FINE-GRAINED VIDEO CLASSIFICATION FOR RARE EVENTS

by

Junjie Shan

A dissertation submitted to the faculty of
The University of North Carolina at Charlotte
in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in
Computer Science

Charlotte

2018

Approved by:

____

Dr. Min Shin

____

Dr. Richard Souvenir

____

Dr. Jianping Fan

____

Dr. Richard Lambert

©2018
Junjie Shan
ALL RIGHTS RESERVED

Abstract

JUNJIE SHAN. Fine-grained Video Classification for Rare Events. (Under the direction of DR. RICHARD M. SOUVENIR)

Video analysis plays an important role in the field of computer vision and finds its application in many areas. Fine-grained event classification is one of the most challenging problems in video analysis due to subtle difference between classes and limited training examples, such as echocardiogram function prediction and social insect behavior classification. The difference between patterns of interest in these tasks is hard to perceive so we must rely on domain experts with professional skills to annotate the unlabeled videos. As a result, the data set of annotated videos is usually in small quantity or severely unbalanced. The performance of various traditional shallow learning methods is bounded by handcrafted feature extraction and data scarcity. Recently, the methods based on deep learning, such as convolutional neural network (CNN), have made substantial advancements in various vision tasks. They learn feature representation in a pure data-driven manner. In this dissertation, we propose a set of methods to address three fine-grained video classification problems for rare events. We first present an approach to classify fine-grained echocardiogram videos with subtle difference and limited training data using 3D CNN. Then, we investigate an autoencoder with 3D CNN structure and additional one-class support vector machine (OCSVM) layer to detect impaired heart videos using unbalanced echocardiogram dataset. Finally, we propose a pipeline to localize fine-grained pairwise ant behaviors, by generating behavior proposals from convolutional feature maps computed by 3D CNN.

ACKNOWLEDGEMENTS

First of all, I would like to thank Dr. Richard Souvenir for the advising of my research. This work would not have been accomplished without the guidance of him. I have learned a lot from him. I took his Machine Learning and Computer Vision courses, which are among the most knowledgeable and challenging classes in the department. He is a great mentor. I've been trained to present slides, make posters, write and review papers. I believe the training I've received from him would be beneficial to the career development in the rest of my life.

And I am grateful to Scott Spurlock and Hui Wu in our research group. They were also advised by Dr. Souvenir, and graduated three years ago. I have worked with them in several different projects for one year. They are good collaborators, good listeners and good researchers. I enjoyed the collaborations with them.

I want to give thanks to Dr. Min Shin for creating and maintaining such an active and comfortable environment in the lab. Our lab, Video and Image Analysis Lab (VIA), is one the best labs in the department. Thank you for organizing the fabulous outdoor activities every semester. I'd like to thank all the other people in VIA lab too. It is my pleasure and luck to work in VIA lab for four years.

Finally, I also would like to thank my wife, Junqing Ma, who always supports and helps me over the past years. It's not an easy process for both myself and my entire family. Thank you for encouraging me and patiently waiting for the completion of my research work.

Contents

List of Figures

List of Tables

CHAPTER 1: INTRODUCTION

Video analysis plays an important role in the field of computer vision, it finds applications in many areas. One of the most fundamental video analyses is the behavior classification or event classification based on the content, it commonly serves as the basis of many other higher-level processing. The difficulty of solving a particular behavior classification task depends primarily on the visual difference of patterns we want to separate. For instance, two video classification tasks are shown in the following:

