IMPROVING OBJECT RECOGNITION IN AERIAL IMAGE AND AMBULATORY ASSESSMENT ANALYSIS BY DEEP LEARNING

A Dissertation

Presented to

The Faculty of the Graduate School

At the University of Missouri

In Partial Fulfillment

Of the Requirements for the Degree

Doctor of Philosophy

By

PENG SUN

Dr. Yi Shang, Advisor

DEC 2019

The undersigned, appointed by the dean of the Graduate School, have examined the thesis entitled

IMPROVING OBJECT RECOGNITION IN AERIAL IMAGE AND AMBULATORY ASSESSMENT ANALYSIS BY DEEP LEARNING

Presented by Peng Sun

A candidate for the degree of

Doctor of Philosophy

And hereby certify that, in their opinion, it is worthy of acceptance.

Dr. Yi Shang

Dr. Dong Xu

Dr. Jianlin Cheng

Dr. Tim Trull

ACKNOWLEDGEMENTS

I would like to first thank my advisor, Dr. Yi Shang. He showed me how to do research in computer science, and he always supported and inspired me through the whole development of my dissertation. He provided me many research ideas and helped me produce solid works. Without his guidance, suggestions, and support this dissertation would not have been possible. His mentoring has been instrumental to my research productivity and efficiency, and his view about problem solving has influenced me to always have a relentless positive attitude in all situations.

I would like to thank my committee members Dr. Dong Xu, Dr. Jianlin Cheng, and Dr Tim Trull, for providing scientific guidance, encouragement and advice throughout my time as a student.

I also want to thank all the people in our research group, especially Zhaoyu Li, Guang Chen, Junlin Wang and Chao Fang, for their selfless help. It was fun to exchange ideas and thoughts with these great guys. I really enjoy the moment we discuss algorithm and machine learning knowledge with them!

Finally, I would like to thank all the people in my family. Thank my wife, Shuhui Jia. Without her support in every day, it is impossible for me to finish my PhD program! Thank my children, Jason Sun and Jenny Sun. You are the precious gifts of God in my life.

Thank my parents in law for taking care of us when our family needs help! Thank my parents for their support from the other side of the Earth.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

With the widespread usage of many different types of sensors in recent years, large amounts of diverse and complex sensor data have been generated and analyzed to extract useful information. This dissertation focuses on two types of data: aerial images and physiological sensor data. Several new methods have been proposed based on deep learning techniques to advance the state-of-the-art in analyzing these data. For aerial images, a new method for designing effective loss functions for training deep neural networks for object detection, called adaptive salience biased loss (ASBL), has been proposed. In addition, several state-of-the-art deep neural network models for object detection, including RetinaNet, UNet, Yolo, etc., have been adapted and modified to achieve improved performance on a new set of real-world aerial images for bird detection. For physiological sensor data, a deep learning method for alcohol usage detection, called Deep ADA, has been proposed to improve the automatic detection of alcohol usage (ADA) system, which is statistical data analysis pipeline to detect drinking episodes based on wearable physiological sensor data collected from real subjects.

Object detection in aerial images remains a challenging problem due to low image resolutions, complex backgrounds, and variations of sizes and orientations of objects in images. The new ASBL method has been designed for training deep neural network object detectors to achieve improved performance. ASBL can be implemented at the image level, which is called image-based ASBL, or at the anchor level, which is called anchor-based ASBL. The method computes saliency information of input images and anchors generated

by deep neural network object detectors, and weights different training examples and anchors differently based on their corresponding saliency measurements. It gives complex images and difficult targets more weights during training. In our experiments using two of the largest public benchmark data sets of aerial images, DOTA and NWPU VHR-10, the existing RetinaNet was trained using ASBL to generate an one-stage detector, ASBL-RetinaNet. ASBL-RetinaNet significantly outperformed the original RetinaNet by 3.61 mAP and 12.5 mAP on the two data sets, respectively. In addition, ASBL-RetinaNet outperformed 10 other state-of-art object detection methods.

To improve bird detection in aerial images, the Little Birds in Aerial Imagery (LBAI) dataset has been created from real-life aerial imagery data. LBAI contains various flocks and species of birds that are small in size, ranging from 10 by 10 pixel to 40 by 40 pixel. The dataset was labeled and further divided into two subsets, Easy and Hard, based on the complex of background. We have applied and improved some of the best deep learning models to LBAI images, including object detection techniques, such as YOLOv3, SSD, and RetinaNet, and semantic segmentation techniques, such as U-Net and Mask R-CNN. Experimental results show that RetinaNet performed the best overall, outperforming other models by 1.4 and 4.9 F1 scores on the Easy and Hard LBAI dataset, respectively.

For physiological sensor data analysis, Deep ADA has been developed to extract features from physiological signals and predict alcohol usage of real subjects in their daily lives. The features extracted are using Convolutional Neural Networks without any human intervention. A large amount of unlabeled data has been used in an unsupervised learning matter to improve the quality of learned features. The method outperformed traditional feature extraction methods by up to 19% higher accuracy.

1. INTRODUCTION

Nowadays, sensor data analysis has been researched by computer scientists for many years. Based on different types of sensor data, variety of data mining and pattern recognition algorithm are developed in computer science domain. However, due to the specific character of physiological data and remote sensor data, both domains are still challenging. For instance, the noisy information and low sample rate are included in the physiological data and make analyze much harder using traditional method. In addition, in terms of remote sensor data, the scale and angle of object varies much more than the conventional objects in natural images. With the power of machine learning and deep learning in recent year, analyze for physiological data and remote sensor data with good performance are much more promising. In this dissertation, based on the problem of physiological data and remote sensor data, different types of data mining techniques, like ADA, are proposed to explore the world of sensor data, and novel machine learning algorithm, like Deep ADA, and Adaptive Saliency Biased Loss (ASBL) has been proposed for each domain.

1.1 Object detection using deep learning in aerial images

In recent years, deep neural networks have achieved huge success in many areas of computer vision, such as image classification, object detection, and remote sensing. With the development and success of DNNs, deep learning has been applied to various sensor data domains in the past several years, such as bio-sensors and remote sensing. Although the past decade has brought many advances in object detection, it remains a challenging

problem in aerial images. Sensor data have some unique features, different from conventional object detection datasets. For example, aerial images are different from conventional image in the following ways: (1) Objects in aerial images often appear with arbitrary orientations. (2) The scale variations of objects in aerial images are much larger than those in conventional images, and many small objects are crowed together in aerial images. (3) The backgrounds of some aerial images are uniform and simple, while others have complex backgrounds. These characteristics make object detection in aerial images a challenging problem. To improve recognition accuracy, in recent years, rotated box-based and multi-scale-based DNNs [22] have been proposed to address the first two issues. However, these networks are mostly complicated with many parameters, which leads to slow inference speed. Existing deep neural networks for object detection in computer vision can be classified as one-stage or two stage detectors. Two-stage detectors consist of a detection network to generate region proposals, followed by a classification network to recognize the object in each proposed region. In the first stage, a neural network, such as RCNN, is used to generate the potential location of each target object; In the second stage, another neural network determines whether each candidate location contains a target object or not. In comparison, one-stage localization and recognition in one shot. Examples include YOLO, SSD, and RetinaNet. One-stage detectors are usually simpler and faster than two-stage detectors, while achieving similar accuracy. For example, one-stage detector RetinaNet outperformed one of the best two-stage detectors, Faster RCNN, with a 4.0 mAP improvement on the COCO dataset [17]. In terms of object detection on aerial images, inference speed is a critical evaluation metric so that our work focus on developing

algorithm on one-stage detector. For object detection in aerial images in real time, one-stage detectors, such as YOLO, SSD and RetinaNet, have the speed advantage.

1.2 Ambulatory assessment analysis

Currently, most methods in clinical psychology research primarily rely on questionnaires and interviews with examiners in the lab setting. With the rapid development of mobile technologies, a new promising solution is a mobile ambulatory assessment system with real-time data monitoring and collection of real-life subject behavioral and psychology data, as well as physiological data. Ambulatory assessment is the use of field methods to evaluate subjects in natural or unconstrained environments.

By combining information about the external environment, and participants' physiological and mental states, collected through system-generated and self-report surveys, machine learning models can be developed to identify changes in mood, alcohol use and/or craving, as well as other psychological problems. This same information can also be applied to context aware applications. In context aware computing, context is information that can be used to describe the state of something that is relevant to a user's interaction with an application. Combining methodology from psychophysiological field research with body area wireless sensor networks and mobile devices can improve context aware computing.

1.3 Contributions

This dissertation makes the following contributions:

1. A new Adaptive saliency Biased Loss (ASBL) method has been proposed for training deep neural networks, which is defined based on adaptive saliency

information of the input image. ASBL can be implemented at the image level, which is called image-based ASBL, and at the anchor level, which is called anchor-based ASBL. They use complexity information of input images to weigh the inputs differently in training. Without loss of generality, the ASBL approach was applied to RetinaNet to show its effectiveness. Using two large benchmark datasets, DOTA and NWPU VHR-10, experimental results show that ASBL-RetinaNet outperformed existing state-of-the-art deep learning methods, with at least 6.4 mAP improvement on DOTA, and 2.19 mAP on NWPU VHR-10. Furthermore, ASBL-RetinaNet improved over the original RetinaNet by 3.61 mAP on DOTA and 12.5 mAP on NWPU VHR-10.

2. Improved deep learning models have been developed for a new bird detection dataset of aerial images, Little Birds in Aerial Imagery (LBAI). The dataset was created from real-life aerial imagery. Some of the best deep learning architectures have been applied and improved on LBAI, which include object detection techniques such as YOLOv3, SSD, and RetinaNet, and small instance segmentation techniques such as U-Net and Mask R-CNN. Experimental results show that RetinaNet performed the best, outperforming other models by 1.4 and 4.9 F1 scores on the Easy and Hard subsets of LBAI, respectively.
3. A new data analysis pipeline for detecting alcohol usage based on wearable psychological sensor data, called ADA (Automatic Detection of Alcohol), has been developed. A new deep learning method, called Deep ADA, has been developed for extracting features from psychological signals to predict alcohol usage of real subjects in their daily lives. Deep ADA uses a large amount of

unlabeled data in unsupervised learning to enhance the learned features. It outperformed traditional feature extraction methods by improving detection accuracy by up to 19%.

1.4 Dissertation Organization

The rest of the thesis is organized as follows:

1. Chapter 2 presents the new adaptive salience biased loss for object detection in aerial images.
2. Chapter 3 presents deep learning object detectors for aerial images and experiments on the new bird detection dataset.
3. Chapter 4 presents the new CNN based feature extraction method, Deep ADA, for analyzing physiological sensor and survey data and detecting alcohol usage from physiological data.
4. Chapter 5 summarizes the dissertation.

2. NEW LOSS FUCNTIONS FOR IMPROVING OBJECT DETECTION IN AERIAL IMAGES

2.1 Abstract

Object detection in aerial images remains a challenging problem due to low image resolution, complex backgrounds, and variations of scale and orientation of objects in images. In recent years, several multi-scale and rotated box-based deep neural networks have been proposed and achieved promising results. In this chapter, a new loss function, called Adaptive saliency Biased Loss (ASBL), is proposed for training deep neural networks, which is defined based on adaptive saliency information of the input image. The proposed loss functions weights training examples and anchors differently based on input and saliency map complexity measurement in order to avoid over-contribution of easy cases in the training stage. In our experiments using two large public benchmark data sets of aerial images, DOTA, and NWPU VHR-10, RetinaNet was trained with ASBL to generate a one-stage detector, ASBL-RetinaNet. ASBL-RetinaNet outperformed the original RetinaNet by 3.61 mAP and 12.5 mAP, respectively, on the two data sets. In addition, ASBL-RetinaNet outperformed 10 other state-of-art object detection deep neural networks.

2.2 Introduction

In recent years, deep neural networks have achieved huge success in many areas of computer vision, such as image classification [1]–[3], object detection [4]–[11], and remote sensing [12]–[15]. Although the past decade has brought many advances in object

detection, it remains a challenging problem. For instance, CNNs have been applied to image classification problems in ImageNet [16] and surpassed the error rate of human vision ability; however, the best-performing object detection model on the COCO dataset [17] only achieved around 40 mAP (mean Average Precision) when the IoU (Intersection over Union) of the ground truth box and predicted box is 0.5. In addition to prediction accuracy, the inference time of a neural network model is another important performance metric.

Existing deep neural networks for object detection in computer vision can be classified as one-stage or two stage detectors. Two-stage detectors consist of a detection network to generate region proposals, followed by a classification network to recognize the object in each proposed region. In the first stage, a neural network, such as RCNN [4], is used to generate the potential location of each target object; In the second stage, another neural network determines whether each candidate location contains a target object or not. In comparison, one-stage localization and recognition in one shot. Examples include YOLO [8], SSD [11], and RetinaNet [18]. One-stage detectors are usually simpler and faster than two-stage detectors, while achieving similar accuracy. For example, one-stage detector RetinaNet [18] outperformed one of the best two-stage detectors, Faster RCNN [5], with a 4.0 mAP improvement on the COCO dataset [17]. In terms of object detection on aerial images, inference speed is an critical evaluation metric so that our work focus on developing algorithm on one-stage detector.

With the development and success of DNNs, deep learning has been applied to various sensor data domains in the past several years, such as bio-sensors [19], [20] and remote sensing [12]–[15], [21]. Sensor data have some unique features, different from

conventional object detection datasets. For example, aerial images, as shown in Fig. 1, are different from conventional image in the following ways: (1) Objects in aerial images often appear with arbitrary orientations. (2) The scale variations of objects in aerial images are much larger than those in conventional images, and many small objects are crowed together in aerial images. (3) The backgrounds of some aerial images are uniform and simple, while others have complex backgrounds. These characteristics make object detection in aerial images a challenging problem. To improve recognition accuracy, in recent years, rotated box-based [22], [23] and multi-scale-based DNNs [22] have been proposed to address the first two issues. However, these networks are mostly complicated with many parameters, which leads to slow inference speed. For object detection in aerial images in real time, one-stage detectors, such as YOLO, SSD and RetinaNet, have the speed advantage.

Figure 1. Sample Images of DOTA, showing the variation of scale and orientation of target objects (boat, truck, car, airplane) in aerial images. Harbor, Plane, Small Vehicle, Large Vehicle are the target objects.

In this chapter, we propose a new loss objective function, Adaptive saliency Biased Loss (ASBL), that can be used to train one-stage detectors to achieve better recognition accuracy, while keeping the one-stage detectors' speed advantage. We used the idea of saliency-based detection [24]–[26] in deep learning neural networks to map different level of features in aerial imagery in order to extract object information. The new loss function has two terms, image-based and anchor-based loss term. In the image-based term, input

images are weighted differently based on their saliency complexity. If the input images are with higher saliency information, it will be given with more weight on its classification loss function. In the anchor-based term, all anchors are given adaptive weights trained by neural network based on saliency complexity of interested objects during training phase. When loss converged during the training phase, with the same scale of training loss decrease, the images and anchors with high saliency information will contribute more. The goal of this loss function is to focus training on complicated images and saliency areas, which prevents the vast number of easy images and negative anchors from overwhelming the cross-entropy loss of the model. The loss function can be applied to any one-stage mutli-scale feature extraction detector network. In our experiment, the loss function was applied to train RetinaNet [18] and the trained network is called ASBL-RetinaNet. Two widely used public benchmark datasets were used for performance evaluation: DOTA [21] (one of the largest object detection dataset of aerial images) and NWPH VHR-10 [27]. Experimental results show that ASBL-RetinaNet outperformed other state-of-the-art object detectors. It outperformed RetinaNet with post-tuning [18] by 4.35 mAP (mean Average Precision) on DOTA and yields a 12.5 mAP improvement over a set of existing methods on NWPU VHR-10 data.

2.3 **Related Work**

Deep learning methods have been applied to object recognition in images, including aerial images, and achieved state-of-the-art results. For detecting objects in aerial images, there are major 4 kinds of methods have been used in research, template matching-based, knowledge-based, OBIA-based, and machine learning-based, as discussed in [28]. In terms

of machine learning based methods, HOG, Haar-like and SR-based information are extracted, then features fusion and dimension reduction are used to filter necessary information. Based on useful information, the feature extracted are fed into classifiers, like SVM, Adaboost. In recent year, most existing works use object detectors using deep learning that have achieved good performance on natural images. However, due to the unique properties of aerial images, these object detectors did not perform well compared with natural images detection. Basically, researchers [13]–[15] have proposed various methods based on fine-tuning pretrained networks, such as pretraining on ImageNet [16] and COCO data [17]. Since most objects in aerial images are quite small, the fine-tuning using aerial images helped improving accuracy. In addition, computer vision scientist also design and propose unified deep learning network for characters of aerial images, like multi-scale and multi-angle, to achieve better performance on aerial image detection. For example, existing work [29], [30] propose rotation-invariant deep learning models with variant of regularization to achieve the SOTA performance on remote sensing images. Moreover, instead of all supervised learning, weakly supervised learning methods [31] in deep learning has been proposed to learn high-level features in unsupervised manner to capture the structural information of object in remote sensor images. These methods reduce the human labeling work of training data.

In terms of deep learning network detectors, models like SSD [11], YOLO [8] and RetinaNet [18] have been proposed and achieved good performance in object detection in images. Previously, one-stage detectors have faster inference speed than two-stage detectors, but their accuracy is not as good. However, one-stage detector RetinaNet [18] was able to outperform state-of-the-art two-stage models on both speed and accuracy: 4.0

mAP higher than Faster RCNN [5], on the COCO data [17]. RetinaNet [18] combines the advantages of the SSD [11] and YOLO [8] networks by performing a multilayer feature extraction and then feeding them into a subnetwork to generate final outputs. RetinaNet [18] uses Focal Loss to address the one-stage detector problem in which there is an extreme imbalance between foreground and background classes during training.

In training object detectors, imbalances of easy and hard cases and positive and negative examples will lead to poor performance. In general, more hard positive examples enable the model to discover and expand sparsely sampled minority class boundaries, while more hard negative examples improve the margins of minority class boundaries corrupted by visually similar classes. Random sampling techniques have been used to address the class imbalance problem [32]. Mining hard examples has been shown to be effective [33]. Recently, Online Hard Example Mining (OHEM) [10] has been proposed, which is an online bootstrapping algorithm for training region-based ConvNet object detectors like Fast RCNN [34]. For one-stage detector, specifically SSD [11], the ratio of positive and negative examples with random sampling is more balanced, which led to faster convergence and more stable training. However, most one-stage detectors still have the problem of unbalanced positive and negative anchors. For example, DSSD [35] and RetinaNet, could have up to 40k and 100k anchors, on benchmark images, with a very small fraction of positives. The proposed new loss function aims to address both the training example imbalance and anchor imbalance problem.

The Focal Loss function [18] was designed to improve the cross entropy loss function on class-imbalance and easy/hard example imbalance problem in neural network

training. The cross entropy between a predication by a network model and the target label is defined as follows.

$$CE(p,y) = \begin{cases} -log(p), & if\ y = 1 \\ -log(1-p) & if\ y = 0 \end{cases} \quad (1)$$

where p is the class probability by the model, and y is ground-truth class label. y=1 and y=0 means positive and negative samples, respectively. For convenience, let CE(p, y) =-log(p_t).

$$p_t = \begin{cases} p, & if\ y = 1 \\ 1-p & if\ y = 0 \end{cases} \quad (2)$$

In practice, after an object detector DNN is trained based on cross-entropy, easy examples still incur a small amount of positive loss. When the number of easy samples is very large compared to hard training examples, the sum of the small losses of the easy examples dominates the loss of hard examples. Focal Loss function was proposed to address the class-imbalance and easy/hard example problem in one-stage detectors. It prevents the easily classified negatives from overwhelming the loss function and dominating the gradient. The idea is to introduce a weight factor α for foreground and 1- α for background and add a new factor (1 - p_t) to cross-entropy loss with tuning parameter. Now, if an example is mis-classified, the new factor will be near 1 and the loss function will be about the same; however, if an example is predicted correctly, the factor will scale the loss near to 0 so that the importance of the easy class in the loss function will be very small. With the tuning parameter, the scale of importance of the factor can be tuned empirically. The focal loss function is:

$$FL(p,y) = \alpha * (1-p_t)^{\gamma} * CE(p,y) \quad (3)$$

where α and γ are constants. We used α= 2 and γ = 0.25 in our experiments, as suggested in previous work for natural image object detection.

2.4 Adaptive Saliency Biased Loss

When a detector network is trained using a set of training examples, the training images are commonly treated equally. If the majority are easy cases, the trained model may focus on the easy cases and the hard cases do not exert sufficient influence to make the model more generalize. In addition, because the prediction of most of one-stage detectors are based on anchors of reception fields, class imbalance and improper hyper-parameter selection could lead to poor performance. To address these issues, we propose a novel loss function, called Adaptive Saliency Biased Loss, to train and improve object detectors. The loss function has two terms, one giving complicated images more weights during training and the other dealing with the anchor problem in one-stage detector. Our idea is to use the saliency map of images to represent the complexity and important areas of input and dynamically weight each input sample and anchors in feature map during training.

