bring any improvement on our dataset. Considering extra computation cost of fancy PCA, we give up utilizing it.

5.2 Results

Our results on seal trajectory dataset are given in Table 5.1 and Table 5.2. Here we implement 5-fold cross-validation to train CNN. 88 seals randomly choose to be training set, while the rest 22 seals treat as the testing set. Our model takes the
parameters of Alex Net as initial values, and then finely tunes model parameters on our dataset. Cross-validation results indicate that overall accuracy rate of age task reaches to 70.72% (110 * 1200 images, three categories). Meanwhile, the mean of accuracy rate on gender task achieves to 79.95% (2 categories). For comparison, entire experimental accuracy rates on age and gender classification via CNN as well as classical algorithms list in Table 5.3. Regarding age classification, our deep system improves 1% accuracy compared with SVM, the best model among four classical models. On the other hand, regarding gender recognition, our accuracy rate is also better (0.75%) than K-NN, the best one. However, our deep model still exists limitations. As we can see, the f-score of the juveniles on age classification is very low. It means that our deep model still cannot give a correct prediction on the juveniles. Hence, the accuracy of age task is 9.23% lower than that of the gender. In the section of Qualitative Evaluations, we will discuss why the subadults and the juveniles are difficult to predict.
Table 5.1: Accuracy for age classification

<table>
<thead>
<tr>
<th>Age</th>
<th>ACC = 71.67%</th>
<th>Prec.</th>
<th>Recall</th>
<th>F-score</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>AD SUB JUV</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Age1</td>
<td></td>
<td>0.452</td>
<td>0.633</td>
<td>0.528</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.759</td>
<td>0.827</td>
<td>0.791</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.093</td>
<td>0.208</td>
<td>0.128</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Age2</td>
<td>ACC = 69.57%</td>
<td>0.500</td>
<td>0.571</td>
<td>0.533</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.688</td>
<td>0.846</td>
<td>0.759</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.167</td>
<td>0.333</td>
<td>0.222</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Age3</td>
<td>ACC = 71.43%</td>
<td>0.500</td>
<td>0.667</td>
<td>0.571</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.733</td>
<td>0.846</td>
<td>0.786</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Age4</td>
<td>ACC = 78.26%</td>
<td>0.556</td>
<td>0.833</td>
<td>0.667</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.800</td>
<td>0.857</td>
<td>0.828</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.250</td>
<td>0.333</td>
<td>0.286</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Age5</td>
<td>ACC = 62.68%</td>
<td>0.337</td>
<td>0.444</td>
<td>0.383</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.658</td>
<td>0.810</td>
<td>0.725</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.065</td>
<td>0.139</td>
<td>0.089</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Average</td>
<td>ACC = 70.72%</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>AD SUB JUV</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.537</td>
<td>0.645</td>
<td>0.777</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.777</td>
<td>0.643</td>
<td>0.145</td>
</tr>
</tbody>
</table>

Table 5.2: Accuracy for gender classification

<table>
<thead>
<tr>
<th>Gender</th>
<th>ACC(%)</th>
<th>Prec.</th>
<th>Recall</th>
<th>F-score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gender1</td>
<td>80.95</td>
<td>0.667</td>
<td>0.667</td>
<td>0.667</td>
</tr>
<tr>
<td>Gender2</td>
<td>81.82</td>
<td>0.667</td>
<td>0.667</td>
<td>0.667</td>
</tr>
<tr>
<td>Gender3</td>
<td>77.27</td>
<td>0.556</td>
<td>0.833</td>
<td>0.667</td>
</tr>
<tr>
<td>Gender4</td>
<td>86.96</td>
<td>0.833</td>
<td>0.714</td>
<td>0.769</td>
</tr>
<tr>
<td>Gender5</td>
<td>72.73</td>
<td>0.500</td>
<td>0.333</td>
<td>0.400</td>
</tr>
<tr>
<td>Average</td>
<td>79.95</td>
<td>0.645</td>
<td>0.643</td>
<td>0.634</td>
</tr>
<tr>
<td></td>
<td>CNN</td>
<td>SVM</td>
<td>Naive Bayes</td>
<td>K-NN</td>
</tr>
<tr>
<td>----------</td>
<td>-------</td>
<td>-------</td>
<td>-------------</td>
<td>-------</td>
</tr>
<tr>
<td>Age</td>
<td>70.72</td>
<td>70.0</td>
<td>51.2</td>
<td>67.0</td>
</tr>
<tr>
<td>Gender</td>
<td>79.95</td>
<td>77.5</td>
<td>74.7</td>
<td>79.2</td>
</tr>
</tbody>
</table>

Table 5.3: Comparison of the age and gender classification

Figure 5.3: The Loss of gender classification
More details show in Fig 5.2 for age and Fig 5.3, where the blue line is the loss of training and the red line is the loss of testing. X-axis is the number of iterations and Y-axis is loss value. Totally, age task runs 40 thousand iterations, and gender task accomplishes 50 thousand iterations. Regarding training curve, both of curves have a substantial decreasing tendency. Eventually, the training loss reduces to 0.001. On the other hand, gender’s curve looks slightly more smooth than age loss, such that gender’s precision is 5.8% better than age’s. Additionally, the initial learning rate is 0.001, and it reduces by a factor of 10 every 10 thousand iterations. As a result, we can observe that loss curve does decrease slightly in the position of 10,000 iterations. When iterations go on, such tendency is not so apparent. Based on the overall trend of the loss, we believe that it is proper to set learning rate between 0.001 and 0.0001. Moreover, we can observe that both testing curves have an increasing tendency. It illustrates that overfitting does occur during the training phase. In this experiment, the primary factor leading to overfitting is the limitation of the size of the dataset. We attempt to solve the overfitting problem by several methods.