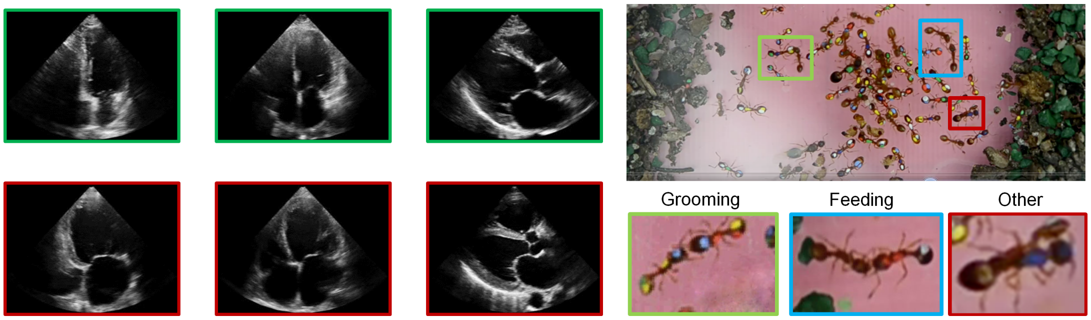
A) Given a collection of sports games videos containing basketball game and soccer game, classify each video based on game category, i.e, basketball or soccer.

B) Given a collection of basketball dribble videos, classify each video into legal dribble or illegal dribble.

It is generally considered that Task B is more difficult than Task A. The reason is that Task B classifies two behaviors with high similarities that belong to the same category of sport. The difference between legal and illegal basketball dribbles only lies in the insignificant wrist and palm movement. It is challenging to design a model to captures those minor difference in details. However, the difficulty of recognizing basketball game and soccer game is relatively lower since they are perceivably distinct in term of limbs movement. The type of video classification in Task B, is often referred to as fine-grained classification because it operates in subcategory level.

1.1 Problems

In this dissertation, we attempt to solve fine-grained event classification of echocardiogram and social ant videos. They are the real-world problems not as commonly seen as sport videos. Figure 1.1 (a) depicts three echocardiograms from healthy human heart (top row) and three echocardiograms with severe heart disease (bottom row). As we can see, the difference between healthy echocardiogram and impaired echocardiogram is very subtle. In Figure 1.1 (b) we show three types of pairwise ant behaviors, which are used by the biologists to study the societies of insects. It is also difficult to recognize different pairwise ant behaviors since they look visually almost the same.



(a) (b)

Figure 1.1: Two fine-grained video classification problems we solve in this dissertation. **(a)**, classification of healthy echocardiograms and impaired echocardiograms: three echocardiograms from healthy human heart are in the top row, three more echocardiograms with severe heart disease are shown in the bottom row; **(b)**, detection of different pairwise ant behaviors: three types of ant behaviors are shown, which are "Grooming" (in yellow-green box), "Feeding" (in blue box) and "Other" behaviors (in red box). They are challenging to solve due to subtle appearance difference.

In the aforementioned fine-grained video classification problems, subtle difference is not the only issue that makes the problem challenging. The behaviors we attempt to separate in both problems are so similar that most people who do not have professional knowledge are unable to perceive the difference. The echocardiogram functionality classification relies on the cardiologists to evaluate the healthiness of patient and provide function label of the patient's heart condition; pairwise ant behavior classification depends on the annotations of the biologists, who watch and review the ant videos frame-by-frame. In both problems, domain knowledge or learned skills are needed to acquire the training examples. Still, the annotation of video dataset is often time-consuming due to high inter-class variations between the patterns to separate. For this reason, the volumes of dataset in both problems are not able to increase through crowd-sourcing. It's the primary factor why the annotated dataset of this type is usually in small quantity or unbalanced.

As a result, it is more challenging to design algorithms for these video analysis tasks due to both high-similarity and small-size of the data.

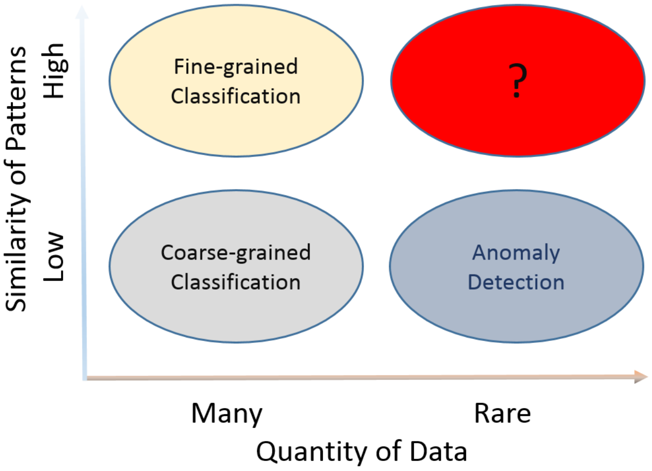


Figure 1.2: Taxonomy of video classification problems. The difficulty increases as different patterns become fine-grained and data size gets smaller. The problems we attempt to solve are the intersection of both fine-grained and rare video classification.