2.4.1 Image-Based Adaptive Saliency Biased Loss Function

We propose an image-based adaptive saliency biased loss function to direct training more toward difficult cases, i.e., images containing objects that are hard to detect and recognize. Some existing methods use Edgebox [36] to determine the complexity of training images, like WiderFace [37] and other self-design and labeled images [38], [39]. However, all these approaches are complicated and time consuming and the proper parameter values in Edgebox are hard to decide.

In our method, a pretrained deep neural network is used to determine the complexity of input images based on the assumption that an input image is more complex if there are more activated neurons in a hidden level. Many state-of-the-art DNNs have

been trained on large-scale image datasets, such as ImageNet [16], and have the ability to detect features and shapes of general objects at different levels. In computer vision, a saliency map is an image that shows each pixel's unique quality. Based on all these insights, pretrained DNNs by ImageNet can be used as saliency estimators to estimate the complexity of an input image.

Specifically, we use a CNN (convolutional neural network) pretrained on ImageNet as a saliency estimator and extract features from different convolutional layers to represent the complexity of an input image, S_I, as defined in the following formula:

$$S_I(x) = \frac{1}{C * W * H} \sum_{c=1}^{C} \sum_{w=1}^{W} \sum_{h=1}^{H} f_{c,w,h}(x) \qquad (4)$$

where S_I is the saliency of an image defined as the average activation of an convolution layer; x is the input image; $f_{c,w,h}$ is the output of a convolutional layer in a pretrained CNN with output dimension C*W*H. According to this formula, easy input images would have fewer activated neurons in a convolutional level and will result in smaller values of S_I than complicated input images. S_I values computed based on different convolutional layers in a DNN represent complexity at different feature levels. In the experiments, we investigated S_I from different individual convolutional layers, as well as composite S_I from multiple layers, which captures a multi-scale view for each input image.

In order to fix range of weighting factor, we propose a normalization formula as follows:

$$S'_I = \frac{S_I - S_{min}}{S_{max} - S_{min}} (S_{new\_\max} - S_{new\_\min}) + S_{new\_\min} \qquad (5)$$

where S_I is the original saliency value; S_{min} and S_{max} are the overall minimum and maximum S_I value of the training set, calculated once before the training phase; S_{new_max},

S_{new_min} are constants. S_{new_max} is set as 1, and S_{new_min} is set based on empirical results. In our implementation, we tried different values of S_{new_min}, such as 0.3, 0.5, 0.7, etc.

The new Image-Based Adaptive saliency Biased Loss function, $ASBL_I$, incorporates the saliency information as follows:

$$ASBL_I(p, y) = S'_I * FL(p, y) \quad (6)$$

where p is the class probability generated by the model, y the ground-truth class label, and FL (p, y) the Focal Loss. The saliency value of each image becomes the weight on the focal loss of the image. Therefore, the loss values from easy cases will be smaller due to smaller S_I' values. Fig. 2 shows an example of $ASBL_I$ based on RetinaNet and ResNet50.

The Image-based Adaptive saliency Biased Loss has two major properties: (1) As the loss converges, the hard cases will contribute more and the easy cases will contribute less, because the easy cases will have small loss values. (2) When S_I' are computed based on different convolution layers, multi-scale features are incorporated in the loss function. For instance, the lower level features have larger feature map, and each point in its feature map represents a small object in the original images.

2.4.2 **Anchor-Based Adaptive Saliency Biased Loss Function**

Redundant anchors cause unbalanced classification problems in single-shot object detectors, such as RetinaNet and DSSD. Each anchor in multi-scale feature map will make prediction for category and localization of objects in object detectors. To fully cover an input image, single-shot detectors usually generate many anchors of difference sizes. However, in aerial images, most objects of interests are small and some of the images are with clear background, which leads to more redundant anchors than those for larger objects.

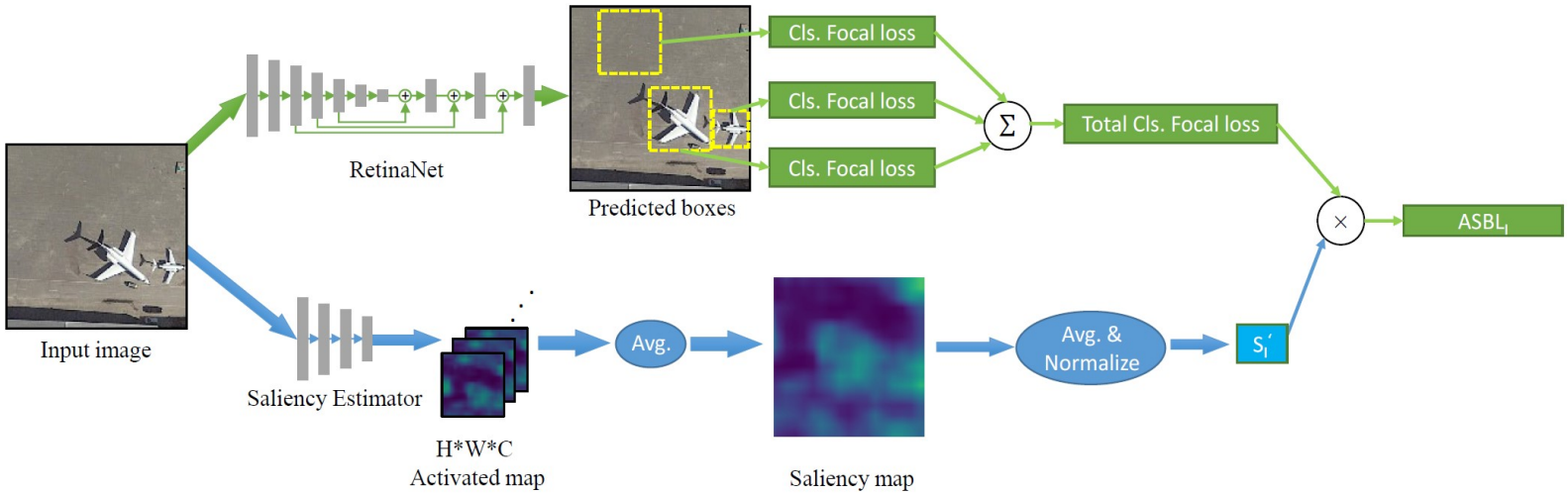

Figure 2. An illustration of the Image-Based Adaptive Saliency Biased Loss (ASBLI) function. The top branch is RetinaNet. The bottom branch is the saliency estimator network. In the bottom branch, saliency estimator is the activation of conv2 of ResNet50. $ASBL_I$ is generated by multiplying the Focal Loss of the top network with the average activation of the saliency estimator.

To address this issue, we propose anchor-based adaptive saliency biased loss, $ASBL_A$. We assume there are two classes for each point in activated feature map, saliency and non-saliency. If the point in feature map with high probability of saliency information of objects, it will be given higher weight. In general, each point in feature map has a set of anchors with different aspect ratios and scales in single-stage object detectors. In our theory, we applied Bayesian theory to each point in activated feature map to calculate the probability of saliency information on the point and then the saliency information will be fed into the series of anchors of the point. The idea is to use attention mechanism of saliency map to present the saliency complexity of each anchor and then weight the predicted anchors accordingly as follows:

$$Pr(S|A,I) \propto Pr(A|S,I) * Pr(S|I) \tag{7}$$

where Pr (S|I) is the prior probability of saliency information for each point in feature map given an input image I and Pr (A|S, I) is the likelihood probability of positive anchors of each point on feature map given image saliency information S and input image I. Based on Bayesian theory, Pr (S|A, I) is the posterior probability of saliency information of anchors on each point on feature map given input images I and anchors A. In addition, S, A and I are independent events so that there is no correlation between all these events. In our implementation, feature maps trained by the same single-shot object detector with $ASBL_I$ will be used to represent Pr (A|S, I) and Pr (S|I) is derived from ResNet50. The representation of saliency map for a set of anchors is as follows:

$$SA_{u,v}(x) \propto \frac{1}{C}\sum_{c=1}^{C} R_{c,u,v}(x) \odot \frac{1}{C}\sum_{c=1}^{C} f_{c,u,v}(x) \tag{8}$$

where u, v is coordinate and c is channel of feature map; $R_{c,u,v}$ is the feature map of a single-shot object detector; $f_{c,u,v}$ is a pretrained convolution layer by ImageNet with dimension C * W * H, the same as $R_{c,u,v}$ is the input image; and $SA_{u,v}$ is the saliency level for each set of anchors. For $R_{c,u,v}$ and $f_{c,u,v}$, we average all the channels for each one to get likelihood probability of positive anchors of each point on feature map and prior probability of saliency information for same point on feature map which is Pr (A|S, I) and Pr (S|I) in (7), respectively. In this formula, $f_{c,u,v}$ is used to estimate prior knowledge of the complexity of an image and $R_{c,u,v}$ is the likelihood of positive anchors of object in an image. During training phase, $R_{c,u,v}$ is dynamically updated and learned so that $SA_{u,v}$, saliency information, is also dynamical adaptive in each training based on input images. Thus, the final anchor-based ASBL is as follows:

$$ASBL_A(p, y) = \sum_{a=1}^{As} \sum_{v=1}^{H} \sum_{u=1}^{W} SA_{u,v}(x) * FL_{u,v,a}(p, y) \quad (9)$$

where $FL_{u,v,a}$ (p, y) is the loss objective function for each anchor; As is the number of anchors for a feature map; W and H is the dimension of each feature map. $ASBL_A$ can be learned and adapted in the training because of dynamic of $SA_{u,v}$. Fig 3 shows an illustration of training process of $ASBL_A$. The top branch shows the inference process of a single shot detector, which is RetinaNet, the middle branch shows the generation of the likelihood of positive anchors of object in an image, $R_{c,u,v}$, and the bottom branch shows the process of prior knowledge of the complexity of an image, $f_{c,u,v}$. According to formula (8) and (9), $ASBL_A$ has these properties: (1) $R_{c,u,v}$ will be dynamically updated during the training process so that $SA_{u,v}$ will be learned. (2) Each set of anchors of each point on feature map have the same weighting value. Anchors predicted wrong will carry more weights. (3) If $SA_{u,v}$ is small, it means the content in the anchors is simple, which leads to small

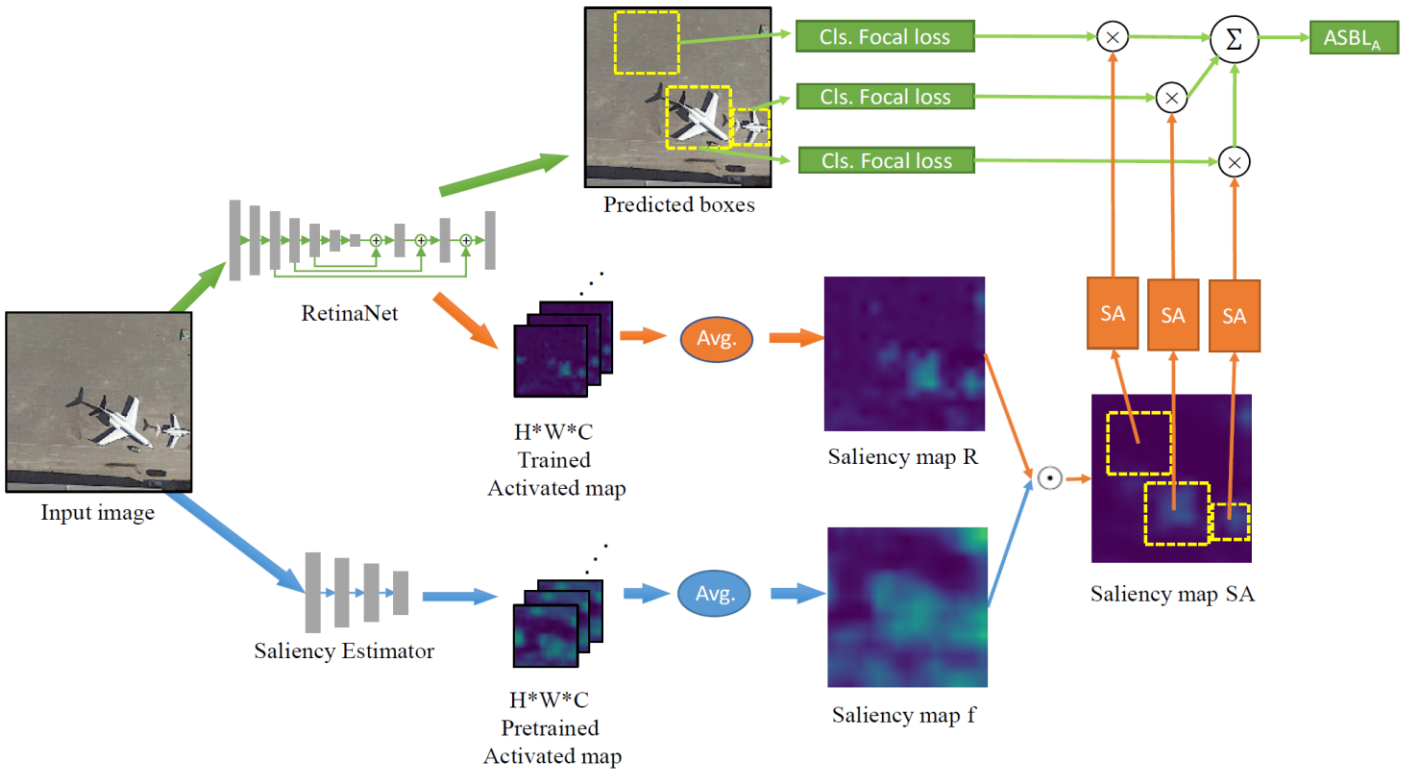

Figure 3: An illustration of the Anchor Based Adaptive Saliency Biased Loss Function, ASBLA. The top branch is to generate inference result of RetinaNet. The middle branch shows the process of saliency map $R_{c,u,v}$. The bottom one demonstrates how to generate saliency map $f_{c,u,v}$. After generating saliency map $SA_{u,v}$, each value in $SA_{u,v}$ will be used to weight classification loss of anchors. ASBLA uses the focal loss of each anchor to weight the final loss based on its saliency information.

contribution in the loss function. In addition, during training $ASBL_A$, $SA_{u,v}$ will be adaptively learned so that more redundant anchors without useful information will be ignored. Thus, training phase are more straight-forward and work more on the anchors with higher saliency information.

2.4.3 **ASBL-RetinaNet**

Our final loss function, ASBL, combines the two loss functions, $ASBL_I$ and $ASBL_A$, in the training process. Specifically, $ASBL_I$ is used in the first half of the training process and $ASBL_A$ is used in the second half, as shown in the following formula:

$$ASBL(p,y) = \begin{cases} ASBL_I(p,y) & if\ e \leq 0.5 * ep \\ ASBL_A(p,y) & otherwise \end{cases} \quad (10)$$

where e is the epoch index and ep are the total number of epochs for training.

ASBL can be instantiated based on any one-stage deep neural network detector. For example, if ASBL is computed based on RetinaNet [18], which is one of the best one-stage detectors, as shown in Fig. 2 and Fig. 3, we call the instantiation ASBL-Retina. In this case, the detector is RetinaNet, while the training is based on the ASBL loss function. The performance of the trained network can be compared directly with that of the network trained in the original way. The inference times of the networks trained in the two different ways will be similar.

In our experiment, ResNet50 [3] is used to extract prior saliency information of input. In order to extract the same level of features as RetinaNet, we pretrained ResNet50 using ImageNet with two more convolution blocks to get intermediate results in the same dimension and shape as the encoder part of RetinaNet. The features extracted from the revised ResNet50 are denoted as {C2-C7}. The corresponding feature maps in the encoder

part of RetinaNet are denoted as {P2-P7}. These features are used to generate saliency information of input images. Each extracted feature will be used as a weight factor of training images in the loss function.

Algorithm 1 shows the method to train RetinaNet using ASBL. The inputs are the original RetinaNet and ResNet50. ResNet50 provides stationary image level saliency information. The updated parameters, W, in RetinaNet is the output. First, we pretrain ResNet50 with two more convolution block with same architecture of encoder of RetinaNet using ImageNet in order to generate image level saliency information. Then, in our implementation, we use 50 epochs in training. The first 25 epochs are used to train RetinaNet based on $ASBL_I$, and the remaining ones are to train the network based on $ASBL_A$. In the first 25 epochs, $ASBL_I$ is calculated by retrained ResNet50 using formula (4) and (6). In terms of the remaining ones, $ASBL_A$ is generated by the features map of retrained ResNet50 and RetinaNet in the first half of epochs according to formula (8) and (9). The feature map of RetinaNet in the second half of epochs are updated during the training process so that the weighting factors are dynamically adjusted.

Algorithm 1 Training ASBL-RetinaNet

Require: RetinaNet, ResNet50
Ensure: Parameter W in RetinaNet
$N = 0$
Pre-trained ResNet50 with 2 more conv-block using ImageNet
while $N < 50$ **do**
  if $N < 25$ **then**
    1. Calculate S_I using C2 of Retrained-ResNet50
    2. Update parameter W in RetinaNet with $ASBL_I$
  else $\{N >= 25\}$
    1. Output C3-C7 feature maps of Retrained-ResNet50
    2. Output P3-P7 feature maps of RetinaNet
    3. Calculate $SA_{c,w}$
    4. Update parameter W in RetinaNet with $ASBL_A$
  end if
  $N = N + 1$
end while

2.5 Experimental Results

In this section, experimental results on two benchmark datasets of aerial images, DOTA and NWPU VHR-10, are presented.

2.5.1 Dataset

DOTA is the largest and most diverse public dataset for multi-class object detection in aerial images. It consists of 2806 images collected from various camera sensors. The images are acquired from Google Earth and China Center for Resources Satellite Data and Application. The 15 object categories are: plane, baseball diamond (BD), bridge, ground field track (GTF), small vehicle (SV), large vehicle (LV), tennis court (TC), basketball court (BC), storage tank (SC), soccer ball field (SBF), roundabout (RA), swimming pool

(SP), helicopter (HC), and harbor. Across these categories, 57% are small objects that are within 50 *50 pixels. The DOTA dataset is split into training (1/2), validation (1/6), and test (1/3) sets.

NWPU VHR-10 is another widely used public dataset that consists of a positive image set including 650 images and a negative image set including 150 images over ten object categories. In our experiments, we used the official 1172 images (400 * 400 pixels) cropped from the positive image set of NWPU VHR-10 [27]. The data set contains ten classes of geo-spatial objects: airplane, ship, storage tank, baseball diamond, tennis court, basketball court, ground track field, harbor, bridge, and vehicle. To have a fair comparison with previous results, we used the existing train, validation and test dataset that contains 679 images for training, 200 images for validation and the rest 293 images for test. For performance evaluation, we followed the official way [27] to evaluate the performance of our methods.

2.5.2 **Evaluation Metric**

The performance metric used in our experiments is mean Average Precision (mAP), as for PASCAL VOC [30]. In our experiments, we focused on the HBB task in DOTA and set non-maximum suppression (NMS) to 0.3 for all categories. The IoU ratio of predicted and ground truth boxes is 0.5, as commonly used in the object detection domain.

For NWPU VHR-10 data, the parameter setting of performance evaluation is the same as the original paper [27]. We used public tools (https://github.com/Cartucho/mAP) to calculate mAP score. In order to show the robustness of the proposed ASBL-RetinaNet

method, ablation study is only done on the DOTA dataset. All hyper-parameters are fixed for experiments on the NWPU VHR-10 dataset.

2.5.3 **RetinaNet modification**

In addition to reporting the performance of the original RetinaNet, we also made some minor changes to RetinaNet and achieved improved performance. Specifically, we changed aspect ratios to {1:3, 1:1, 3:1}, and anchor sizes to $\{2, 2^{0.5}, 0.3\}$. The reason for these changes is that there are some object categories, such as bridge or harbor, that have long rectangle shapes. Anchor size 0.3 was used because some objects in aerial images are very small. In terms of data augmentation, instead of random flip used by original RetinaNet, random flip and flop were used.

2.5.4 **Ablation study**

2.5.4.1 Image based ASBL analysis

a) Image Complexity Analysis: In our method, we use the amount of activation in certain layers of a deep neural network to represent the complexity of an input image in $ASBL_I$. When the background of an image causes more neurons to be activated, the image is more complex. Fig. 6 shows examples of DOTA images selected based on their S_I values. The images in each group are selected based on their S_I values from C2 to C5 layers of ResNet50, respectively. The first 5 images in each group have the smallest S_I values and are visually simple, whereas the last 5 images have the largest S_I values and are visually complicated. Images with large S_I values obtained from earlier layers of ResNet50 (C2 and C3) have dense low-level image features (small objects), whereas those from latter layers of ResNet50 (C4 and C5) have more higher-level image features (large objects). The reason

is because the receptive field of C2 and C3 is smaller than the one in C4 and C5. The feature map generated from C2 and C3 contains more small objects' information, but C4 and C5 focus on larger objects' information. The examples show that S_I could capture the complexity of an input image quite well.

b) Saliency Normalization: Fig. 4 shows the distribution of saliency values S_I of DOTA training images obtained from residual block C2 and C5 of ResNet50, as an example. The S_I values from some layers, such as C5, have small ranges, which does not separate easy and hard cases sufficiently. Saliency normalization would be good solution in our implementation to solve out this problem and also fix the range of weighting factor of loss function.