5.2.1 Drop-out

To resist overfitting, we applied “drop-out” technique in each fully connected layer. Fig 5.4 and Fig 5.5 demonstrate that drop-out plays a significant role on overcoming overfitting. Specifically, Fig 5.4 illustrates the tendency of testing loss curve during the training phase. As we can see, drop-out does make practical constraint on overfitting. Under the effort of drop-out, overall loss value of blue line is around 2. For comparison, the loss of red line (no drop-out) exceeds 4. Therefore, drop-out can help our model to resist overfitting. The testing accuracy rates in Fig 5.5 also support our conclusion, where using drop-out improves nearly 10% accuracy rate.
Figure 5.4: Comparison of Loss on dropouts

Figure 5.5: Comparison of accuracy rates on dropouts
5.2.2 Kernels

In this section, we evaluate several active kernels that can potentially be helpful to our task. Fig 5.6 shows the histogram of the output in the first fully connected layer. There are totally 4096 neurons. The top figure shows overall value of the output. All negative output sets to 0, because of ReLU. As we can see, more than 3,500 neurons generate 0 as output, which means these neurons are useless for our task. According to our analysis, there are 496 neurons in our model playing active capacity.

Additionally, we apply filter visualization to analyze filters maps. Fig 5.7 and Fig 5.8 respectively display 96 filters in conv1 and corresponding output learned during training phase. Filters in the first five rows aim to detect texture feature while the rest focus on color information. Meanwhile, we can observe that the output of color filters are entirely white figures, which indicate that trajectory problem does not exist color pattern. It is a reasonable conclusion because all images only contain two colors, black and white. On the other hand, most filters for texture detection make good efforts.

5.2.3 Qualitative Evaluations

This section analyzes potential factors leading to incorrect results. We try to use correlation coefficient to evaluate a pair of seals. Specifically, we extract the output of the second fully connected layer as seal’s feature vector and then calculate the correlation coefficient for each pair of seals. For those testing samples, especially those incorrectly classified samples, we manually check their correlation coefficient with all training samples. Table 5.4 shows some representative examples. Five seals from the testing set give in the second column, and they are seal #18, #1, #87, #89 and #37 respectively. Three training samples with top-3 correlation coefficients are listed in following columns. For each seal, their actual age and gender are attached in brackets. Here, “Ad” is short for adult, “Sub” is subadult, “Juv” is juvenile, “M” is
Figure 5.6: The output of fully connected layer

Figure 5.7: Kernels visualization of conv1
male and “F” means female. The row beginning with a string “Age” indicates the
 correlation of age, the row of “Gender” is the correlation of gender. A mark (✓)
 shows that CNN predicts a correct label for a testing seal while the other mark (×)
 indicates an incorrect prediction.

In Table 5.4, the first and second testing samples are two positive examples. As
we can see, seal #18 has the high correlation with seal #28, #26 and #29, also their
age and gender are the same. Thus, a correct prediction is reasonable. Similarly, seal
#1 is also correctly labeled as “Ad” and “F”.

#87 and #89 are two counterexamples concerning age classification. For seal
#87, top 3 correlated seals are adults and female, such that CNN predicts #87 to be
an adult as a mistake. We notice that, although the gender of #87 is different from
three training samples since its correlation coefficient of sex is low, CNN still regards
#87 as male. We suspect that one possible explanation for this case is one of three
female seals is a parent of #87, such that they have similar trajectories. Thus, the
parent-child relationship is a factor causing an incorrect prediction.

Let’s see the case of #89. For #89 seal, its top-3 correlated seals are subadult and
male. One reasonable conjecture is the age of #89 is closing to subadult. Although #89 is a juvenile, it is likely to display some characteristic matching a male subadult seal, for example, its speed, overall moving distance and so on. At least, their correlation coefficient on age and gender seems to support our conjecture. Thus, one more factor for age prediction is physiological characteristics of male juvenile seals and subadult seals have no substantial difference.

At last, the fifth case of seal #37 is an incorrect gender prediction case. We observe that most counterexamples on gender task (e.g. #37) have a common characteristic that its correlation coefficient is implicit concerning overall training sample. Taking #37 as an example, its correlation of top-3 samples is around 0.4, which is not so discriminative to predict its gender. For comparison, for a male seal (e.g. #18), its correlation usually exceed 0.8. In addition, the correlation between a male and female is less than 0.3 (e.g. #87). Thus, one possible inference for incorrect gender prediction is that those seals’ trajectories are different with seals in a regular group, such that their trajectories lack discriminative patterns, in terms of gender.