In summary, there are two common issues in echocardiogram classification and ant pairwise ant behavior classification, which are subtle difference between different patterns and limited training examples. Figure 1.2 shows the taxonomy of video classification problems in two dimensions, the data size and similarity of different patterns. Compared against other video classification tasks, our problems are more challenging to solve due to both high-similarity and small-size of the data.

1.2 Related Work

In this section, we first review recent literature that is related to fine-grained and rare video classification. In the following subsection we briefly introduce the most important method in this dissertation, 3D convolutional neural network (CNN).

1.2.1 Fine-grained rare video classification

In the literature of content-based video classification, both *event classification* and *behavior classification* are used as frequently as each other. Although their definitions is not well defined and the difference between them is quite debatable, we review related work that use either term. We also use them interchangeably throughout the context of this dissertation.

The fine-grained and rare video event classification work that are most related to our research is the method in [22]. The author employs weakly supervised topic model based on Latent Dirichlet Allocation (LDA) [5] to detect subtle and rare events from traffic surveillance videos. Other than that, no previous work appears to explicitly address the fine-grained and rare issues simultaneously. For this reason, we will review existing methods to analyze *fine-grained event classification* and *rare event classification* separately as follows.

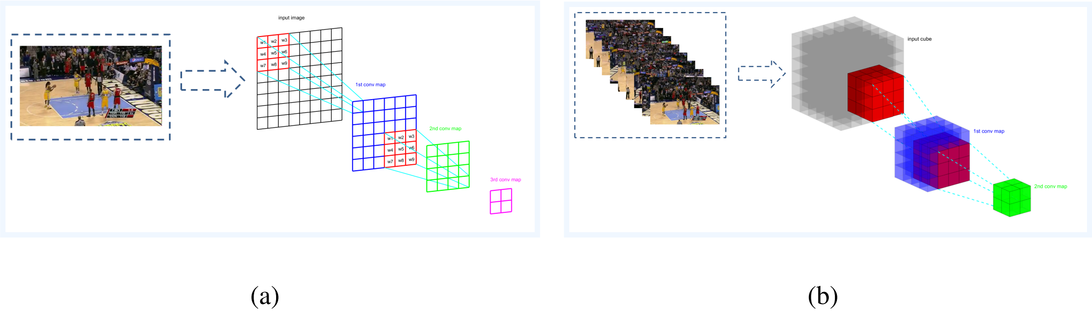
Concerning *fine-grain event classification*, Eulerian Video Magnification in [52] uses spatial and temporal frequency decomposition to select and amplify small movements in specific frequency range from videos. The method in [1] adopts similar techniques to detect human pulse from videos. The author use principal component analysis (PCA) to select the component that is closest to electrocardiogram (ECG) signal. Both work focus on amplifying one single type of imperceptible vibration of object in the videos. Our goal is different from them since we are more interested in classifying two or more categories of events in videos. Rohrbach et al. [39] collected a video database for fine-grained cooking video detection and extracted two features based on articulated human pose and holistic video to recognize fine-grained cooking behaviors. There has been several recent work in specific areas or application scenarios, such as fine-grained bird classification [40], fine-grained pedestrians classification [20]. Unlike our problems, which classify video based on behaviors presented in the video, these works classify video based on the object or people. Another approach [44] proposed an method to detect fine-grained shopping behaviors using a combination of two-stream convolutional neural network (CNN) and bi-directional long short-term memory (LSTM).