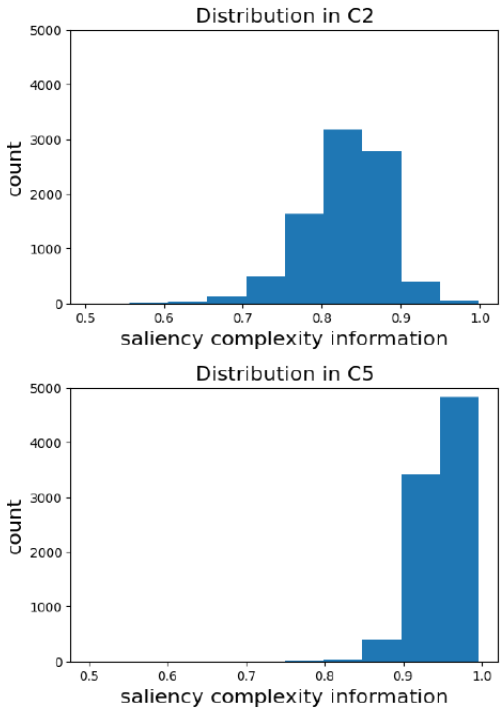

Figure 4. Distribution of saliency values of DOTA training images obtained from residual block C2 and C5, respectively, of ResNet50.

To show the effectiveness of the new loss function with new complexity information S_I, we compared the performances of the modified RetinaNet trained using the new loss function with those trained using the original loss function. Table. 1 shows some experimental mAP results using the new loss function with or without saliency normalization. The saliency values were calculated from the C5 layer of ResNet50. Among these results, the best performance was achieved when saliency normalization was used with S_{new_min} 0.5, which is 0.62 mAP higher (63.48 vs. 62.86) than the result without normalization. In comparison, the mAP of the modified RetinaNet trained using the original loss function is 62.51, which is almost 1 mAP lower than the result using the new loss function (63.48). We ran these experiments multiple times and the mAP standard deviations for these two methods are 0.017 and 0.022, respectively, which means their performance improvement is significant. Based on the ablation study of saliency normalization, we notice that if weighting factor of easy cases are too small, it will make underfitting for those easy cases to lower the performance, however, if it would be larger on easy cases, it would have similar results compared with no weighting factors on loss function.

c) Comparison of Saliency at Different Feature Levels: Saliency values calculated based on features at different levels could lead to different model performances. In these experiments, we calculated saliency values using C2 to C5 layers from ResNet50 with normalization. Using each layer's output, $ASBL_I$ is calculated and fed to the new loss function to train RetinaNet. Table. 2 shows that the best performance (64.77 mAP) was achieved when using saliency values calculated from C2 layer, which is 2.26 mAP higher than the original RetinaNet (62.51 mAP).

Table 1.Performances of the modified RetinaNet on the DOTA dataset trained using the new loss function with or without saliency normalization.

$S_{new\_\min}$	**Normalization**	**mAP**
0.3	Y	61.79
0.5	Y	**63.48**
0.7	Y	62.4
-	N	62.86

Table 2. Performance Comparison of RetinaNet trained using the new loss function with saliency values calculated at differ layers (C2 to C5) of ResNet50.

Conv Block	**C2**	**C3**	**C4**	**C5**
mAP	**64.77**	64.51	63.32	63.48

2.5.4.2 Anchor based ASBL analysis

Similar to image-based ASBL, saliency values computed to represent the complexity of anchors can also be normalized. For anchor saliency, we use the same normalization formula as for image-based saliency. For each anchor, we use the maximum and minimum value of each feature map as max and min in the formula.

Table 3 shows experimental results using saliency normalization with different S_{new_min} (0.3, 0.5, and 0.7), or without normalization. The best result was achieved with $S_{new\_min} = 0.5$.

For anchor based ASBL, saliency values could be calculated during the training phase. In our experiments, comparison with fixed and dynamic anchor based ASBL is provided to show the efficiency of dynamic anchor based ASBL. The fixed loss used initial feature map generated by RetinaNet with $ASBL_I$ as $R_{c,u,v}$ to calculate $ASBL_A$. Table 3 shows that using dynamically updated saliency values can improve the performance from 64.82 to 66.12, with the normalization of anchor saliency values. Table 3 also compares

the performance difference between using image-based and anchor-based ASBL. The best performance using anchor-based ASBL is 66.12, which is higher than using image-based ASBL (64.77), on DOTA test dataset.

Table 3. Performance comparison of anchor-based and image-based ASBL methods.

Model	Dynamic	Normalization	$New\_\min$	mAP
$ASBL_A$	Y	Y	0.3	65.54
			0.5	**66.12**
			0.7	65.58
		N	-	65.53
	N	Y	0.5	64.82
$ASBL_I$	-	Y	0.5	64.77

2.5.5 **Experiment setup**

In our models, we used ResNet50 [28] as a backbone of RetinaNet [43]. The input image size was 1024*1024 for DOTA images and 400 * 400 for NWPU VHR-10 images. For NWPU VHR-10, we resized the images to 600*600, the same as the original paper of RetinaNet. We used Adam as the solver in training. One Titan X GPU desktop was used in the experiments with training batch size 2. Pretrained ResNet50 weight by ImageNet are applied as initial parameters of backbone model. Random flip and flop are used as data augmentation. Unless otherwise specified, all models were trained for 50 epochs with initial learning rate 0.0001, which was then divided by 10 after every 20 epochs. During training, for the ASBL method, we first used the image-based ASBL loss function to train the network for 25 epochs, and then used the anchor-based loss function to train the network for 25 more epochs. The training dataset of DOTA and NWPU VHR-10, the same as previous published work, were used in training.

2.5.6 **Experimental results on DOTA**

On the largest aerial image dataset, DOTA, we compare the performance of RetinaNet trained using the new ASBL loss function with some recent state-of-art deep learning methods, including both one-stage and two-stage detectors. Table 5 shows the performances of various existing deep learning methods, including YOLO, SSD, RFCN, Faster RCNN, RetinaNet, as well as ASBL-RetinaNet, on the test dataset. In the "Data" column, "T" means that a model was trained using the training dataset of DOTA, whereas "T+V" means that a model was trained using the combined training and validation dataset. Their performances on each target categories, as well as the overall average in mAP (the last column) are shown.

The results show that the new method ASBL-RetinaNet trained using T+V achieved the highest average precision, 66.86, which was 6.4 mAP higher than the closest competitor, Faster RCNN [5] (60.46). Across all 15 target categories, our ASBL-RetinaNet outperformed Faster RCNN in 10 categories and modified RetinaNet in all 15 categories using DOTA train data only. Note that the modified RetinaNet is much better than the original RetinaNet. ASBL-RetinaNet outperformed the modified RetinaNet by 3.61 mAP (66.12 vs. 62.51) when trained using DOTA training dataset only. The inference speeds of the various RetinaNet models are the same, since their architectures in inference are the same. Compared to all other models, ASBL-RetinaNet is the best for 8 out of 15 target categories.

Fig. 7 shows the detection results of modified RetinaNet (top) and our proposed ASBL-RetinaNet (bottom) on four examples of DOTA test images. Comparing the two results in the first column, RetinaNet misclassify harbor object as background due to the

crowded of boat but ASBL-RetinaNet detect it and keep other detected objects. Comparing 2nd and 3rd columns in Fig. 7, RetinaNet miss different scales of objects, like swim pools crossed the river and airplane in the top left corner, however, ASBL-RetinaNet improve the accuracy for object in different scales, even though there is nothing special design to solve out multi-scale problem. Consider the high complexity of 2nd column images, the weighting will be given higher in the training phase. Moreover, due to the high complexity of anchors in the top-left corner, the more training weight also given in the training phase. That is the reason why ASBL-RetinaNet can improve accuracy in different scales of object. In addition, the 4th columns in Fig. 7 shows that ASBL-RetinaNet also can improve the performance of sample with easy background, even if it focus more on training ones with complex background.

2.5.7 **Performance on NWPU-VHR 10**

Table 6 shows the performances of various existing deep learning methods, including COPD [41], Transferred CNN [1], RICNN [29], Faster RCNN [5], Li's method [27], modified RetinaNet, as well as ASBL-RetinaNet, on the test dataset of NWPU VHR-10. The proposed method is the best, achieving 89.31 mAP, which is 2.19 mAP higher than the best previous result (87.12) and 12.5 mAP higher than the modified RetinaNet. In order to provide fairly comparison, ASBL-RetinaNet with VGG 16 backbone has been implemented and the performance is better than Li's method and Faster RCNN with the same backbone by 1.42 mAP. Compared to RetinaNet trained using original loss function, RetinaNet trained using the new ASBL loss function is better for all target categories significantly. To analysis the variance of performance improvement with DOTA and

NWPU VHR-10 datasets, there are mainly two reasons. 1) Average of image complexity of test data in NWPU VHR-10 is higher than the one in DOTA, which is with 0.88 and 0.84 on C2 saliency information, respectively. Thus, ASBL works better on NWPU VHR-10 datasets. 2) NWPU VHR-10 only has 293 test dataset and DOTA has more than 1000 test images so that the variation of model performance will be larger in NWPU VHR-10. Table 4 shows the inference speed comparison of various methods. The inference speed of ASBL-RetinaNet is 2 times faster than Faster RCNN, which is the fastest previous model and took 45ms for each image using an NVIDIA Titan X GPU and 16 GB of memory as reported in their paper [27]. In our work, we use the similar devices which is NVIDIA Titan X GPU and 12 GB of memory. Fig. 5 shows the results of ASBL-RetinaNet on 6 examples from NWPU VHR-10 test dataset. The method detected objects of various sizes and shapes in these images successfully.

Table 4. Inference time comparison of various detection models on NWPU VHR-10 images.

Models	**Average running time per images (s)**
COPD [41]	1.16
Transferred CNN [1]	5.09
RICNN [29]	8.47
Faster RCNN [5]	0.09
Li etc [27]	2.89
ASBL-RetinaNet	**0.045**

Figure 5. Detection results of ASBL-RetinaNet on 6 examples from NWPU VHR-10 test dataset.

2.6 **Conclusion**

In this work, we proposed a new loss function, Adaptive Saliency Biased Loss (ASBL). ASBL can be implemented at the image level, which is called image-based ASBL, and at the anchor level, which is called anchor-based ASBL. They use complexity information of input images to weigh the inputs differently in training. Without loss of generality, the ASBL approach was applied to RetinaNet to show its effectiveness. Using two large benchmark datasets, DOTA and NWPU VHR-10, experimental results show that ASBL-RetinaNet outperformed existing state-of-the-art deep learning methods, with at least 6.4 mAP improvement on DOTA, and 2.19 mAP on NWPU VHR-10. Furthermore, ASBL-RetinaNet improved over the original RetinaNet by 3.61 mAP on DOTA and 12.5

mAP on NWPU VHR-10. However, this work only considers saliency information of input images which may not be enough to represent the complexity of aerial imagery. To improve current work, rotation and scale information of objects could be also included into objective loss function. Github link: https://github.com/ps793/ASBL-RetinaNet

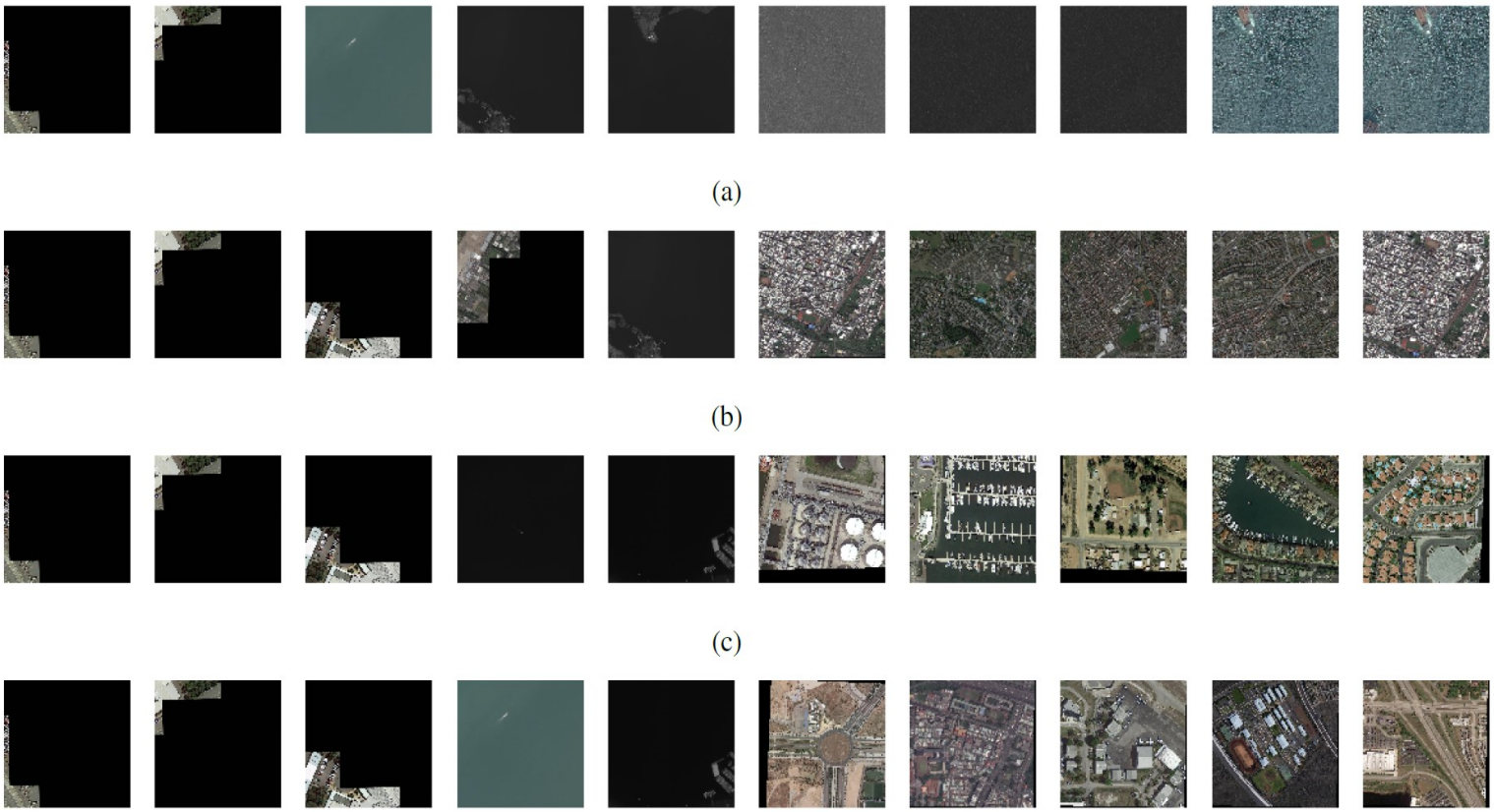

Figure 6. Multi-scale saliency analysis. The 5 images (first 5) of each row with the smallest SI values from the C2 to C5 (a to d) layer of ResNet50 , in comparison with the 5 images of each row (last 5) with the largest SI values from the same layer, respectively. The first 5 images in each group have the smallest SI values and are visually simple, whereas the last 5 images have the largest SI values and are visually complicated. Images with large SI values obtained from earlier layers of ResNet50 (C2 and C3) have dense low level image features (small objects), whereas those from latter layers of ResNet50 (C4 and C5) have more higher level image features (large objects).

Table 6. Results on NWPU VHR-10 test dataset

	Airplane	Ship	Storage tank	Baseball diamond	Tennis court	Basketball court	Ground track field	Harbor	Bridge	Vehicle	mAP
COPD [41]	62.25	69.37	64.52	82.13	34.13	35.25	84.21	56.31	16.43	44.28	54.89
Transferred CNN [1]	66.03	57.13	85.01	80.93	35.11	45.52	79.37	62.57	43.17	41.27	59.61
RICNN [29]	88.71	78.34	86.33	89.09	42.33	56.85	87.72	67.47	62.31	72.01	73.11
Faster RCNN [5]	90.9	86.3	90.53	**98.24**	89.72	69.64	100	80.11	61.49	78.14	84.51
Li etc [27]	99.70	90.8	90.61	92.91	**90.29**	80.13	90.81	80.29	68.53	**87.14**	87.12
RetinaNet* [18]	96.58	83.61	74.76	84.32	63.99	59.66	98.33	62.83	65.72	78.31	76.81
ASBL-RetinaNet (VGG16)	**100**	91.27	**96.76**	96.69	68.54	**87.67**	100	77.10	**83.92**	83.45	88.54
ASBL-RetinaNet (ResNet50)	99.34	**93.19**	94.36	97.70	71.19	84.67	**100**	**87.61**	81.49	83.50	**89.31**

Table 5. Results on DOTA test dataset

	Data	Plane	BD	Bridge	GTF	SV	LV	Ship	TC	BC	ST	SBF	RA	Harbor	SP	HC	mAP
YOLO [8]	T+V	76.9	33.87	22.73	34.88	38.73	32.02	52.37	61.65	48.54	33.91	29.27	36.83	36.44	38.26	11.61	39.2
SSD [11]	T+V	44.74	11.21	6.22	6.91	2	10.24	11.34	15.59	12.56	17.94	14.73	4.55	4.55	0.53	1.01	10.94
RFCN [9]	T+V	79.33	44.26	36.58	53.53	39.38	34.15	47.29	45.66	47.74	65.84	37.92	44.23	47.23	50.64	34.9	47.24
Faster RCNN [5]	T+V	80.32	**77.55**	32.86	**68.13**	53.66	52.49	50.04	90.41	**75.05**	59.59	**57**	49.81	61.69	56.46	**41.85**	60.46
RetinaNet [18]	T	78.22	53.41	26.38	42.27	63.64	52.63	73.19	87.17	44.64	57.99	18.03	51	43.39	56.56	7.44	50.39
RetinaNet*	T	89.03	62.14	43.88	47.05	73.57	65.18	78.65	90.86	66.28	70.26	35.07	58.26	68.93	66.34	22.16	62.51
ASBL-RetinaNet	T	89.09	67.96	46.38	57.12	73.55	66.19	**78.67**	90.86	71	**73.88**	45.15	60.92	70.01	68.51	32.49	66.12
ASBL-RetinaNet	T+V	**89.51**	74.07	**46.91**	55.54	**73.78**	**66.87**	78.48	**90.86**	70.09	73.2	46.71	**61.34**	**70.5**	**72.17**	32.84	**66.86**

* RetinaNet with modified anchor sizes and ratios

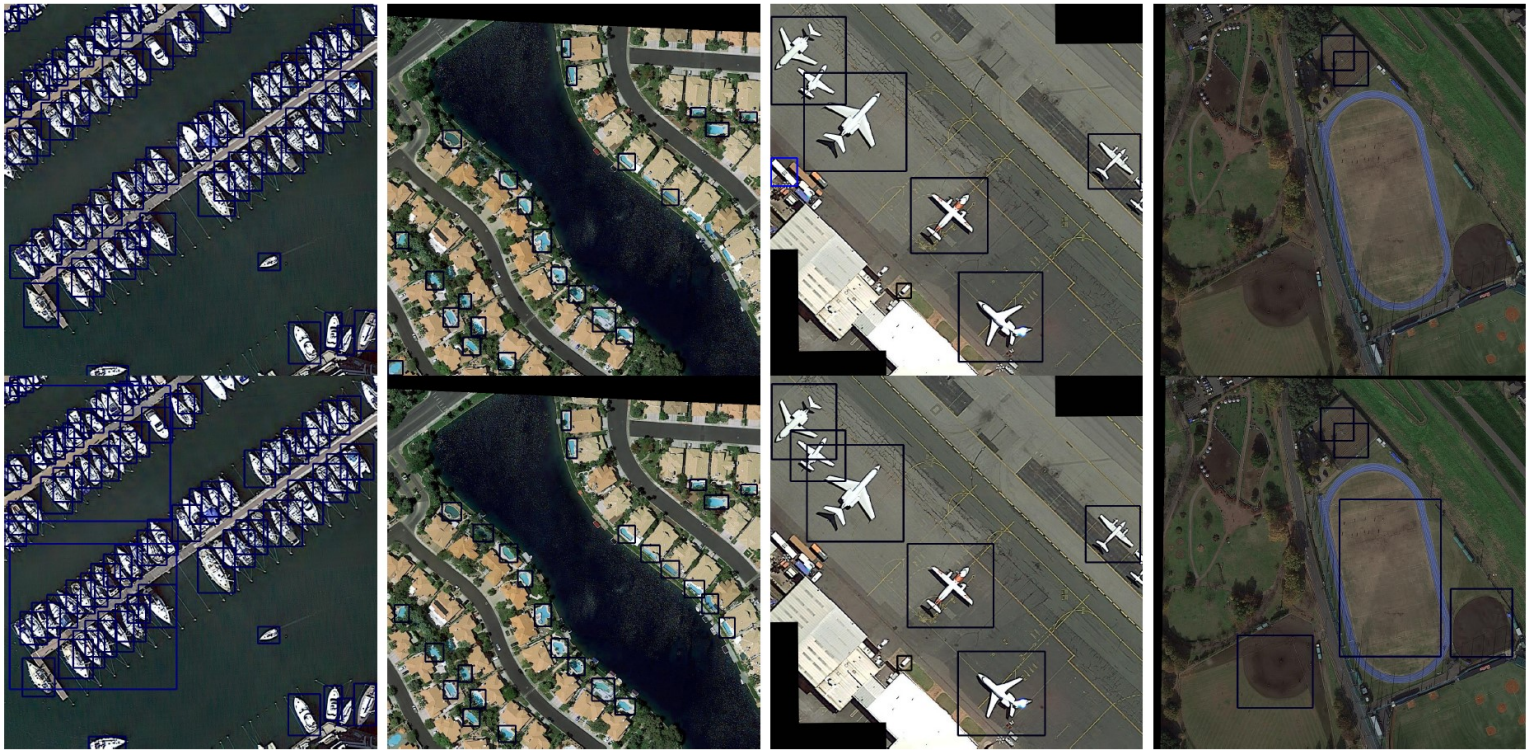

Figure 7. Visual comparison of test results between modified RetinaNet and ASBL-RetinaNet (threshod =0.5). The top images are output of modified RetinaNet, the bottom ones are ASBL-RetinaNet. The first 3 columns show the improvement of different scales of objects with crowded and complex background using our proposed ASBL. The 4th column shows the improvement of simpler images using ASBL.