<table>
<thead>
<tr>
<th>Sample</th>
<th>#18(Sub, M)</th>
<th>#28(Sub, M)</th>
<th>#26(Sub, M)</th>
<th>#29(Sub, M)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Age:</td>
<td>✓</td>
<td>0.9430</td>
<td>0.8368</td>
<td>0.8028</td>
</tr>
<tr>
<td>Gender:</td>
<td>✓</td>
<td>0.9614</td>
<td>0.9130</td>
<td>0.8885</td>
</tr>
<tr>
<td>Sample</td>
<td>#1(Ad,F)</td>
<td>#109(Ad,F)</td>
<td>#11(Ad,F)</td>
<td>#9(Ad,F)</td>
</tr>
<tr>
<td>Age:</td>
<td>✓</td>
<td>0.8208</td>
<td>0.7992</td>
<td>0.7926</td>
</tr>
<tr>
<td>Gender:</td>
<td>✓</td>
<td>0.6626</td>
<td>0.5832</td>
<td>0.6254</td>
</tr>
<tr>
<td>Sample</td>
<td>#87(Juv, M)</td>
<td>#99(Ad, F)</td>
<td>#97(Ad, F)</td>
<td>#107(Ad, F)</td>
</tr>
<tr>
<td>Age:</td>
<td>×</td>
<td>0.6545</td>
<td>0.5991</td>
<td>0.5926</td>
</tr>
<tr>
<td>Gender:</td>
<td>✓</td>
<td>0.3072</td>
<td>0.0606</td>
<td>0.1102</td>
</tr>
<tr>
<td>Sample</td>
<td>#89(Juv, M)</td>
<td>#38(Sub, M)</td>
<td>#53(Sub, M)</td>
<td>#62(Sub, M)</td>
</tr>
<tr>
<td>Age:</td>
<td>×</td>
<td>0.7500</td>
<td>0.6890</td>
<td>0.6882</td>
</tr>
<tr>
<td>Gender:</td>
<td>✓</td>
<td>0.8052</td>
<td>0.7648</td>
<td>0.7843</td>
</tr>
<tr>
<td>Sample</td>
<td>#37(Sub, M)</td>
<td>#96(Ad, F)</td>
<td>#85(Ad, F)</td>
<td>#108(Ad, F)</td>
</tr>
<tr>
<td>Age:</td>
<td>✓</td>
<td>0.2720</td>
<td>0.3576</td>
<td>0.2248</td>
</tr>
<tr>
<td>Gender:</td>
<td>×</td>
<td>0.4106</td>
<td>0.3953</td>
<td>0.3782</td>
</tr>
</tbody>
</table>

Table 5.4: Accuracy for age and gender classification
Chapter 6

Conclusion and Future Work

6.1 Conclusion

In this paper, we trained a combined deep convolutional neural network to predict seals’ age (3 categories) and gender (2 categories). We created five convolutional layers for feature representation and established two fully connected structure as age’s and gender’s classifier respectively. Each classifier consists of three fully connected layers, and they share the parameters from feature representation phase. Entire deep learning network includes 780,000 neurons and 2,097,000 parameters.

We imported each seal’s trajectory to CNN in the format of an image. 5-fold cross-validation was adopted at the training step. Final experimental results indicated our model reached to 70.72% accuracy rate for predicting age and simultaneously achieved 79.95% for gender estimation. As a reference, we also provide the results of the preliminary study via classical algorithms. SVM provides the best performance (70.0%) regarding age classification, and K-NN has the best accuracy rate (79.2%) on gender classification. Our experimental results indicate that CNN gets better accuracy rate than SVM regarding age task and simultaneously makes a slight improvement on gender task. Qualitative analysis shows that the main factors leading to incorrect prediction come from three aspects. First, parent-child relationship, especially mother-child relationship tends to result in an erroneous prediction that
the child/subadult regarded as the mother/adult, and vice versa. The second factor is that physiological characteristics of male juvenile seals and subadult seals have no substantial difference. Our experiment indicates that several juveniles regarded as subadults. At last, for gender prediction, we observe several seals’ trajectories are different with regular seals’ path, such that their trajectories lost discriminative patterns concerning the gender.

6.2 Future Work

According to the requirement of the input of CNN, we have to describe seal’s trajectory in the format of an image. As one kind of two-dimensional data storages, the image breaks local structure of one-dimensional sequential data, particularly affecting its continuity in the time domain. Thus, our future work will focus on how to make directly use of original trajectory data as model input. We will explore the feasibility of Recurrent Neural Networks (RNN) on ocean animal trajectory.

Additionally, we will design novel deep models that make good efforts on solving previous three potential factors which lead to incorrect prediction. An alternative solution is to extend the size of the input of CNN, such that we can import two or more trajectories every round. The purpose is to extract more individual difference information, such that the novel deep system has the capacity that distinguishes parent and its child also discriminates the juvenile from subadults.
REFERENCES