To address the problem of *rare event classification*, a variety of methods have been proposed. There are two broad categories of approaches in term of the how unbalanced the different classes are distributed. First, data in all classes are in small amount but equally or approximately equally distributed, transfer learning [36] is usually applied by taking advantage of well trained model for similar tasks. Secondly, data in majority class are sufficient but there is few data in minor class, i.e. training data are unbalanced. In this case, methods such as anomaly detection [7], resampling [8] and hard-negatives mining [15, 33] are often used. Among various anomaly detection algorithms, one-class classification [26], especially one-class support vector machine (OCSVM) [41], have been utilized to solve real-world anomaly detection problems. Another form of one-class classification, support vector data description (SVDD) [46] is also a method for anomaly detection but less widely used.

1.2.2 3D CNN

The convolutional neural network (CNN) [10] was proposed by Yann Lecun. CNN receives enormous attention in computer vision community and obtains the state-of-the-art performance in many image analysis tasks. For instance, Krizhevsky et al. [29] employs CNN to obtain the lowest top-5 error rate on a 1000-category object classification benchmark, outputperforming all the other methods by a large margin. Recently, the development of CNN has greatly advanced the progress in many image-related tasks, including image recognition [45, 43], objection detection [17] and semantic segmentation [32].

While CNN achieves impressive result in image-related tasks, it is still an active research area to utilize CNN or explore new type of CNN that works also on video-related tasks. The most straightforward approach to take advantage of CNN is to independently extract features from each frame in videos using pretrained CNN. Karpathy et al. [25] proposes to concatenate the convolutional features of multiple video frames to fuse the final spatial-temporal feature descriptor. However, this method is incapable of preserving the rich motion information across multiple frames. Two-Stream CNN [42] and Action Tubes [18] uses two separate CNNs to process original RGB image and optical flow. The feature maps of two CNNs are merged for jointly training.



(a)

(b)

Figure 1.3: (a) CNN for image analysis and (b) 3D-CNN for video analysis

The concept of 3D Convolutional Neural Network (3D-CNN) is a natural extension of CNN for image task. Shuiwang et al. [23] develops 3D CNN for human action recognition, and Tran et al. [47] applies it on large scale sport dataset. In the case of 3D-CNN, both the input data and convolution kernel are three dimensional. A video can be viewed as 3D tensor, since it represents a sequence of images and each pixel value can be uniquely indexed by a triplet (width, height, time). The difference between CNN and 3D-CNN is depicted in Figure 1.3. Comparing to the approach to fuse features of multiple still images using CNN, one single 3D-CNN better captures spatial-temporal features in video.

1.3 Summary

In this chapter, we first present the problems that we attempt to solve in this dissertation and why they are challenging. In addition to problems, we also review the related works and fundamental method 3D-CNN.

In the following chapters, we investigate three different methods based on 3D CNN for fine-grained and rare video analysis. The problems that we attempt to address are: 1) fine-grained echocardiogram viewpoint classification; 2) fine-grained echocardiogram function classification; 3) fine-grained pairwise ant behavior localization and classification.