3 IMPROVING BIRD RECOGNTION IN AERIAL IMAGES USING DEEP LEARNING

3.1 **Abstract**

In computer vision, significant advances have been made in recent years on object recognition and detection with the rapid development of deep learning, especially deep convolutional neural networks (CNN). The majority of deep learning methods for object detection have been developed for large objects and their performances on small-object detection are not very good. This chapter contributes to research in low-resolution small-object detection by evaluating the performances of leading deep learning methods for object detection using a common dataset, which is a new dataset for bird detection, called Little Birds in Aerial Imagery (LBAI), created from real-life aerial imagery data. LBAI contains birds with sizes ranging from 10px to 40px. In our experiments, some of the best deep learning architectures were implemented and applied to LBAI, which include object detection techniques such as YOLOv3, SSD, and RetinaNet, in addition to small instance segmentation techniques including U-Net and Mask R-CNN. Model analysis based on bird detection problem are discussed in this chapter. Among the object detection methods, experimental results demonstrated that RetinaNet performed the best across all the models. Among small instance segmentation methods, experimental results revealed U-Net achieved slightly better performance than Mask R-CNN.

3.2 Introduction

Object detection is one of the crucial tasks in computer vision. In the past few years, the performance of object detection [26-39] has dramatically improved due to the success of deep convolutional neural networks (CNN). Typically, object detection and recognition involve two steps: first, deep neural networks are used to localize the potential location of each target object; then, objects are classified into appropriate classes. If the first step can effectively localize the potential object, the second step will be easier. Even though the two-step approach achieved state-of-the-art performance, the running times are usually slow [36]. Therefore, one-stage detectors have been developed to improve the speed.

Small-object detection remains challenging because small objects usually have lower resolution and less context information. Finding a 20 × 20 size object located in a 5000 × 5000 image is a difficult task, even for humans. As described in the literature, state-of-the-art methods for object detection usually performed poorly on small objects [36]. Recent research has shown the importance of context information and scale for small-object recognition [33][34]. In addition, it has been reported that lower-layer features extracted from CNNs are very useful for small-object detection and segmentation [33-46].

The work presented in this chapter focuses on low-resolution small-object detection by evaluating the performances of leading deep learning methods using a common dataset, which is a new dataset for bird detection, called Little Birds in Aerial Imagery (LBAI). This dataset was created from real-life aerial imagery data, provided by the Illinois Natural History Survey at the University of Illinois at Urbana-Champaign. LBAI contains images of waterfowl and other water birds in shallow lakes within the Illinois River Valley. LBAI

includes different colors, shapes, poses, resolutions, and bird sizes range from 10px to 40px. The dataset contains different backgrounds of rivers, vegetation, land, and mixtures between each type of background. Overall, LBAI captures the diversity of real-life situations for bird detection in shallow lake and wet lands across the Midwest. Some of the birds have larger sizes, in higher resolutions and homogenous backgrounds, which make them easier to be identified. While others have smaller sizes, in lower resolutions with blurry contours, making them hard to be detected. LBAI is designed to identify the difficulties and improve existing methods on small object detection.

Using the LBAI dataset, we compared a wide-range of representative state-of-the-art deep learning methods. The results shed a light on the strengths and weaknesses of different deep neural network architectures for small object detection. The contributions of this research include applying and adapting leading deep-learning methods to the LBAI dataset, evaluating performances of these methods on a common benchmark dataset for small-object detection and segmentation, and automating the time-consuming process of manual image processing from waterfowl surveys.

3.3 **Related Work**

Two major approaches are popular in the detection domain. The dominant approach in modern object detection is based on a two-stage approach which generates a set of proposed targets and detects the bounding box and label for each proposed region. The second approach is using one-stage model applied over regular and dense sampling of object scales, locations, and aspect ratios to generate the location and label for each target

object. Both approaches can be used in aerial image object detection; however, due to the special character of aerial images, a more suitable design of the experiment is necessary.

There are two major approaches for object detection and recognition. The first detection-based approach is the traditional one that generates a bounding box of the detected objects and then identifies the type of objects. The second approach, which is a segmentation-based approach, can also be used for object detection. This approach first generates labelling at the pixel level and then tries to identify the class of the objects to which each pixel belongs.

3.3.1 **Object detection methods**

Existing deep learning algorithms for object detection falls into two categories: one-stage detectors and two-stage detectors [26-34]. At first, two-stage detectors generate many region proposals, which may potentially contain the objects. Then, these sparse proposals are further classified into different object categories. In general, two-stage detectors are more accurate, but slow when compared to one-stage detectors. In one-stage detectors, the bounding boxes proposal step is eliminated and both, object localization and classification, are done in one pass. This strategy significantly improves the speed of detection when compared with two-stage detectors.

In a two-stage detector, the regions that potentially contain objects are first proposed. Then, detection refinement is applied to classify proposed regions and regress the bounding box location. For example, the Selective Search method [26] is used in R-CNN [27] to generate category-independent region proposals to localize the regions that may contain the target objects. R-CNN then uses a convolution neural network to refine

regions. Each region proposal is fed into the CNN independently, which is a slow process. Fast R-CNN [28] addressed these issues by only computing the convolutional feature map once. Therefore, each region proposal shares the computation of the same feature map. The region proposals are generated in a Region of Interest (RoI) pooling format to feed into fully connected layers [28].

Faster R-CNN [29] further improved the detection speed by using a fully convolutional network, called Region Proposal Networks (RPNs), to generate the region proposals, replacing the Selective Search method used in previous methods. In the second stage, a CNN is used for proposal refinement and object classification. The main benefit of this design is that the RPN shares the same convolutional layers with the object detection network, which reduces the detection time [29]. Furthermore, FPN was proposed to improve Faster R-CNN [37]. FPN used the concept of a feature pyramid. Instead of applying a pyramid on the input images, it used the feature map pyramid, since CNN already provide a hierarchy between the different feature layers. The idea was implemented in a bottom-up and top-down path with lateral connections. FPN utilized a lower, high-resolution feature layer, compared to other algorithms, which dramatically improve detection accuracy, especially for small objects.

In one-stage detectors, the proposal generation stage is removed, resulting with localization and classification being performed in one stage. Recently, YOLO [30][31], SSD [32], and their variations achieved promising results. YOLO [30] divided input images into 7×7 cells and each cell predicts two bounding boxes. The network has convolutional layers followed by fully connected layers. Even though YOLO achieved a 45 frame per second detection speed, which is extremely fast when compared to other

algorithms, the main drawbacks result from localization prediction errors and low object detection recall [30].

DSSD [35] is a variation of SSD [32]. It improved the performance of SSD, especially for small objects, by using a larger network as well as adding additional context information with de-convolutional neural networks. DSSD achieved higher accuracy, especially for small objects. Recently, RetinaNet [38] achieved state-of-the-art results for one-stage detection. It outperformed existing two-stage detectors while maintaining a fast detection time. The work in [38], found that the accuracy gap between one-stage detectors and two-stage detectors was mainly due to the positive and negative examples being highly unbalanced, since there are extremely large amounts of background examples overwhelming the process. Even though each loss of the background examples is small, the large number of the background cases result in dominating the total loss, which results in a degenerated model. This problem was solved by introducing a new loss function, called focal loss, to change the weights between positive examples and negative examples, so they cannot affect the loss function dramatically. Huang et.al. in [36] compared the detection accuracy and detection time between two-stage detectors and one-stage detectors. They concluded that, on average, one-stage detectors are faster than two-stage detectors, while two-stage detectors tend to be more accurate than one-stage detectors. The performance of most detection algorithms dropped dramatically when applied to small-object detection. In addition, several one-stage detectors were developed for small face detection, including Tiny Face [33] and SSH [34].

3.3.2 **Instance segmentation methods**

FCN [40] is one of the first methods that use CNNs in the semantic segmentation area. FCN employs CNNs without fully connected layers, which allows the input image to have an arbitrary size. This method laid the foundation for later methods.

A key issue of segmentation methods is the pooling layers. Adding pooling layers can reduce the computation time and increase the reception field size. U-Net is based on FCN [40], with the encoder-decoder architecture to address the issue of determining the appropriate numbers of pooling layers. It has a U-shape architecture to balance the trade-off between good localization accuracy and efficient context information. Therefore, it only needs a small number of training images. In the encoder stage, it uses pooling layers to gradually reduce the layer size, whereas, in the decoder stage, it uses up-convolution to gradually increase the layer size. Moreover, U-Net uses the short-cut connection from encoder to decoder to help the decoder recover fine-grain information. Regarding the trade-off between reception field and localization accuracy, large reception fields lead to lower localization accuracy. On the other hand, when the reception field is too small, the localization accuracy may also decrease due to the lack of context information.

Mask R-CNN [39] is a recent work based on Faster R-CNN and FCN. Faster R-CNN already provides two predictions: bounding box localization and recognition. Mask R-CNN added the third output on top of Faster R-CNN, which is the instance mask prediction for segmentation. The Mask R-CNN architecture can output bounding box localization, classification, and segmentation at the same time. The improvement of Mask R-CNN from FCN comes from the new ROI-Align layer, multitask training, and a better backbone network [39] [40].

3.4 LBAI Dataset

3.4.1 Dataset overview

The LBAI dataset was provided by the Illinois Natural History Survey at the University of Illinois at Urbana-Champaign. The total dataset has 230GB of data, with 440 high-resolution images that have a resolution of 5760 × 3840, and an altitude value of approximately 90 meters above ground level (AGL). LBAI has the cropped images, with different color, shape, resolution, background, and scale, as shown in Fig 8. Due to the large size of images, it is difficult to train CNNs directly on the original images.

LBAI: There were 336 images with high-resolution that were used, and this dataset was divided into the training, validation, and test sets based on these images. For each set, we take the original image and crop it into the small patches with a size of 512 × 512, without overlapping the patches. This will insure that the original image does not get put into different sets (e.g. a patch from the original image gets put into the training set and another patch from the same original image gets put into the validation set). The incomplete boundary regions were discarded after cropping, since resizing may change the ratio and shape of the birds. For the training set, there are a total of 3,158 cropped images with 24,836 birds. We only keep the small patches with birds in the training and validation set. However, the test set contains all the cropped images, both with birds and without birds. After applying the various object-detection methods on the cropped images to detect the birds, the detection results from the patches were merged back into the original images. In our experimental results on this dataset, the performance comparisons of the various methods are based on the merged patches which form the original images.

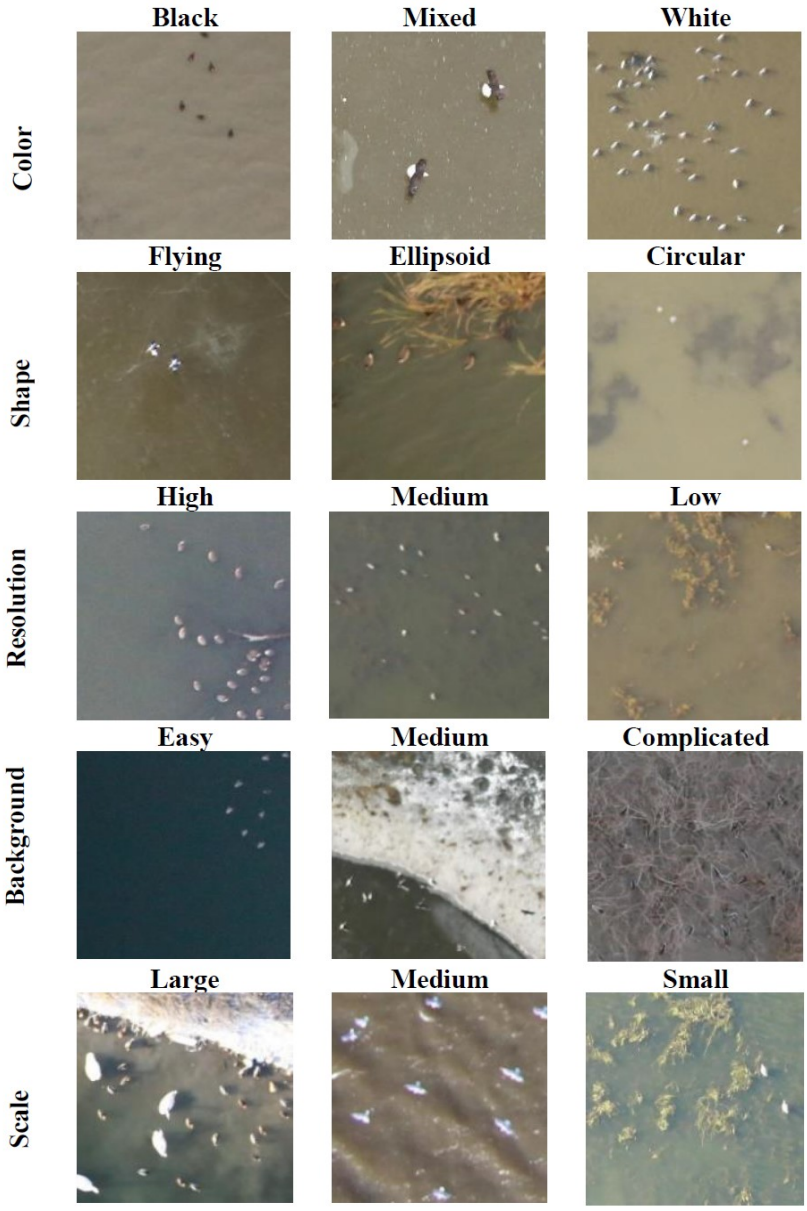

Figure 8 Examples of the new LBAI dataset for small object detection and instance segmentation. Cropped images with different color, shape, resolution, background, and scale are shown.

3.4.2 **Dataset labelling**

When we received this dataset, it contained the bird counting labels, i.e. the number of birds per image, from the Illinois Natural History Survey at the University of Illinois at Urbana-Champaign. However, it did not contain the bounding box locations for the birds, which is the labelling needed for detection. We generated the annotations for the birds' location, so that the number of birds would match the total number of birds received from the expert annotations. A labelling tool, called Sloth, was used to label the images. For each image, a dot was put at the center of each visible bird for all of the birds. This dot label was used for blob detection to generate the bounding box and pixel level labels. Next, we used image processing techniques to find the contour of the labeled birds. A bounding box was drawn around the bird's contour to generate bounding box labels. All labeled results are saved in an xml file. These labels were created from multiple observers with varying levels of training and experience.

3.4.3 **Dataset separation based on difficulty levels**

The backgrounds of the LBAI images are very different, which have a significant impact on the bird detection results. Some images have clear backgrounds with uniform colors, which usually correspond to rivers and water. In this case, the main problem is to identify the birds among different colors, shapes, and resolution situations. On the other hand, in the images with backgrounds of land, trees, or vegetation, detection of birds is much harder, even for humans with great eyes. It is hard to distinguish emergent vegetation and submersed aquatic vegetation from birds. Therefore, following ideas from other datasets, we split each dataset into easy and hard cases based on the background. In LBAI-

A, 3,158 images are categorized as easy cases, which contributed 52% of our labeled data, and 2,907 images as hard cases. In LBAI-B, there are 2,416 easy case images and 2,056 hard case images. The proportions of easy and hard cases are 54% and 46%, respectively.

3.5 Model Adaption of DNN Object Detector

3.5.1 Single Shot MultiBox Detector

SSD is a one-stage detector that performs object localization and classification in a single forward pass of its CNN. SSD's network is built on the VGG-16 architecture, with the fully connected layer removed. Instead of using a fully connected layer, several small convolutional feature maps are added on top of VGG-16 to predict the target objects. Moreover, to capture different object scales, SSD generates different scales of feature maps for detection. This will result with two predictions being generated, one predicts the bounding box category and the other predicts the location of the bounding box. At the end, non-maximum suppression (NMS) is used to generate the final detection results. SSD achieved good accuracy, comparable to two-stage detectors, but much faster. However, SSD's performance on smaller objects was much worse. The reason is that small objects may not appear on higher-level feature maps. Even though increasing the input image size can help slightly, SSD cannot address the problem well.

In our experiments on the new LBAI dataset, we used the source code of SSD built on the Caffe framework with a VGG-16 architecture as the backbone network. VGG-16 is pretrained on ImageNet for image classification and fine-tuned on our LBAI dataset. We used the same data augmentation and hard negative mining as SSD. In addition, we set batch size set to 16 and input image size of 512 × 512. In order to generate promising

results, default anchors are changes based on our LBAI bird dataset, due to the small size of pixels for each bird. In terms of model optimizer, we used Adam with 1e-4 as initial learning rate in order to converge faster than SGD implemented in the original SSD architecture.

3.5.2 **YOLO v3**

YOLO v3 is an improved version of the original YOLO network with several adjustments. It is the modification version of YOLO v2 but keep most of advantage of YOLO v2. In YOLO v2, batch normalization on the convolutional layers is used to stabilize network training. The performance is increased by approximately 2% mAP with batch normalization. As found by other research, higher resolution can capture more information, especially for small objects [30]. This strategy increases the performance by approximately 4% mAP. In YOLOv2, the fully connected layers are removed. Instead of directly predicting the location of bounding boxes, YOLOv2 adopted an anchor box strategy similar to that used by Faster R-CNN. This can improve the recall by a large margin while only slightly lowering precision. A dimension clustering algorithm is used to find the starting anchor box dimensions based on the data from the training set. With dimension clustering and direct location prediction, the location accuracy is improved by over 4%. Finally, for improving performance on small objects, the lower feature map is concatenated with the higher feature map. In YOLO v3, multi-scale strategy is used to improve the performance from YOLO v2.

In our experiments on LBAI, we used the source code for YOLOv3 built on the Darknet framework. We loaded weights from pretrained weight generated by COCO

dataset and fine-tuned them on the LBAI dataset for 16,000 batches. We changed the number of output classes to one and adjusted the last convolutional filter to 30. The network was trained with a batch size of 64 and subdivisions set to 8. We applied a jitter of .4 to the training set and used a resolution of 512 × 512 without any randomization. We set the learning rate to 0.0001 with a decay of 0.0005. In terms of anchor sizes, we use k-means to pre-calculated box aspect ratio of training data. For different scale of feature map, we put box sizes which is 20 *20 into scale information.

3.5.3 **RetinaNet**

Retinanet is current state-of-the-art one stage detector using deep learning. RetinaNet uses FPN as feature extractor and then feed all the convolution features into classification and box subnet. For each convolution feature, it uses anchor box to make prediction, for each cell in the feature map, it generates 3 different aspect ratios and 3 different scales of anchor boxes. For each anchor, it will make prediction using subnet. The classification loss is using Focal Loss instead of normal cross-entropy to solve out unbalanced classification problem. Regression loss are using L2-smooth loss. The final objective loss function uses Focal loss + Regression loss with same weight.

In our implementation of RetinaNet, modification is necessary to get better performance based on our LIBAI data. Specifically, we keeped aspect ratios to {1:2, 1:1, 2:1}, but changed anchor sizes to $\{2, 2^{0.5}, 0.3\}$. The reason is that all of objects in the data, which is waterfowl, are relatively small compared with raw images so that small anchor will provide better precision information and features on localization and classification. In our training, we trained the whole network instead of fixing any parameter in the network.

The optimizer is using Adam optimizer with 1e-4 as starting point, the learning rate decay is 0.1 for each 7 epochs.

3.6 **Model Adaption of DNN Instance Segmentation**

3.6.1 **U-Net**

U-Net is built on fully convolutional networks, specifically designed for biomedical image segmentation. In the contracting path, the convolutional layers are applied with pooling layers to extract context features. In the expanding path, the up-sampling layers are added to increase the localization accuracy. More importantly, the feature maps from the contracting path are concatenated with the up-sampling layers to improve localization. In addition, elastic deformations are applied as data augmentations during training. U-Net is the winner of the ISBI challenge for segmentation and the ISBI cell tracking challenge in 2015. With a 512 × 512 input image, the inference time is less than one second.