CHAPTER 2: Fine-grained Echocardiogram Viewpoint Classification

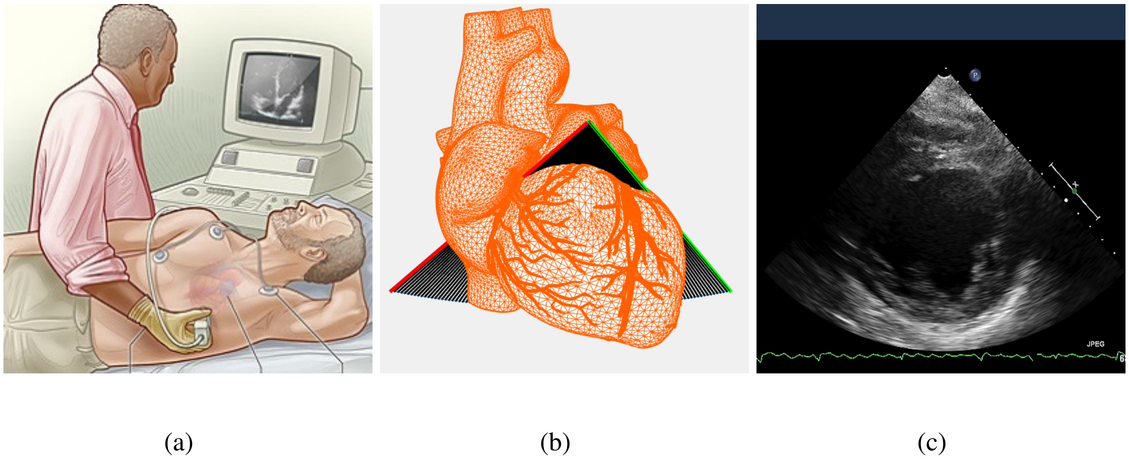
In this chapter, we present a method based on 3D-CNN to classify visually similar yet different echocardiogram videos. The echocardiogram is a medical ultrasound technology used for cardiac diagnostic. It is non-invasive, and relatively inexpensive. During the acquisition of echocardiogram, the positioning of the probe has a direct impact on the quality of the acquired images; shifts of as little as a few millimeters can render the images unusable. We present a method which interpolates the predictions of a deep convolutional neural network classifier to classify the viewpoint of the imaging probe directly from the visual data. For echocardiogram view classification, our method outperforms recent approaches on real-world data. Additionally, we present an application prototype which leverages the probe pose estimates to provide guidance to ultrasound technicians and can be used as a teaching tool.

2.1 Problem Statement

Medical ultrasound is a reliable, ubiquitous tool in clinical settings. Among many uses, handheld ultrasound provides imagery that allows physicians to quickly assess the cardiac health of critically-ill patients. For the most common modality, 2D echocardiography, the placement of the transducer on the patient determines what structures are visible in the resulting images. Certain locations correspond to common canonical viewpoints. Figure 2.1 shows how a typical transthoracic echocardiography is acquired and the sample echocardiogram.

Obtaining the most diagnostically-relevant images is a learned skill. Undertrained technicians often acquire images of poor quality or those with important structures not visible. The difference between a usable and unusable image can be due to positioning the probe in the wrong intercostal space or a tilt of the head or the tail as small as a millimeter from the ideal positioning. In the cases where ultrasound cannot be used diagnostically, the next step is often a more expensive and invasive approach.

we present an automated method to estimate the transducer viewpoint from an echocardiogram and an application that provides guidance to the technician on moving the transducer to obtain the desired view.


(a)

(b)

(c)

Figure 2.1: For 2D transthoracic echocardiography (a), the imaging plane is based on the positioning of the probe (b), and results in an image (c), which visualizes a "slice" of the heart. Our method estimates the transducer viewpoint from an echocardiogram for both standard and non-standard viewpoints and can be used to improve transducer positioning.

2.2 Related Work

Our method and intended application are related to echocardiogram view classification and self-calibration of sensors from images.

2.2.1 View classification

Previous methods have considered the problem of echocardiogram view classification. The method of Ebadollahi et al. [13] uses a generic cardiac chamber template for detection and models the properties of detected chambers using Markov Random Fields and multi-class SVM for classification. The method of Zhou et al. [59] uses boosted weak classifiers of Haar-like local rectangle features to generalize the detection of specific heart structures for view classification. The method of Kumar et al. [30] employs video features based on optical flow and the image edge maps. CardiacVC [37] uses the multi-class Logit-Boost algorithm to perform 4-way echocardiogram view classification. Wu et al. [53] employ low-level image features for 8-way echocardiogram classification. All of these methods classify echocardiograms to one of a discrete set of canonical views. Our method generalizes this problem to the regression setting to consider “in between” views and provides guidance to the operator for obtaining the desired view.