In our experiments, the basic U-Net architecture was used to train on the LBAI dataset. However, because of the significant difference between the natural images in LBAI and bio-cell images, we added zero-padding after each convolution operation block, instead of cropping the reception field as in Isola [49] and Zhu's work [50]. This prevented the network from losing too much pixel label information, which was needed because objects in LBAI are very small. With padding, the U-Net architecture would have the same size of input and output.

In order to apply the segmentation method for object detection, instance segmentation labels were prepared as the ground truth. However, it is time and labor consuming to generate segmentations for every target object in LBAI. So, instead of using

object contours as labels, we used a 20 × 20 square as a ground truth mask, centered at the coordinate of each object. After fixing the network architecture, specifically the inputs and outputs, we fed 512 × 512 images into U-Net and trained the network. In the training phase, we used the VGG-16 pretrained weights on ImageNet [46] as initial weights in all encoder blocks and the Xavier initializer in all decoder blocks. The learning rate was set to 0.001 with a learning rate decay equal to 0.1 for every 7 epochs. The batch size was set to 2 in the training phase and since we were using a GTX 980M GPU with 8GB memory, the Adam optimizer was used. In the inference phase, blob detection on the final output was used to calculate the coordinates after running through the segmentation network.

3.6.2 **Mask R-CNN**

Mask R-CNN is a recent work for segmentation and object detection, as explained in the related work section. The major change in Mask R-CNN is that it solves the ROI pooling problem that causes the feature maps and original image not to be aligned. Mask R-CNN uses the ROIAlign layer to replace the ROI pooling layer. In the ROIAlign layer, the rounding for boundaries or bins are removed and bilinear interpolation is applied to compute the exact values for the feature maps. Moreover, Mask R-CNN uses a binary cross-entropy loss for the instance mask, which avoids the competition among all classes. Mask R-CNN achieved state-of-the-art results of instance segmentation on the COCO [47] test dataset with a running time of 5 FPS.

When implementing Mask R-CNN on the LBAI dataset, we used the same input and output described in the U-Net implementation. In the training phase, we froze the weights of ResNet-101 and trained all the other weights in the original Mask R-CNN

architecture. For the hyper-parameters, we used the Adam optimizer with the value of 0.0001 as the learning rate until the loss curve converged. Other implementation details for the training and inference phases were the same as U-Net for a direct comparison between these two methods, e.g. batch sizes and blob detection.

3.7 Experimental Results and Analysis

In this research, we evaluate the detection results and the counting results of various deep learning methods on a common dataset, the LBAI dataset. For detection, the performance metrics include precision, recall, and F1 score. Precision is the percentage of correctly predicted instances over the total number of predictions, while recall is the percentage of correctly predicted instances over the total amount of instances, defined as follows:

$$Precision = \frac{tp}{tp + fp} \tag{11}$$

$$Recall = \frac{tp}{tp + fn} \tag{12}$$

where tp is true positive, fp is false positive, and fn is the false negative instances.

F1 is the harmonic mean of precision and recall:

$$F1 = 2 * \frac{precision * recall}{precision + recall} \tag{13}$$

For counting results, the performance metric is the mean absolute error (MAE), i.e., the difference between the predicted count of birds in an image and the true count based

off the labels described in the previous section. Using LBAI-A, we compared the performance of five representative state-of-the-art deep learning methods, including YOLOv3, SSD, RetinaNet, Mask R-CNN, and U-Net.

Table 7. Performances of object detectors on the EASY CASES in the LBAI-A dataset.

Methods	Precision	Recall	F1 Score	MAE
YOLOv3	0.887	0.909	0.898	23.6
SSD	0.202	0.869	0.328	155.2
RetinaNet	0.906	0.917	**0.912**	**17.8**
Mask R-CNN	0.772	0.842	0.805	49.0
U-Net	0.861	0.781	0.819	38.5

The results on the test cases are shown in Table 7 and 8. There is a total of 944 test images for the easy cases and 944 images for the hard cases in LBAI-A. As shown in Table 7, on the easy cases in LBAI-A, RetinaNet obtained the highest F1 score, 91.2%, which were much higher than the other 4 methods. In terms of MAE, RetinaNet was much better than the other methods, outperforming them by at least 53%.

Table 8. Performances of object detectors on the HARD CASES in the LBAI-A dataset.

Methods	Precision	Recall	F1 Score	MAE
YOLOv3	0.568	0.238	0.335	22.0
SSD	0.182	0.534	0.213	105.3
RetinaNet	0.595	0.572	0.582	**19.8**
Mask R-CNN	0.193	0.659	0.299	89.5
U-Net	0.55	0.51	0.53	20.1

As shown in Table 8, focusing on the hard cases in LBAI-A, the precision, recall, and F1 scores of all methods were much worse than their corresponding results on the easy

cases in Table 8, however, the best model is still RetinaNet, regarding F1 score and MAE score.

Based on the results shown in Table 7 & 8, models with feature pyramid network (FPN) architecture, like YOLO and RetinaNet, outperformed with the model without it, even though the model extract different levels of features, like SSD. The reason is that the features extracted in FPN are finetuned by higher level of features and reconstruction from lower level features. It will generate more robustness features of objects. However, even though the model with different scale of features, like SSD, also extract small objects information and features, the performance of small objects, like bird, are sensitive to the extracted features so that it makes the poor performance on LIBAI dataset.

In terms of instance segmentation models, the UNet outperformed with Mask R-CNN, the main reason is similar as DNN object detectors, which is the small objects, like bird, are sensitive to the features extracted from CNN. Across two tables, the performance of Mask R-CNN drops more on the hard cases compared with other 4 models. The classifier in Mask R-CNN won't provide too much help on final performance if the features extracted are bad and noisy.

3.8 **Conclusion**

In this chapter, we have presented a new aerial imagery dataset based on real-life images including waterfowl and other water birds in wetlands around the Midwest. Different from most of the existing datasets, the new LBAI dataset contains small birds of sizes ranging from 10px to 40px. Several state-of-the-art deep learning object detection and instance segmentation techniques have been applied to the LBAI database and obtained

a range of performance results. Among object detection methods, RetinaNet performed the best on both cases. Between instance segmentation methods, U-Net achieved better performance than Mask R-CNN. These results are useful for identifying the strengths and weaknesses of existing methods and the development of future methods with improved performance.

4. NEW DEEP LEARNING BASED AUTOMATIC DETECTION OF ALCOHOL USAGE (DEEP ADA)

4.1 Abstract

With the development of IoT and mobile health, biosensor has been widely used to collect research data. However, in terms of data analysis and prediction of signal data collected by bio-sensor data, it is still a challenging problem because of the difficulty of useful features and information extraction from signal data and the shortage of label data in the experiment. Most feature learning techniques on bio-sensor data are handcrafted features so it may be too arbitrary to select features. In terms of those two problems, in this chapter, we proposed a ADA system (Automatic Detection of Alcohol) which can provides the statistical analysis of bio-sensor data at first. Then, based on the ADA system, we extended it with a novel deep learning based feature extraction method on bio-sensor signal data using deep learning algorithm to predict alcohol usage of real subjects in their daily lives (Deep ADA). The features extracted are based on Convolutional Neural Network without any human intervention and uses a significant amount of unlabeled data to augment the features. The method proposed using deep learning outperformed the other traditional feature extraction methods by 19% accuracy improvement on the real subject's data.

4.2 Introduction

Currently, most methods in clinical psychology research primarily rely on questionnaires and interviews with examiners in the lab setting. With the rapid development of mobile technologies, a new promising solution is a mobile ambulatory

assessment system with real-time data monitoring and collection of real-life subject behavioral and psychology data, as well as physiological data. Ambulatory assessment is the use of field methods to evaluate subjects in natural or unconstrained environments [67]. By combining information about the external environment, and participants' physiological and mental states, collected through system-generated and self-report surveys, machine learning models can be developed to identify changes in mood, alcohol use and/or craving, as well as other psychological problems. This same information can also be applied to context aware applications. In context aware computing, context is information that can be used to describe the state of something that is relevant to a user's interaction with an application [68,69]. Combining methodology from psychophysiological field research with body area wireless sensor networks and mobile devices can improve context aware computing. Mobile systems based on wireless wearable sensors have been actively developed for a variety of applications in mobile health and physiological monitoring. They are capable of continuously collecting bio-sensor and self-report data to assess or predict physical and psychological conditions, such as alcohol consumption, in daily life. Automatically identifying patterns of interests based on various physiological signals and survey results for each individual remains a challenge.

In recent years, deep neural networks have achieved huge success in many areas of computer vision, such as image classification [82]–[84], and object detection [85]–[92]. The reason why CNN has a breakthrough improvement is that it can generate multi-layer features with better representation of input data to make classifications. In addition to the development of computation power, deep learning has been widely used in popular classification problem, like image, audio and speech. However, to our knowledge, few

efforts have been made in trying to identify, using deep learning, certain attributes about ones current state of well-being based off of body sensor data, such as heart rate. There are three reasons. (1) Noisy information. The raw sensor data collected from bio-sensor is too noisy to make prediction. The raw data may be affected by human emotion, activity or other environmental factors. Most of researchers in bio-senor may focus on extracting useful features using domain knowledge from pre-cleaned signal data and then make predictions [93], [94]. (2) Hard to explain features using deep learning. Feature extraction from sensor data or transformed sensor data is very necessary since the raw sensor data is too noisy to get better performance. In terms of feature selection, since the CNN based model is hard to explain, the majority of researchers mainly used CNN as black box to make classification for other specific research problem. (3) Lack of labeled data. Subjects' action in daily life for the study is very hard to collect, for example, the drug use study [95]. Basically, deep learning may need to be fed by large-scale input data to have better performance on classification. This requirement may not be applied to small samples data. The goal in this chapter is to solve out the problems of bio-sensor data classification using deep learning. We propose to use a deep learning approach to predict whether or not someone has consumed alcohol based off their physiological sensor data, by extracting features using 1D CNN deep learning model and then feeding useful features which can reconstruct raw 1D signal into machine learning model to make classifications. Instead of using CNN as black box, multiple tests in architecture of CNN have been applied. An accurate prediction model for alcohol consumption would be very useful and open avenues into research where self-reporting of alcohol consumption would no longer be necessary, giving more accurate prediction results. In our study, all the experiments are based on

sensor data collected from a newly developed mobile ambulatory assessment system for automatic detection of alcohol usage and craving, ADA [96]. We feed our proposed models into ADA system to have a better prediction performance. Furthermore, it acts as proof that useful features of waveforms outside of audio can be extracted by CNNs and outperformed with other traditional hand engineer features. The findings in this work are only the beginning of this area of research that will continue to expand. In terms of evaluation of performance of the model with our proposed method, we collected 16 real subjects with multiple drinking periods to make classifications. Our experimental results show the features extracted by deep learning have 19% accuracy improvement compared with other feature extraction methods.

4.3 Related Work

Deep learning for classification task has been widely used in multiple domain, including image and audio. However, due to the limitation of physiological bio-sensor data, raw input with deep learning classifier may not be a good solution. Feature extraction from raw input is necessary. In this section, we will discuss the existing work and its potential problem for feature extraction of physiological sensor data and the solution of few labeled data.

4.3.1 Physiological sensor data collection and analysis

Wireless body-area sensor networks have been a hot topic and used for a variety of applications in mobile health, physiological monitoring, and context aware computing. Mobile systems have been developed to continuously collect biosensor and self-report data to assess or predict psychological states [73,81]. For example, in [73], the iHeal project

uses a biosensor that measures electro-dermal activity, motion, temperature, and heart rate to attempt to identify substance cravings. When the system detects a change in sympathetic nervous system activity, it collects information from the biosensor and self-reported information from the user about stress, cravings, activities, and other various information.

Self-assessment of emotion, usually through surveys, provides important, yet oftentimes inaccurate information [78]. In lab experiments in [72], users correctly self-assessed their own stress only 84% of the time. To try to explain the incorrect self-assessments, humans do not necessarily experience emotions in a binary way. For example, a person can experience different degrees of happiness at different times. Another problem with self-assessed psychological information in a natural environment is there is little control over the participants' physical and social environments, unlike in a laboratory setting, which makes the ability to identify the participants' contexts critical [79].

Research to identify drug use in daily life is also being actively pursued. In [80], experiments and analysis were done separately in field studies and in labs. Mathematical models were built to predict if the subject has used cocaine. A main difference between detecting cocaine use and alcohol use is that the effect of cocaine is much greater and sharper on the human body when compared to alcohol use or smoking cigarettes.

4.3.2 **Feature Engineer of Physiological Sensor**

In recent years, bio-sensor data, like EEG, ECG, are used to analyze human activity and prevent human disease, like sleeping quality assessment [97], [98] and disease detection [93]. Most of their model pipeline uses preprocessed data to transfer the data into different format using data transformation, like spectrogram [97], [98] or FFT

transformation [97], [98], and then use statistical features to represent the transformed data to make predictions. However, feature extraction based on transformed data may lose raw data information and selection of transformation methods is arbitrary for different domains of bio-sensor study. To avoid problems of transformation, feature engineering on raw data is also provided by multiple bio-sensor research studies [99]. Most of them use statistical features, FFT features and other domain features for each window of raw data to establish feature pools and then feed them into machine learning pipeline to make predictions [94], [100]. This method may incur several problems: (1) high dimension of features; (2) domain knowledge required; (3) arbitrary feature selection. Even though it may have promising results on some domains [94], [98], [100], it may not be the most robust solution to other biosensor studies.

4.3.3 **Few Labeled Data**

Another common scenario happening in bio-sensor data is few labeled data problems, especially in drug detection [95]. Most of the successful model using deep learning on biosensor data are based on large scale labeled data to train a better model using supervised learning. Deep learning on classification needs large scale labeled data, like COCO [101] and ImageNet [102]. On time series domain, most of the promising results are generated from many labeled events' data using deep learning [103]. In order to solve out this problem, feature extraction of raw input sensor signal data using unsupervised learning [104], [105] and few-shot learning [106] are promising methods. Recently, auto-encoder for feature extraction [107] are used to work on MFCC 1D signal data. The feature extracted from bottle-neck layers of Deep Neural Network will be used to feed into other

machine learning classifiers which are good at working on few labeled data problems. However, it may incur two important factors of performance for feature extraction using auto-encoder. (1) performance of reconstruction. Since the complexity of input data, it may not get good performance for all types of signal data. If the reconstruction is bad, represented features from bottle layers may not be useful for prediction. (2) Dependency data. In the time-dependency or spatial dependency data, most of auto-encoder does not consider the correlation of each input data point. Most of them are treated as independent points and it may lose information of correlation.

4.4 Automatic Drinking Analysis (ADA)

ADA is a data analysis and machine learning system designed to investigate the relationship of many factors related to alcohol use, including participants' activities, emotional states, emotion dysregulation, and surroundings in order to better understand the conditions and triggers of alcohol usage and craving. All sensor and survey data are first cleaned and then analyzed or run through machine learning methods. Next, two main components of ADA will be discussed in detail. One is data preprocessing that cleans sensor data and survey data automatically. The other is statistical data analysis and visualization, enabling domain experts to understand the data and perform their investigation and discovery.

4.4.1 Sensor Data Cleaning

The physiological data obtained using mAAS came from the Affectiva Equivital EQ2 sensor and Hexoskin Wearable Body Metrics. Due to the project's integration of multiple

sensor data sources, the raw data are heterogeneous, which needed to be addressed prior to analysis. A few issues were:

- Sampling frequency of accelerometer data did not match heart rate, breathing rate, and RR interval.
- The EQ2 sensor exported multiple files for each metric, each containing features needed for analysis.
- The sensor would generate a new file, whenever the user would take off the sensor, therefore each day of data collection included a different number of files.
- Timestamps were not specified in a uniform format between the EQ2 sensor and the Hexoskin.

The data cleaning module corrects mismatched data format, removes outliers and missing values, filters out noisy data, and smooth out the data using regression.

Fig. 9 shows an example of a patient's sensor data of heart rate, breathing rate, activity, and skin temperature in one day. The data are noisy with large fluctuations and missing data (points with 0 values).

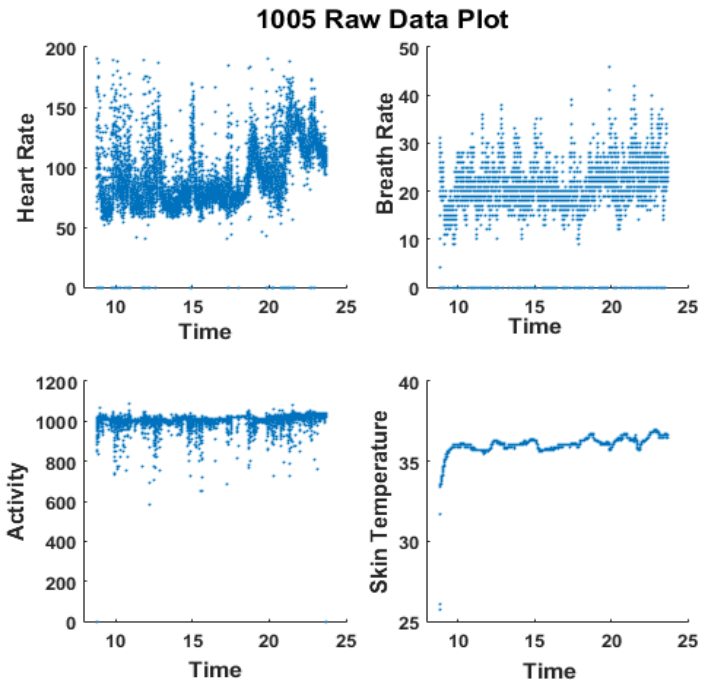

Figure 9. Raw signal visualization

After removing all missing values, Loess Smoothing Model, a locally weighted method, is applied to smooth out the noisy data. Equation (14) shows the weighted function of Loess Smoothing:

$$w_i = (1 - |\frac{x - x_i}{d(x)}|^3)^3 \qquad (14)$$

Where x is the predictor value associated with the response value to be smoothed, xi are the nearest neighbors of x as defined by the span, and d(x) is the distance along the abscissa from x to the most distant predictor value within the span. In this study, a span parameter of 0.01 is chosen, since the observation number is large enough. After fitting Loess model for the four types of data, the outliers were detected using a 95th percentage confidence interval. In Fig. 10, the red crosses are the outliers detected by Loess model, the black solid line is the Loess fitting result, and the blue points are smoothed data.

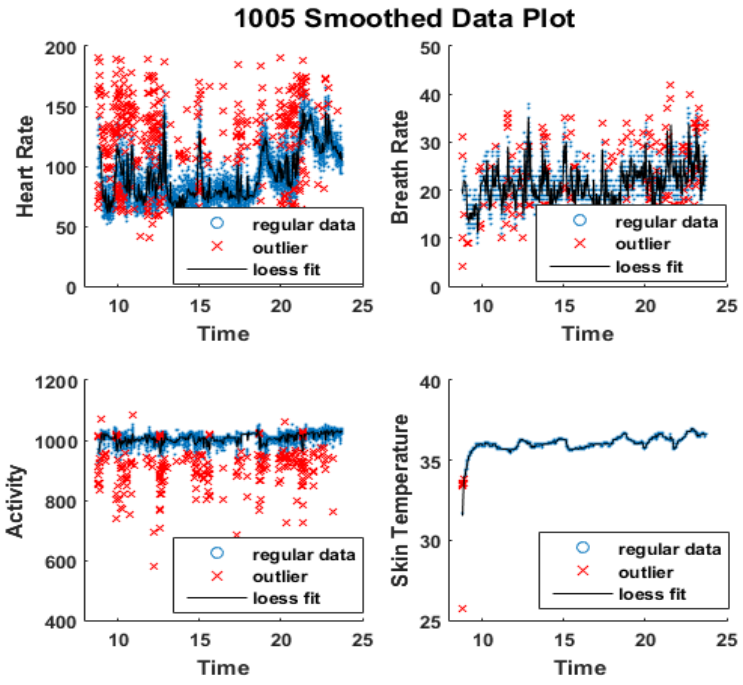

Figure 10. loess fit and outlier remover for physiological signal

To find the underlying tendency of the smoothed signals, a moving average, median filter, and a smoothing spline were applied. Equation (15) shows the object function of smoothing spline spline s:

$$\min \; p\sum_i w_i(y_i - s(x_i))^2 + (1-p)\int(\frac{d^2s}{dx^2})^2 dx \qquad (15)$$

Where p is the smoothing parameter between 0 and 1, w_i is the weight, and x_i and y_i are a training example. If p = 0, this will produce a least-squares straight-line fit to the data, while if p = 1 produces a cubic spline interpolant. The smoothing parameter of 0.5 is chosen for all signals. Fig. 11 shows the result of the fitting given the result shown in Fig. 10. The legend shows the quality of the smoothing line, the R^2 value, has a close correspondence with how noisy the original data are.

4.4.2 **Survey Data Cleaning**

The other type of data in this research is survey data, collected using mAAS's survey module from subjects in the natural environment. While using the smartphone app, the users answer questions during different times of the day. The survey data includes different attributes, such as the type of survey trigger, survey time, user ID, and many different survey questions. For example, the survey questions for mood dysregulation include how much did your mood change, are you in a better or worse mood now than before, and what triggered your mood change. Based on the user's responses, different mood indexes are calculated in automatically.