2.2.2 Self-calibration and egomotion

Outside of the domain of biomedical image analysis, there is a large amount of work on sensor self-calibration and egomotion estimation. Our work shares similarity to these approaches in that the pose of the sensor is estimated directly from the visual data. Cao and Shah [6] detect vertical lines and their shadows for calibration. Another method [57] calibrates cameras using appearance and motion in videos of traffic scenes. The method of Zhang et al. [58] estimates the parameters of a lens-distorted camera directly from low-rank textures. Koch and Teller [28] estimate the 6-DOF egomotion of an omnidirectional camera given a coarse 3D model of the environment. Domke and Aloimonos [12] use Gabor filters to compute correspondences and estimate egomotion. Our problem, echocardiogram localization, differs from these methods in multiple ways. First, self-calibration approaches typically seek to compute coordinates in a metric space, which is unnecessary for probe orientation. Second, most of these methods assume rigid object motion, whereas human hearts exhibit deformable motion.

2.3 Method

Echocardiograms are generated from the transmission and reflection of high frequency sound waves through human tissue. In the case of 2D transthoracic echocardiography, the most commonly applied variant of cardiac ultrasound, the resulting images represent a “slice” of the heart, as shown in Figure 2.2. Typically, images are acquired from a small set of predefined standard views. These standard views are described using alphanumeric codes, which refer to the combination of transducer location window: parasternal (PS), apical (A), or subcostal (SC), and image plane: long-axis (LX), short-axis (SX), four-chamber (4C), or two-chamber (2C). For example, A4C stands for the apical, four-chamber viewpoint.

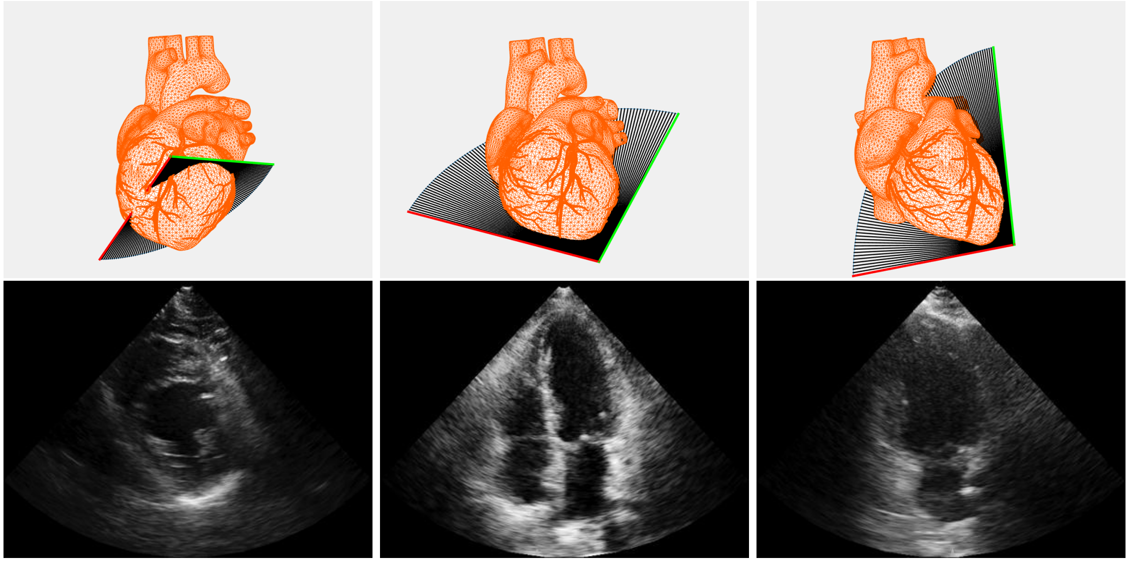


Figure 2.2: Each column shows the position of the transducer relative to the heart (top) and an output echocardiogram from that viewpoint (bottom). These examples correspond to the PSSX, A4C, and A2C viewpoints, respectively.