A goal of the research is to identify when drinking episodes occur in order to predict alcohol usage from the sensor and survey data. A drinking episode is defined as when the subject endorses the activity of drinking alcohol. To determine a drinking episode from survey data, a dynamic moving window searching algorithm was developed.

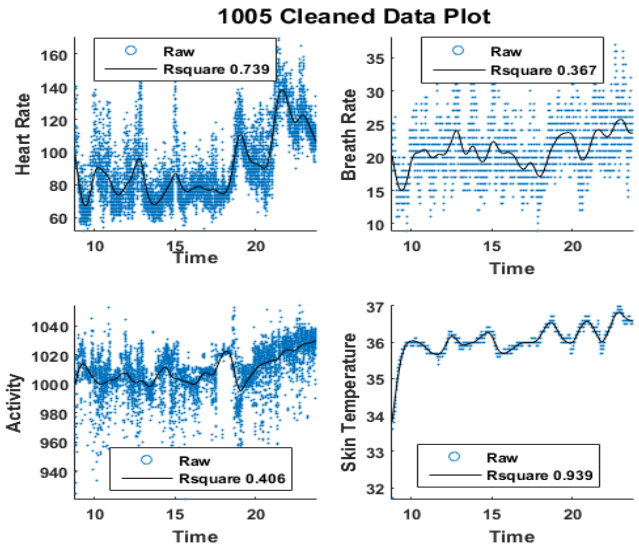

Figure 11. Cleaned physiological signal

Because the survey data report discrete drinking times, which have unknown time offsets from real drink activities, the reported drinking time points are enlarged to a window of time. In our experiments, the window size is 2 hours. If a user has multiple drinking events in one window, they are all considered as one drinking episode. As the window moves, if the user has another drink within the two-hour window, it will be considered the same drinking episode, but the number of drinks and drink times will increase. If the user does not have a drink within the two-hour window, the current drinking episode ends, and the next drinking episode will begin when the subject drinks again. Therefore, the subject may have many drinks and drink times during one episode and there may be multiple episodes in one day. One important variable used in this research is the number of drinks per episode.

4.5 1D CNN for feature engineer

Due to the limit and problem of 1D bio-sensor physiological data, current work mentioned in related work may not be the best solution. In our section, we proposed a novel deep learning feature extraction on 1D bio-sensor physiological data. This method mainly uses Convolution Neural Network and keeps encoder-decoder shape to extract features of raw input signal data without any transformation or information loss. The convolution operator will consider the information of time correlation and encoder-decoder shape will help to extract useful features of raw input. In addition, unsupervised learning for feature extraction will solve out the few labeled data problem in bio-sensor physiological data. After generating features from deep learning feature extraction, it will be fed into machine learning classifiers, like SVM, to make classification and prediction of drinking action of

subjects in real lives. In order to prove the efficiency of our proposed method, real subjects' data generated from ADA [96], which is the data process and analysis pipeline focusing on Hexoskin sensor data, will be used to make prediction.

4.5.1 **Data preparation**

The sensors used, the Equivital EQ2 and Hexoskin smart shirt, collected the sensory data. The data collected was heart rate, breath rate, skin temperature, and activity level, at a frequency of one recording every 5 seconds. We used ADA system [96] to find the drinking episode based on self-report survey data. Every time the user consumes a drink of alcohol, they fill out a survey which then indicates in the data when the drink was consumed. The data is set up so that each row has the associated date and time, followed by the sensor data and survey data. While the survey also records the mood the user is in, the only data used for this research was the time, date, heart rate, skin temperature, activity and instances of alcohol consumption. After generating drinking episode data, we treat them as positive samples. For each drinking episode, we consider each 30 minutes as positive drinking blocks to keep time series correlation information. In terms of negative samples, because there are much more data point than drinking data for each study user, down-sampling method is applied to figure out the unbalanced classification problem. In order to compare the model performance, we randomly selected negative samples from non-drinking days for each user to make it a 50:50 ratio. Finally, in our research, two types of classifications are demonstrated, one is within-subjects, the other one is cross-subjects. Within-subjects case are using 80% one user's data as training data, the remaining data for this user are test data, all the data are sorted by time order to mimic the real scenario. Cross-subjects are

using 80% user's data as training data, the remain users are test data in order to test the generalization across all the people. In our study, there are 214 samples used as training and 50 samples as testing for within-subjects case. For the Cross-subjects case, 212 samples are used as training and other 52 samples are testing from 3 independent subjects. In order to generate competitive results, 4 methods are demonstrated in our experiments, stats-feature engineer, CNN-based feature engineer, ResNet50, and SVM with raw input for classifications.

4.5.2 **Descriptive statistics features**

To describe features of each signal, basic descriptive statistical features are calculated to represent each data block. It is the common way in physiological domain to analyze the data block. Since the complexity and noise of information for each drinking block and some information are redundant for future analysis, basic tendency will be discovered using descriptive statistics. In terms of our physiological data block, we follow the other papers' work [94], [100] to extract mean, standard deviation, covariance, skewness, range, root mean square, zero crossing rate, and mean crossing rate for each signal. In our study, there are 3 signals for consideration, heart rate, skin temperature and accelerator, so there are 24 features in the dataset. Due to the limitation of sample size, we hold the view that 8 features for each signal would be enough for future analysis.

4.5.3 **CNN-based features**

The popular CNN model for signal analysis is to transfer the signal into spectrogram, however, in physiological data, it may have many limitations, because of the low frequency of raw signal. Spectrogram transformation will lose significant information

and no pattern can be recognized. In our experiments, instead of using any transformation, we only work on the raw signals to extract features using CNN. In addition, due to the number of labeled data are limited and lots of unlabeled data are not redundant, we also want to take advantage of unlabeled data to improve the models. We proposed a novel 1D CNN feature extraction for physiological data, the methods we used came from image segmentation. In the image segmentation, encoder-decoder architecture has been much more popular than other segmentation models, like Unet [108] and Segnet [109]. The key idea is to use encoder to down sample the raw input into high level feature description and use decoder to reconstruct the high-level feature into raw segmentation. In our work, we combine semantic segmentation and auto-encoder ideas together, our input of architecture are raw signals and output is same as input. Our goal is to let CNN models reconstruct the raw signal. If performance of reconstruction is pretty good, the bottleneck features in the middle will be very important features to represent the input. As shown in the Fig. 17, Our proposed network keeps encoder and decoder shape with multiple convolution block. For each convolution block, it contains convolution operator with 1*3 kernel, zero-padding and activation function. In our convolution operator, the formula is as follow:

$$Conv_j^l(x) = \sum_{i=1}^{I} (w_{ij}^l \odot x_i^{l-1}) + b_j^l \qquad (16)$$

Where x_i^{l-1} is 1D time-series signal data from l-1 layer, I is kernel size of convolution filter, j is j^{th} output of convolution operator, w_{ij} is weight matrix as convolution filter with I*J dimension, b_j^l is bias vector with j dimension in layer l, is $\odot$ dot production. Based on previous formula, the x dimension will be reduced on the boundary. In case of losing information of boundary, in our implementation, zero-padding on boundary are added in each convolution block. After convolution operator, activation function is applied to the

output of previous formula. In our experiments, multiple activation functions are tested based on our 1D signal data, finally, Leaky Relu is used in our network.

$$LeakyRelu(x) = \max(0.1 * x, x) \qquad (17)$$

where x is the output of convolution operator in each convolution block. Based on this formula, the effect of negative value of input are reduced to next block in network. In order to extract context information of input time series and make it robust to small variations for previous learned features, pooling layer is used in our network. The most popular pooling methods is to computer average value in each neighborhood at different position without overlapping, call Average Pooling. In our network, average pooling with kernel size 1*2 are used in first top 3 convolution block and 1*5 in the remaining blocks in encoder section. In terms of decoder section, the process in each convolution block is same as encoder section. However, the main difference in order to reconstruct the raw input is unpooling layer. Unpooling method is to use extracted features to represent and evaluate the learned feature in the network. To perform unpooling, the position of each maximum activation value when doing max pooling need to be remembered. Then, the remembered position is used for unpooling. In our implementation, we did not remember the index of pooling layer since we use average pooling instead of max pooling. When use unpooling, all the receptive position will be given the same value of each point of input. The kernel size of unpooling layer is same as same level of pooling layer, respectively. In order to make reconstruction is independent with given information, we did not use U-Net [108] architecture which are connect encoder information into decoder section to improve the reconstruction performance.

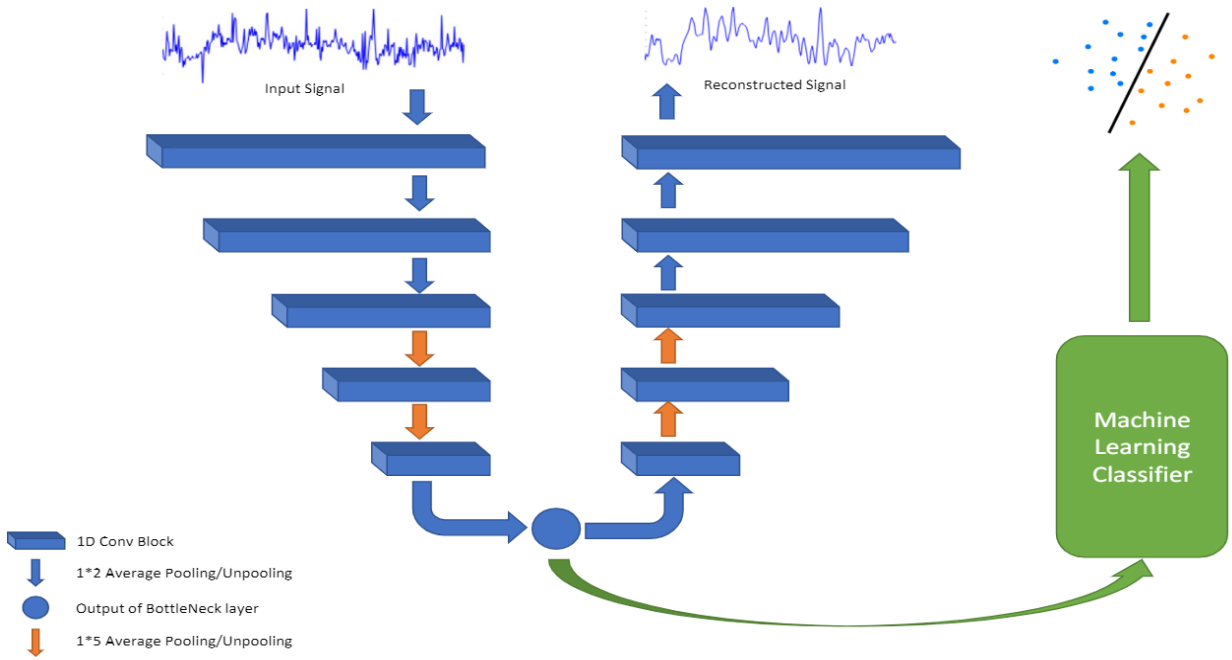

Figure 12 Architecture of 1D CNN feature extraction. All the blue blocks are 1D convolution block with Leaky Relu activation. The blue arrows are pooling/ unpooling layer with 1*2 kernel. The orange ones are pooling/ unpooling with 1*5 kernels. The encoder from top to bottom in the architecture is to extract low level features to represent raw signal. The decode is to reconstruct based on extracted low level features

4.5.4 **Supervised Learning**

To compare with other supervised learning, in our experiments, we also test two popular supervised learning method, ResNet50 and Support Vector Machine (SVM), using the same input. However, ResNet50 usually works on 2D image classification problems so that it cannot deal with 1D signal without any modification. In our implementation, all the 2D convolution operator are modified into 1D convolution operator. All other parameters keep as same as original architecture. In terms of SVM, we considered each data point in the block as one feature instead of extracting any features.

4.6 **Experimental Result**

4.6.1 **ADA Survey Data Analysis**

Table 9 shows the basic drinking statistics of 16 subjects, including the number of alcohol drinks, number of drinking activities, number of drinking episodes, and how many drinks in each episode for each subject, which represents different levels of alcohol use. All these numbers vary significantly from one person to another.

Table 9. statistics of survey data of all subjects in alcohol craving study

ID	Number	Times	Episode	Number/Episode
1001	39	34	17	2.29
1003	6	5	2	3.00
1004	69	56	17	4.07
1005	21	16	8	2.63
1007	27	19	13	2.08
1008	23	9	5	4.60
1009	23	16	8	2.88
1010	3	3	3	1.00
1013	3	3	3	1.00
1014	29	23	11	2.64
1017	16	14	7	2.29
1019	45	38	18	2.50
1020	17	15	7	2.43
1021	30	21	7	4.29
1022	23	19	7	3.29

Next, we divided the survey data for each subject into two categories: drinking and non-drinking days' data. If a person had at least one drink in a day, this day is considered as a drinking day of this person. Otherwise this day is a non-drinking day.

Fig. 13. shows an example for the analysis of drinking versus non-drinking day's data. The graph shows how many drinking and non-drinking days for this subject in the title and how the mood changes for all days. The bar plots give the mean value of the number of drinks, number of drinking activities, and the number of drinking episodes for all days. Moreover, the different colors of each line demonstrate five different moods and how mood changes for each subject over time.

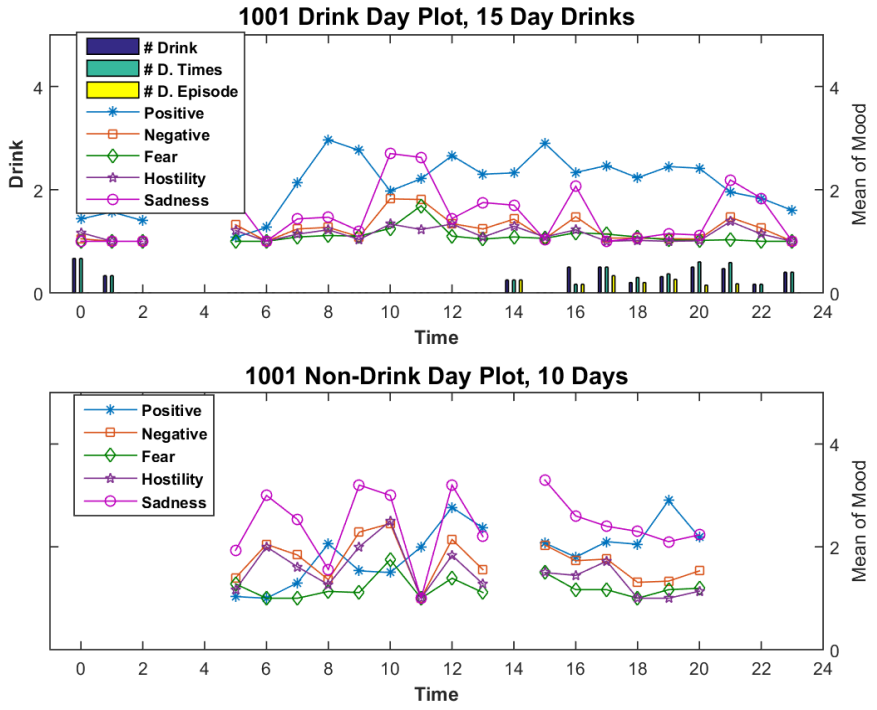

Figure 13. Graph of subject 1001's survey data. (day comparison)

When comparing the plots of drinking and non-drinking day's data, the mood level and mood changes are very different between these two plots. In Fig. 13, the Positive mood slightly decreases over time on drinking days but increases over time on non-drinking days. In addition, the level of different emotions for drinking and non-drinking days is different. The level of Sadness from 15 to 20 for non-drinking days is greater than 2 but it is less than 2 for drinking days.

Fig. 14 shows box plots for two subjects. They compare the distribution of mood data in drinking with non-drinking days. It is apparent that the distribution of some mood data for drinking days is significantly different from the distribution of mood in non-drinking days. In addition, the two subjects have significantly different overall emotional levels. The distribution of sadness for subject 1001 for non-drinking days is remarkably

larger than that for drinking days. The median of level of sadness during non-drinking days for subject 1001 is 2.5 but that value is 1.0 for 1019.

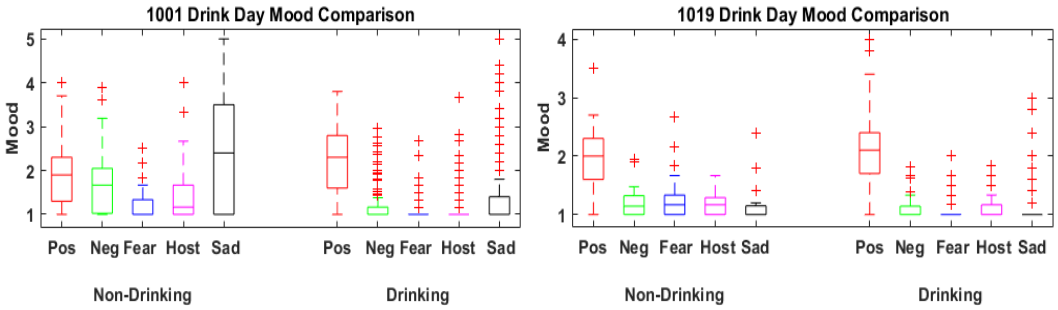

Figure 14. Box plots of two different subjects' survey data (drinking day)

Next, we investigated mood and drinking changes between drinking and non-drinking times. Fig. 15 compares the mean mood for each subject between drinking and non-drinking times. The bar plot also indicates the mean value of drinking. In Fig. 15, it is clear to see how alcohol affects mood for subject 1001. Positive mood changes differently during drinking times when compared with non-drinking times.

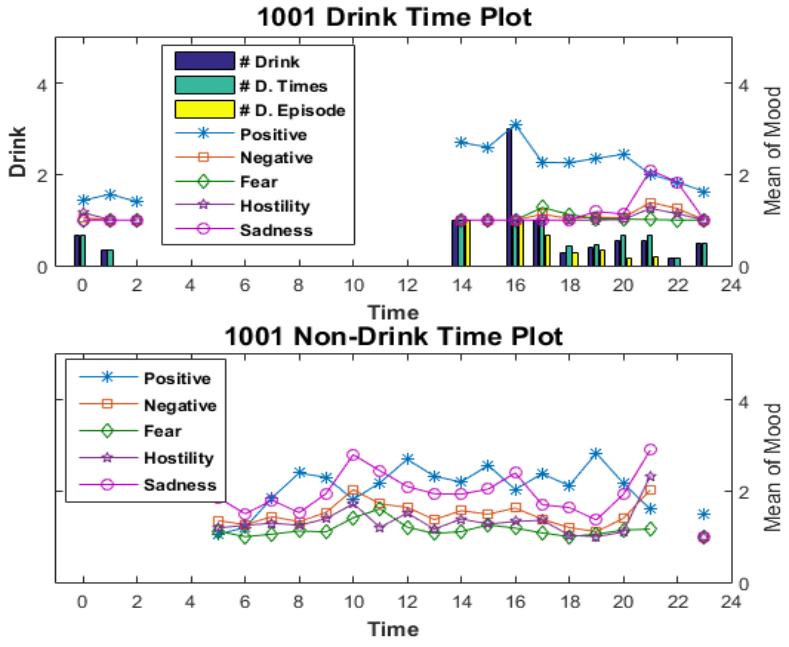

Figure 15. Graph of subject 1001's survey data. (time comparison)

Fig. 16 shows the box plot of mood for each subject's survey data during drinking versus non-drinking times. From the plots, some significant differences can be seen, for example sadness in the top graph, for each subject. In addition, different subjects have different mood levels, e.g. sadness for subject 1001, when they are drinking, as shown in Figure 16.

Table 10. The value in the left sub-column is drinking day's p-value for each subject.

P Value of Individual in Drink Day/Time										
ID	Positive		Negative		Fear		Hostility		Sadness	
1001	0.05	0.29	0.00	0.01	0.33	0.25	0.00	0.03	0.00	0.02
1003	0.88	0.82	0.04	0.05	0.00	0.01	0.24	0.27	0.51	0.38
1004	0.76	0.54	0.74	0.03	0.30	0.20	0.27	0.27	0.15	0.13
1005	0.01	0.87	0.62	0.84	0.09	0.24	0.20	0.72	0.95	0.85
1007	0.37	0.26	0.46	0.98	0.09	0.68	0.64	0.36	0.78	0.92
1008	0.06	0.00	0.19	0.04	0.89	0.98	0.10	0.02	0.17	0.03
1009	0.58	0.82	0.16	0.23	0.37	0.17	0.08	0.25	0.72	0.44
1010	0.00	Null	0.00	Null	0.02	Null	0.00	Null	0.00	Null
1013	0.15	0.30	0.06	0.08	0.12	0.19	0.08	0.17	0.27	0.30
1014	0.04	0.92	0.30	0.51	0.48	0.50	0.58	0.65	0.32	0.56
1017	0.94	0.79	0.84	0.44	0.38	0.74	0.99	0.30	0.49	0.95
1019	0.59	0.73	0.00	0.05	0.00	0.08	0.03	0.13	0.60	0.38
1020	0.17	0.06	0.91	0.64	0.28	0.08	0.47	0.27	0.71	0.77
1021	0.16	0.00	0.16	0.03	0.66	0.33	0.40	0.19	0.00	0.01
1022	0.82	0.14	0.07	0.62	0.30	0.40	0.50	0.97	0.04	0.65
1024	0.18	0.05	0.69	0.25	0.50	0.24	0.83	0.61	0.46	0.15

We also tested whether the levels of each emotion within a drinking day was different that for a non-drinking day, as well as for a drinking versus non-drinking time. In Table 10, the p value of the drinking effect based on an Unbalanced Nested ANOVA is presented. This approach treats drinking as the main effect, and data included each participant's emotion score values, using matched times between drinking and non-drinking. If there was no matched time available for a participant, a null value to the p-value was indicated. In Table 10, at most 40% in the sample have significant differences for negative affect during drinking versus non-drinking matched times. Considering drinking day results indicated that approximately around 25% in the sample shows a significant effect for drinking across all emotions.

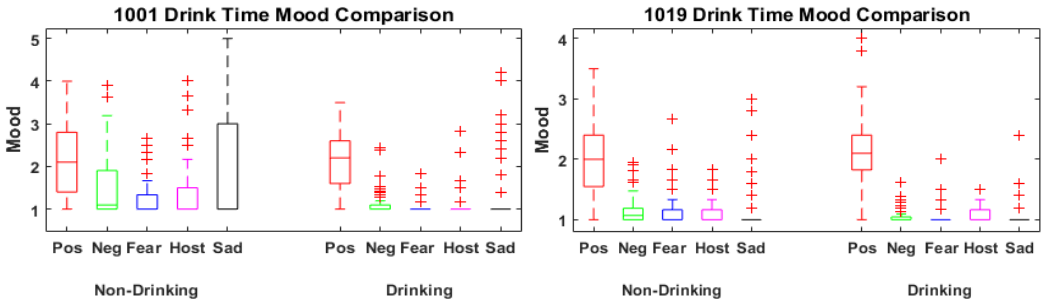

Figure 16. Box plots of two subjects' survey data (drinking time)

Next, we used the Shapiro-Wilk test to determine whether the mean and variance of emotion scores were significantly different between drinking and non-drinking across all subjects. Table 11 indicates that the variance of positive affect scores is significantly different for drinking time while the means of the other four affect scores are statistically different. Table 12 shows the percent increase or decrease of scores by drinking versus non-drinking status, e.g. for mean hostility scores. There is a decrease of 9.68% across all subjects when drinking alcohol. Results from these two tables suggest that drinking time versus day reveals more differences in level of emotions for this sample. In addition, the variance of positive affect significantly decreases (i.e., -27.46%) when people drink alcohol.

Table 11. Comparison of mood in drinking day/time

P Value of All Subjects in Drink Day/Time										
	Positive		Negative		Fear		Hostility		Sadness	
Mean	0.57	0.76	0.03	0.00	0.30	0.04	0.28	0.00	0.41	0.03
Variance	0.30	0.03	0.62	0.30	0.68	0.23	0.87	0.10	0.85	0.74

Table 12. Increasing ratio of mood in drinking day/time

Increasing Ratio(%) in Drink Day/Time										
	Positive		Negative		Fear		Hostility		Sadness	
Mean	2.17	3.46	-4.97	-7.86	-1.76	-4.26	-5.22	-9.68	-5.71	-8.57
Variance	0.07	-27.46	17.83	-12.97	15.43	-21.56	5.00	-24.83	7.32	-10.08

4.6.2 **Analyzing combined sensor and survey data of ADA**

In this section, results of analyzing sensor and survey data together are reported. All the drinking day's data are summed together, for each of the four physiological indices mentioned above, and then the mean value is calculated for each minute to generate the plots for the drinking times' data, as shown in Figure 17. The blue dots represent the mean value averaged by minute. Since the dot plots are too noisy to see any tendencies, the smoothing spline was used again. The black solid line in Figure 16 shows the smoothing line for these values. The smoothing plot for the four variables is created by combining the sensor data with the mood data for each subject. The mood data is found in the survey data. Some basic tendencies in the physical data for this subject are clearly seen. The plots show the respective physical indexes during periods of drinking times for this subject.

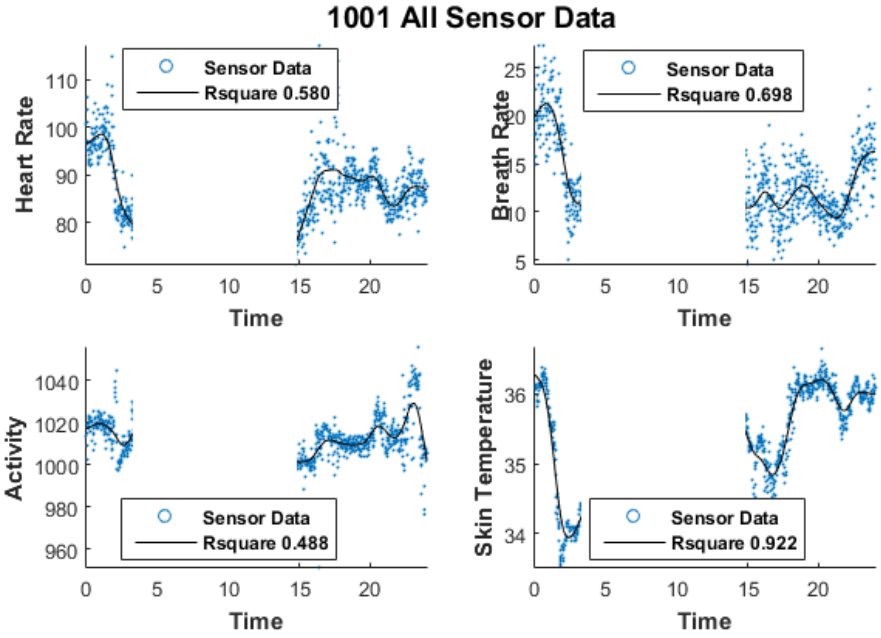

Figure 17. The smoothing graph for 4 signals of all data for 1001

As with the survey data alone, an unbalanced Nested ANOVA for individual case was conducted. We tested whether levels of four physiological variables differed during drinking versus non-drinking periods for each individual.

Table 13. Drinking Effect for Each Individual

P Value of Drinking Effect for Each Individual				
ID	Heart Rate	Breath Rate	Activity	Skin Temp
1001	0.201	0.182	0.352	0.066
1003	Null	Null	Null	Null
1004	0.000	0.001	0.224	0.432
1005	0.001	0.014	0.001	0.639
1007	0.741	0.263	0.163	0.186
1008	0.000	0.004	0.006	0.797
1010	Null	Null	Null	Null
1013	0.970	0.158	0.035	0.386
1019	0.450	0.162	0.578	0.011
1020	0.000	0.949	0.051	0.035

Table 13 shows p value of the drinking effect for four physiological factors mentioned above for each subject, within the same time block. If there was no matched time between drinking and non-drinking periods for an individual, the system assigned a Null value. Four out of eight participants showed significantly different heart rate levels during the drinking versus non-drinking periods, and at most 3 out of 8 showed significantly different for other indexes. After calculating mean value of these four indexes for each subject, the results shows heart rate increases 8.78% in drinking compared with non-drinking. These results suggest that heart rate are promising candidates for the prediction of alcohol use.

Table 14. Correlation matrix between heart rate, breathing rate, activity, and skin temp and different indexes of drinking alcohol for subject 1001 and 1005.

Correlation on Drinking Time for Subject 1001/1005						
	# Drinks		# D. Times		# D. Episode	
HR	-0.03	-0.10	-0.29	-0.23	-0.47	-0.38
BR	-0.18	0.55	-0.40	-0.23	-0.52	-0.10
Act	-0.40	0.41	-0.60	-0.39	-0.86	-0.26
Skin	-0.56	0.59	-0.35	0.13	-0.44	0.19

Pearson correlations were computed to test the associations between sensor and survey data. Table 14 shows the correlation matrix for two individuals. For subject 1001, activity (-0.86) and skin temperature (-0.44) has a strong negative correlation with the number of drinking episodes, the activity variable has a strong negative correlation on number of drink times (-0.60), and skin temperature has a strong correlation on drink quantity (-0.56). However, for subject 1005, the correlation patterns are different, suggesting individual differences in these associations. Table 15 shows the mean and variance of correlations for 8 subjects mentioned in Table 15. The mean correlational value for each index is relatively low but the variance is quite high, suggesting high variability across participants. Overall, this is consistent with our conclusion that there is a wide range both physiological reactions and physical movements when drinking alcohol. Analyses at the individual level seem warranted.

Table 15. correlation between the four factors and different indexes of drinking alcohol for 8 subjects

Correlation on Drinking Time for 8 Subjects						
	# Drinks		# D. Times		# D. Episode	
HR	0.143	0.439	-0.064	0.432	-0.090	0.474
BR	-0.148	0.274	-0.276	0.288	0.009	0.227
Act	0.150	0.349	-0.136	0.421	-0.033	0.372
Skin	-0.077	0.230	-0.172	0.169	-0.153	0.175

4.6.3 **Experimental Design for Deep ADA**

In our experiments, there are two types of experiments, one is within-subject case, the other one is cross-subject case. In the within-subject case, there are 26,366 unlabeled data blocks across 8 subjects, need to be trained using 1D CNN. In order to test the performance of reconstruction, all the labeled data are used as testing data in 1D CNN. After extracting features from human engineer and 1D CNN, there are 214 data blocks across 8 subjects with 80 % head of time as training data, the remaining 50 data blocks as testing data. The cross-subject cases are similar as within-subject cases, however, the main difference is how to prepare the training and testing of data. In this scenario, we use 5 subjects' data as training, the remaining 3 subjects' data considered as testing. In order to train 1D CNN model, there are 13,450 unlabeled data blocks across 5 subjects are used in the training and remaining labeled data across 8 subjects as testing. In training machine learning classifier, 212 data blocks with half positive and half negative across 5 subjects are treated as training, remaining 52 as testing across other 3 subjects. There are three types of signals that are used in our experiment, heart rate, skin temperature and activity. For each signal, hand feature engineer will extract 8 types of statistical features, and 1D CNN extract the same number of features with reconstruction.

After extracting features from hand feature engineer and 1D CNN models, we fed them into several popular machine learning models. The models we used in our experiment includes naïve bayes, decision tree, random forest, adaboost and support vector machine. Our machine learning pipeline implemented in Matlab and multiple parameters and kernels are tested in the experiments to get the best performance. The theory of each model has been discussed in the related work.

4.6.4 **Within-subject cases**

The experimental design has been discussed in the previous section. At first, we will go through the performance of 1D CNN reconstruction to see if our proposed model has capability to reconstruct the raw signal. As shown in the Fig. 18, the worst correlation between group truth signal and reconstructed signal on the test dataset are 0.7259, the best one is 0.9525. The MSE loss and correlation curve of train and test dataset have been demonstrated in the first two plots of Fig. 18, and all of them converged very well. The right two plots on the first row show that the majority of correlation in train and test dataset are around 95% and 85%, respectively. The mean of correlation of test data is 0.85. Due to the complexity of 1D bio-sensor physiological data, the reconstruction performance is promising to extract features which can be used to represent the raw input signal. As shown in the Table 16, we compare the performance of two unsupervised learning methods for feature extraction using statistical features and 1D CNN features with the same machine learning classifier model, and another two supervised learning models, ResNet50 and SVM. Based on the results, feature extraction from deep learning outperforms the hand extracted features by 19% accuracy improvement in test data compared with stats feature extraction methods. Compared with the other two supervised learning methods, both of them are over-fitting on training dataset and our proposed methods outperformed them by around 21% accuracy.

4.6.5 **Cross-subject cases**

The experiments in cross-subject cases are similar as within-subject cases. However, because of the variation of distribution of train and test in cross-subject cases is

greater than its in within-cases, it makes the classification task on cross-subject cases harder. As shown in Fig 19, the mean of correlation in cross-subject cases is 0.81, most of the correlation in train and test is 0.9 and 0.8, respectively, worse than its in within-cases. The reason why it has worse performance compared to within-subject cases is that some of users' data was not provided in the training phase, however, in the within-subject case, all the unlabeled data were used to extract feature. Even though the result is a little bit worse, the features extracted from bottleneck layer are still able to reconstruct the tendency of raw input signal. However, the results showing in the Table 17 using 1D CNN features extraction are still better than stats feature extraction by 19%. Same as within cases, it has the same overfitting problem on two supervised learning methods so that our proposed methods outperform the other two models by 24% accuracy improvement.

4.7 **Conclusion**

In this chapter, the design, implementation, and preliminary analysis results of ADA are presented. The system is reliable, fast, and easy to use. Our analysis results show that the variability of positive affect decreases significantly and the mean level of negative, fear, hostility, and sadness affect also decrease significantly when people are drinking alcohol. In addition, we found that heart rate appears to be promising predictors of alcohol use. At the same time, it is important to note that there appear to be important individual differences in physiological reactions and physical activity associated with drinking alcohol. Finally, the results for drinking time (versus drinking day) reveal more significant patterns of association between both mood and physiology and drinking.

On the other side, we extend ADA system with a novel deep learning-based feature extraction method on bio-sensor physiological data. Our proposed method mainly focuses on extracting features using convolution operator. This method has 3 advantages, first, it can solve out the problem of losing information when transforming input signals into other data structures. Second, it also considers time-correlation in input signal when we use autoencoder to extract feature and reduce dimension. Finally, with the significant amount of unlabeled and few labeled data cases in bio-sensor physiological data, our proposed model fully utilizes all the provided data in the experiment. Based on our experiments, multiple cases of studies have been tested, and our proposed method outperforms other state-of-the-art models with the same bio-sensor dataset. This method can be migrated into other low frequency sensor data domains.

In a focus group interview in [73], participants indicated they would prefer more interactive interventions, such as games or calming music. Once our system can predict alcohol and the mental state of the users more accurately, context-aware features may be added. This might include various interactive intervention methods in cases of predicted alcohol craving or mood dysregulation.

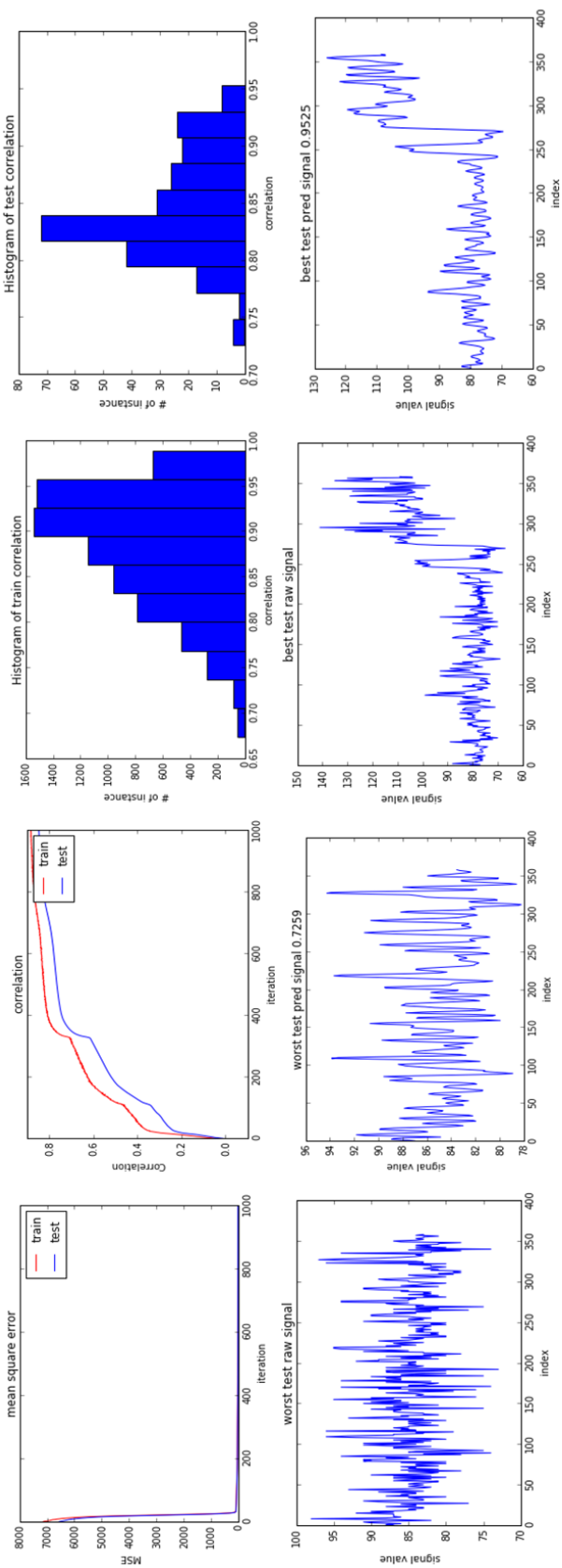

Figure 18. performance of signal reconstruction using 1D CNN in within subject

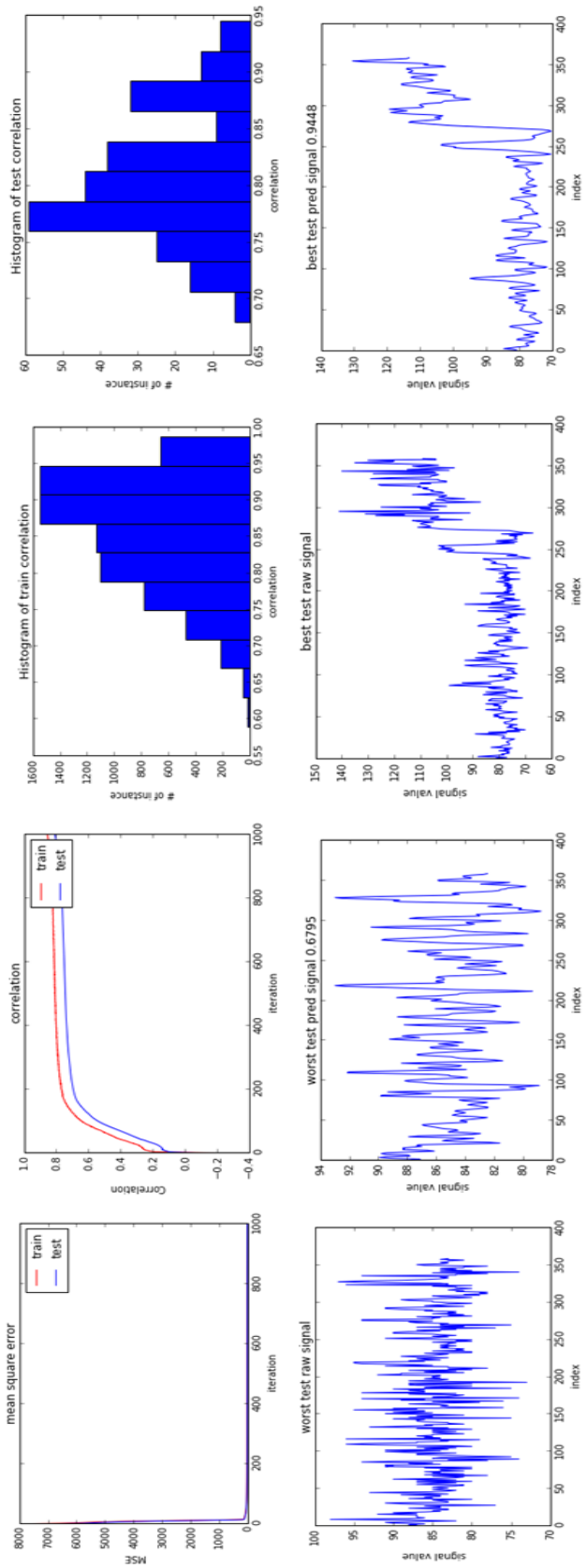

Figure 19. performance of signal reconstruction using 1D CNN in cross subject

Table 16. classification result of within subject case

	Models	Train	Test
Unsupervised Learning	Stats Features	0.72	0.55
	CNN Features	0.88	**0.74**
Supervised Learning	SVM	0.91	0.53
	ResNet50	0.94	0.52

Table 17. classification result of cross subject case

	Models	Train	Test
Unsupervised Learning	Stats Features	0.68	0.52
	CNN Features	0.87	**0.73**
Supervised Learning	SVM	0.92	0.49
	ResNet50	0.94	0.51

5. CONCLUSION

In this dissertation, we proposed data mining and two novel deep learning based algorithm to figure out problem in ambulatory assessment and aerial image detection.

In terms of aerial image object detection, for the problem of bird counting in aerial images, we compared the performance of different types of deep learning architectures for this problem. Based on the results, more discussion of character for each deep learning object detector has been made for this problem. Among object detection methods, RetinaNet performed the best on both cases. Between instance segmentation methods, U-Net achieved better performance than Mask R-CNN. These results are useful for identifying the strengths and weaknesses of existing methods and the development of future methods with improved performance.

In addition, after comparing the performance of the state-of-the-art models, novel deep learning algorithm, adaptive saliency biased loss (ASBL), has been proposed to deal with the problem of object detection in aerial images. The method use complexity information of input images to weigh the inputs differently in training. Without loss of generality, the ASBL approach was applied to RetinaNet to show its effectiveness. Using two large benchmark datasets, DOTA and NWPU VHR-10, experimental results show that ASBL-RetinaNet outperformed existing state-of-the-art deep learning methods, with at least 6.4 mAP improvement on DOTA, and 2.19 mAP on NWPU VHR-10. Furthermore, ASBL-RetinaNet improved over the original RetinaNet by 3.61 mAP on DOTA and 12.5 mAP on NWPU VHR-10.

In terms of ambulatory assessment analysis, the ADA algorithm for alcohol craving is reliable, fast, and easy to use. Our analysis results show that the variability of positive affect decreases significantly and the mean level of negative, fear, hostility, and sadness affect also decrease significantly when people are drinking alcohol. In addition, some other patterns in physiological data has been demonstrated in this dissertation. Based on all the analysis made in ADA, further analysis using machine learning has been used. In terms of the problem of feature extraction in physiological domain, we extended ADA using a novel deep learning based feature extraction method for raw 1D signal, called Deep ADA. Proposed method mainly focuses on extracting features using convolution operator. This method has 3 advantages, first, it can solve out the problem of losing information when transforming input signals into other data structures. Second, it also considers time-correlation in input signal when we use autoencoder to extract feature and reduce dimension. Finally, with the significant amount of unlabeled and few labeled data cases in bio-sensor physiological data, our proposed model fully utilize all the provided data in the experiment. Based on our experiments, multiple cases of studies have been tested, and our proposed method outperforms other state-of-the-art models with the same bio-sensor dataset.

6. BIBLIOGRAPHY

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in Advances in neural information processing systems, 2012, pp. 1097–1105.

[2] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” arXiv preprint arXiv:1409.1556, 2014.

[3] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.

[4] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2014, pp. 580–587.

[5] S. Ren, K. He, R. Girshick, and J. Sun, “Faster r-cnn: Towards real-time object detection with region proposal networks,” in Advances in neural information processing systems, 2015, pp. 91–99.

[6] K. He, G. Gkioxari, P. Doll´ar, and R. Girshick, “Mask r-cnn,” in Computer Vision (ICCV), 2017 IEEE International Conference on. IEEE, 2017, pp. 2980–2988.

[7] T.-Y. Lin, P. Doll´ar, R. B. Girshick, K. He, B. Hariharan, and S. J. Belongie, “Feature pyramid networks for object detection.” in CVPR, vol. 1, no. 2, 2017, p. 3.

[8] J. Redmon and A. Farhadi, “Yolo9000: better, faster, stronger,” arXiv preprint, 2017.

[9] J. Dai, Y. Li, K. He, and J. Sun, “R-fcn: Object detection via region-based fully convolutional networks,” in Advances in neural information processing systems, 2016, pp. 379–387.

[10] A. Shrivastava, A. Gupta, and R. Girshick, “Training region-based object detectors with online hard example mining,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 761– 769.

[11] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, “Ssd: Single shot multibox detector,” in European conference on computer vision. Springer, 2016, pp. 21–37.

[12] G. Chen, P. Sun, and Y. Shang, “Automatic fish classification system using deep learning,” in Tools with Artificial Intelligence (ICTAI), 2017 IEEE 29th International Conference on. IEEE, 2017, pp. 24–29.

[13] T. Tang, S. Zhou, Z. Deng, H. Zou, and L. Lei, “Vehicle detection in aerial images based on region convolutional neural networks and hard negative example mining,” Sensors, vol. 17, no. 2, p. 336, 2017.

[14] L. W. Sommer, T. Schuchert, and J. Beyerer, “Deep learning based multi-category object detection in aerial images,” in Automatic Target Recognition XXVII, vol. 10202. International Society for Optics and Photonics, 2017, p. 1020209. 11

[15] X. Yang, H. Sun, K. Fu, J. Yang, X. Sun, M. Yan, and Z. Guo, “Automatic ship detection in remote sensing images from google earth of complex scenes based on multiscale rotation dense feature pyramid networks,” Remote Sensing, vol. 10, no. 1, p. 132, 2018.

[16] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “Imagenet: A large-scale hierarchical image database,” in Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on. IEEE, 2009, pp. 248–255.

[17] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Doll´ar, and C. L. Zitnick, “Microsoft coco: Common objects in context,” in European conference on computer vision. Springer, 2014, pp. 740–755.

[18] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Doll´ar, “Focal loss for dense object detection,” in Proceedings of the IEEE international conference on computer vision, 2017, pp. 2980–2988.

[19] P. Sun, N. M. Wergeles, C. Zhang, L. M. Guerdan, T. Trull, and Y. Shang, “Ada-automatic detection of alcohol usage for mobile ambulatory assessment,” in Smart Computing (SMARTCOMP), 2016 IEEE International Conference on. IEEE, 2016, pp. 1–5.

[20] J. P. Bernstein, B. J. Mendez, P. Sun, Y. Liu, and Y. Shang, “Using deep learning for alcohol consumption recognition,” in Consumer Communications & Networking Conference (CCNC), 2017 14th IEEE Annual. IEEE, 2017, pp. 1020–1021.

[21] G.-S. Xia, X. Bai, J. Ding, Z. Zhu, S. Belongie, J. Luo, M. Datcu, M. Pelillo, and L. Zhang, “Dota: A large-scale dataset for object detection in aerial images,” in Proc. CVPR, 2018.

[22] S. Li, Z. Zhang, B. Li, and C. Li, “Multiscale rotated bounding box-based deep learning method for detecting ship targets in remote sensing images,” Sensors, vol. 18, no. 8, p. 2702, 2018.

[23] S. M. Azimi, E. Vig, R. Bahmanyar, M. K¨orner, and P. Reinartz, “Towards multi-class object detection in unconstrained remote sensing imagery,” arXiv preprint arXiv:1807.02700, 2018.

[24] D. Zhang, D. Meng, and J. Han, “Co-saliency detection via a self-paced multiple-instance learning framework,” IEEE transactions on pattern analysis and machine intelligence, vol. 39, no. 5, pp. 865–878, 2016.

[25] D. Zhang, J. Han, C. Li, J. Wang, and X. Li, “Detection of co-salient objects by looking deep and wide,” International Journal of Computer Vision, vol. 120, no. 2, pp. 215–232, 2016.

[26] A. Recasens, P. Kellnhofer, S. Stent, W. Matusik, and A. Torralba, “Learning to zoom: a saliency-based sampling layer for neural networks,” in Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 51–66.

[27] K. Li, G. Cheng, S. Bu, and X. You, “Rotation-insensitive and context-augmented object detection in remote sensing images,” IEEE Transactions on Geoscience and Remote Sensing, vol. 56, no. 4, pp. 2337–2348, 2018.

[28] G. Cheng and J. Han, “A survey on object detection in optical remote sensing images,” ISPRS Journal of Photogrammetry and Remote Sensing, vol. 117, pp. 11–28, 2016.

[29] G. Cheng, P. Zhou, and J. Han, “Learning rotation-invariant convolutional neural networks for object detection in vhr optical remote sensing images,” IEEE Transactions on Geoscience and Remote Sensing, vol. 54, no. 12, pp. 7405–7415, 2016.

[30] G. Cheng, J. Han, P. Zhou, and D. Xu, “Learning rotation-invariant and fisher discriminative convolutional neural networks for object detection,” IEEE Transactions on Image Processing, vol. 28, no. 1, pp. 265–278, 2018.

[31] J. Han, D. Zhang, G. Cheng, L. Guo, and J. Ren, “Object detection in optical remote sensing images based on weakly supervised learning and high-level feature learning,” IEEE Transactions on Geoscience and Remote Sensing, vol. 53, no. 6, pp. 3325–3337, 2014.

[32] F. Provost, “Machine learning from imbalanced data sets 101.”

[33] Q. Dong, S. Gong, and X. Zhu, “Class rectification hard mining for imbalanced deep learning,” 2017.

[34] R. Girshick, “Fast r-cnn,” in Proceedings of the IEEE international conference on computer vision, 2015, pp. 1440–1448.

[35] C.-Y. Fu, W. Liu, A. Ranga, A. Tyagi, and A. C. Berg, “Dssd: Deconvolutional single shot detector,” arXiv preprint arXiv:1701.06659, 2017.

[36] C. L. Zitnick and P. Doll´ar, “Edge boxes: Locating object proposals from edges,” in European conference on computer vision. Springer, 2014, pp. 391–405.

[37] S. Yang, P. Luo, C.-C. Loy, and X. Tang, “Wider face: A face detection benchmark,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 5525–5533.

[38] R. Tudor Ionescu, B. Alexe, M. Leordeanu, M. Popescu, D. P. Papadopoulos, and V. Ferrari, “How hard can it be? estimating the difficulty of visual search in an image,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 2157–2166.

[39] Q. Tao, H. Yang, and J. Cai, “Exploiting web images for weakly supervised object detection,” IEEE Transactions on Multimedia, vol. 21, no. 5, pp. 1135–1146, 2018.

[40] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, “The pascal visual object classes (voc) challenge,” International journal of computer vision, vol. 88, no. 2, pp. 303–338, 2010.

[41] G. Cheng, J. Han, P. Zhou, and L. Guo, “Multi-class geospatial object detection and geographic image classification based on collection of part detectors,” ISPRS Journal of Photogrammetry and Remote Sensing, vol. 98, pp. 119–132, 2014.

[42] J. R. Uijlings, K. E. Van De Sande, T. Gevers, and A. W. Smeulders, “Selective search for object recognition,” In IJCV, 2013.

[43] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” In CVPR, 2014.

[44] R. Girshick, “Fast r-cnn,” In ICCV, 2015.

[45] S. Ren, K. He, R. Girshick, and J. Sun, “Faster r-cnn: Towards realtime object detection with region proposal networks,” In NIPS, 2015.

[46] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” In CVPR, 2016.

[47] J. Redmon and A. Farhadi. “YOLO9000: Better, faster, stronger,” In CVPR, 2017.

[48] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, “SSD: Single shot multibox detector,” In ECCV, 2016.

[49] Hu, Peiyun and Ramanan, Deva, “Finding Tiny Faces,” In CVPR, 2017

[50] Najibi, Mahyar and Samangouei, Pouya and Chellappa, Rama and Davis, Larry, “SSH: Single Stage Headless Face Detector,” In ICCV 2017.

[51] C.-Y. Fu, W. Liu, A. Ranga, A. Tyagi, and A. C. Berg. “DSSD: Deconvolutional single shot detector,” arXiv:1701.06659, 2016.

[52] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama et al., “Speed/accuracy trade-offs for modern convolutional object detectors,” In CVPR, 2017.

[53] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie. “Feature pyramid networks for object detection,” In CVPR, 2017.

[54] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar. “Focal loss for dense object detection,” arXiv preprint arXiv:1708.02002, 2017.

[55] .K. He, G. Gkioxari, P. Dollar, and R. Girshick, “Mask r-cnn,” arXiv:1703.06870, 2017.

[56] O. Ronneberger, P. Fischer, and T. Brox. “U-net: Convolutional networks for biomedical image segmentation,” In MICCAI, pages 234–241. Springer, 2015.

[57] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” In CVPR, 2015.

[58] C. Zitnick and P. Dollar, “ Edge boxes: Locating object proposals from edges,” In ECCV, 2014.

[59] S. Yang, P. Luo, C.-C. Loy, and X. Tang. “Wider face: A face detection benchmark,” In ICCV, June 2016.

[60] V. Jain and E. Learned-Miller. “Fddb: A benchmark for face detection in unconstrained settings,” Technical Report UMCS-2010-009, University of Massachusetts, Amherst, 2010.

[61] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, T. Darrell, “Caffe: Convolutional architecture for fast feature embedding,” Proceedings of the 22nd ACM international conference on Multimedia, 675-678, 2014.

[62] K. Simonyan and A. Zisserman. “Very deep convolutional networks for large-scale image recognition,” In ICLR, 2015.

[63] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollar, and C. L. Zitnick. “Microsoft COCO: Common objects in context,” In ECCV. 2014.

[64] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. “The Pascal Visual Object Classes (VOC) Challenge,” IJCV, pages 303–338, 2010.

[65] Isola, P., Zhu, J.Y., Zhou, T. and Efros, A.A., 2017. “Image-to-image translation with conditional adversarial networks,” arXiv preprint.

[66] Zhu, J.Y., Park, T., Isola, P. and Efros, A.A., 2017. “Unpaired image-to-image translation using cycle-consistent adversarial networks,” arXiv preprint arXiv:1703.10593.

[67] Society for Ambulatory Assessment, 2012, www.ambulatory-assessment.org.

[68] G. D. Abowd, A. K. Dey, P. J. Brown, N. Davies, M. Smith, and P. Steggles, “Towards a better understanding of context and context-awareness,” Proc. 1st Int’l Symposium on Handheld and Ubiquitous Computing, HUC '99, pages 30-307, 1999.

[69] T. Starner, Wearable Computing and Contextual Awareness, Ph.D. thesis, MIT Media Lab., Apr. 30, 1999.

[70] T. Choudhury et al., “The Mobile Sensing Platform: An Embedded System for Activity Recognition,” IEEE Pervasive Comp., vol. 7, no. 2, 2008, pp. 32–41.

[71] H. Lu et al., “Sound-Sense: Scalable Sound Sensing for People-Centric Applications on Mobile Phones,” Proc. 7th ACM MobiSys, pp. 165–78, 2009.

[72] K. Plarre et al.,”Continuous inference of psychological stress from sensory measurements collected in the natural environment,” Proc. 10th Int’l Conference on Information Processing in Sensor Networks (IPSN), pp.97-108, 2011.

[73] E.W. Boyer, R. Fletcher, R.J. Fay, D. Smelson, D. Ziedonis, and R.W. Picard, “Preliminary efforts directed toward the detection of craving of illicit substances: the iHeal project,” J Med Toxicol. 8(1):5-9, March 2012.

[74] E. Miluzzo et al., “Sensing meets Mobile Social Networks: The Design, Implementation, and Evaluation of the CenceMe Application,” Proc. 6th ACM SenSys, pp. 337–50, 2008.

[75] M. Mun et al., “Peir, the Personal Environmental Impact Report, as a Platform for Participatory Sensing Systems Research,” Proc. 7th ACM MobiSys, pp. 55–68, 2009.

[76] S. Consolvo et al., “Activity Sensing in the Wild: A Field Trial of Ubifit Garden,” Proc. 26th Annual ACM SIGCHI Conf. Human Factors Comp. Sys., pp. 1797–1806, 2008.

[77] A. Thiagarajan et al., “VTrack: Accurate, Energy-Aware Traffic Delay Estimation Using Mobile Phones,” Proc. 7th ACM SenSys, Nov. 2009.

[78] L. Constantine and H. Haij, "A survey of ground-truth in emotion data annotation," IEEE Int’l Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops), pp.697-702, 2012.

[79] G. Miller, “The Smartphone Psychology Manifesto,” Perspectives on Psychological Science, vol. 7 no. 3, pages 221-237, 2012.

[80] S.M. Hossain et al, “Identifying drug intake events from acute physiological response in the presence of free-living physical activity”, IPSN ’14 Proceedings of the 13th int’l symposium on Information processing in sensor networks, pp71-82,2014

[81] R. Shi, et al., "mAAS – A Mobile Ambulatory Assessment System for Alcohol Craving Studies" IEEE Computer Software and Applications Conference (COMPSAC), 2015 IEEE 39th Annual , pp.282-287, 2015

[82] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in Advances in neural information processing systems, 2012, pp. 1097–1105.

[83] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” arXiv preprint arXiv:1409.1556, 2014.

[84] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.

[85] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2014, pp. 580–587.

[86] S. Ren, K. He, R. Girshick, and J. Sun, “Faster r-cnn: Towards real-time object detection with region proposal networks,” in Advances in neural information processing systems, 2015, pp. 91–99.

[87] K. He, G. Gkioxari, P. Doll´ar, and R. Girshick, “Mask r-cnn,” in Computer Vision (ICCV), 2017 IEEE International Conference on. IEEE, 2017, pp. 2980–2988.

[88] T.-Y. Lin, P. Doll´ar, R. B. Girshick, K. He, B. Hariharan, and S. J. Belongie, "Feature pyramid networks for object detection." in CVPR, vol. 1, no. 2, 2017, p. 3.

[89] J. Redmon and A. Farhadi, "Yolo9000: better, faster, stronger," arXiv preprint, 2017.

[90] J. Dai, Y. Li, K. He, and J. Sun, "R-fcn: Object detection via region based fully convolutional networks," in Advances in neural information processing systems, 2016, pp. 379–387.

[91] A. Shrivastava, A. Gupta, and R. Girshick, "Training region-based object detectors with online hard example mining," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 761–769.

[92] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in European conference on computer vision. Springer, 2016, pp. 21–37.

[93] R. C. King, E. Villeneuve, R. J. White, R. S. Sherratt, W. Holderbaum, and W. S. Harwin, "Application of data fusion techniques and technologies for wearable health monitoring," Medical engineering & physics, vol. 42, pp. 1–12, 2017.

[94] A. Godfrey, "Wearables for independent living in older adults: Gait and falls," Maturitas, vol. 100, pp. 16–26, 2017.

[95] S. M. Hossain, A. A. Ali, M. M. Rahman, E. Ertin, D. Epstein, A. Kennedy, K. Preston, A. Umbricht, Y. Chen, and S. Kumar, "Identifying drug (cocaine) intake events from acute physiological response in the presence of free-living physical activity," in Proceedings of the 13th international symposium on Information processing in sensor networks. IEEE Press, 2014, pp. 71–82.

[96] P. Sun, N. M. Wergeles, C. Zhang, L. M. Guerdan, T. Trull, and Y. Shang, "Ada-automatic detection of alcohol usage for mobile ambulatory assessment," in Smart Computing (SMARTCOMP), 2016 IEEE International Conference on. IEEE, 2016, pp. 1–5.

[97] L. Wei, Y. Lin, J. Wang, and Y. Ma, "Time-frequency convolutional neural network for automatic sleep stage classification based on single channel eeg," in 2017 IEEE 29th International Conference on Tools with Artificial Intelligence (ICTAI). IEEE, 2017, pp. 88–95.

[98] Y. Zhang, Y. Chen, L. Hu, X. Jiang, and J. Shen, "An effective deep learning approach for unobtrusive sleep stage detection using microphone sensor," in 2017 IEEE 29th International Conference on Tools with Artificial Intelligence (ICTAI). IEEE, 2017, pp. 37–44.

[99] R. F. Borkenstein and H. Smith, "The breathalyzer and its applications," Medicine, Science and the Law, vol. 2, no. 1, pp. 13–22, 1961.

[100] B. Nassi, L. Rokach, and Y. Elovici, "Virtual breathalyzer," arXiv preprint arXiv:1612.05083, 2016.

[101] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Doll´ar, and C. L. Zitnick, "Microsoft coco: Common objects in context," in European conference on computer vision. Springer, 2014, pp. 740–755.

[102] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on. IEEE, 2009, pp. 248–255.

[103] Y. Zheng, Q. Liu, E. Chen, Y. Ge, and J. L. Zhao, “Time series classification using multi-channels deep convolutional neural networks,” in International Conference on Web-Age Information Management. Springer, 2014, pp. 298–310.

[104] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, “Extracting and composing robust features with denoising autoencoders,” in Proceedings of the 25th international conference on Machine learning. ACM, 2008, pp. 1096–1103.

[105] J. Masci, U. Meier, D. Cires¸an, and J. Schmidhuber, “Stacked convolutional auto-encoders for hierarchical feature extraction,” in International Conference on Artificial Neural Networks. Springer, 2011, pp. 52–59.

[106] J. Snell, K. Swersky, and R. Zemel, “Prototypical networks for few-shot learning,” in Advances in Neural Information Processing Systems, 2017, pp. 4077–4087.

[107] X. Feng, Y. Zhang, and J. Glass, “Speech feature denoising and dereverberation via deep autoencoders for noisy reverberant speech recognition,” in 2014 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, 2014, pp. 1759–1763.

[108] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” in International Conference on Medical image computing and computer-assisted intervention. Springer, 2015, pp. 234–241.

[109] V. Badrinarayanan, A. Kendall, and R. Cipolla, “Segnet: A deep convolutional encoder-decoder architecture for image segmentation,” IEEE transactions on pattern analysis and machine intelligence, vol. 39, no. 12, pp. 2481–2495, 2017.

7. VITA

Peng Sun was born in Tianjin, China. He is currently a PhD candidate in EECS Department at the University of Missouri, Columbia, MO 65211, USA. He got MA in Statistics at the University of Missouri, Columbia, MO 65211, USA in 2014 and BS in Applied Mathematics at Ningbo University, Ningbo, China, in 2011. In his PhD degree, he published 8 papers, and 3 more peer-review papers are in progress. His research interests include machine learning, statistically learning, deep learning, computer vision and object detection. He was a summer intern with The Climate Corporation (Bayer Crop Science) as AI & Machine learning position in San Francisco, CA, in 2019. He will be joining The Climate Corporation as AI Scientist.