The image formation model is a complex function of the subject's body composition, ultrasound signal characteristics, and the position and orientation of the probe. We consider a simplified model, where the resulting echocardiogram image, I, for a a given subject, is an unknown, nonlinear function of the pose parameters, Θ, and the auxiliary causes of image variation are not explicitly modeled. Our goal is to infer the sensor pose parameters for a given image. This is a generalization of the echocardiogram view classification problem where a given image is classified into one of a discrete set of standard views. In fact, our method builds upon view classification to estimate the pose of the probe in the case of both standard and non-standard positioning.

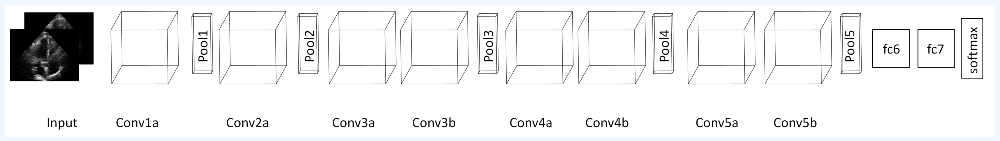


Figure 2.3: The network architecture of the C3D model [47]. The network contains 8 convolutional layers, 5 pooling layers, 2 fully connected layers and a softmax output layer.

We finetune one pretrained network for our goal of echocardiogram classification because of limited data we have. Figure 2.3 shows the network architecture of C3D [47], a 3D-CNN trained for generic video (2D+T) analysis, such as action recognition from image sequences. Compared to other popular convolution nets (e.g., [29]) for (2D) image classification, C3D is composed of 3D convolution and pooling operations. For echocardiogram view classification, the input is a sequence of echocardiogram frames. The output is a posterior distribution over the different view classes (e.g., A4C, PSSX).

We assume that echocardiogram images lie on or near a low-dimensional manifold parametrized by the viewpoint of the sensor. In this framework, we treat the posterior probabilities of class membership as barycentric coordinates on this manifold. Figure 2.4 shows an example of barycentric interpolation on an echocardiogram viewpoint manifold with four standard viewpoints.

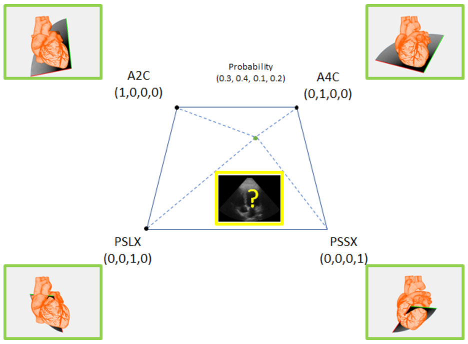


Figure 2.4: The posterior class probabilities are treated as barycentric coordinates to interpolate the position of a query image on the echocardiogram viewpoint manifold.

For an input image sequence, we obtain class posterior probabilities, $\hat{\mathbf{p}}$, as the output of the trained network. The goal is to estimate the probe pose parameters, $\hat{\Theta}$, of the input. Let Θ_i^* represent the ideal pose parameters for the i^{th} standard view. The inferred pose of sensor probe for the input echocardiogram is computed as the convex combination of pose parameters for the standard views:

$$\hat{\Theta} = \sum_{i=1} \hat{p}_i \cdot \Theta_i^* \tag{2.1}$$

where $\hat{p}_i$ is the class posterior probability for the i_{th} standard viewpoint.

2.4 Results and Discussion

We evaluate our approach on both the problem of echocardiogram view classification for standard views and also probe pose localization. For both problems, we apply our method to real-world data and compare to related approaches. Training and testing for all methods were carried out on a standard PC with a K40 Tesla GPU.

2.4.1 Data

Data was collected using a Philips Healthcare iE33 xMATRIX Ultrasound System. Echocardiogram scans were collected from 60 patients from four viewpoints: apical 2-chamber (A2C), apical 4-chamber (A4C), parasternal long axis (PSLX) and parasternal short axis (PSSX), which are four of the most commonly used views in echocardiography. Figure 2.5 shows sample echocardiograms used in these experiments.

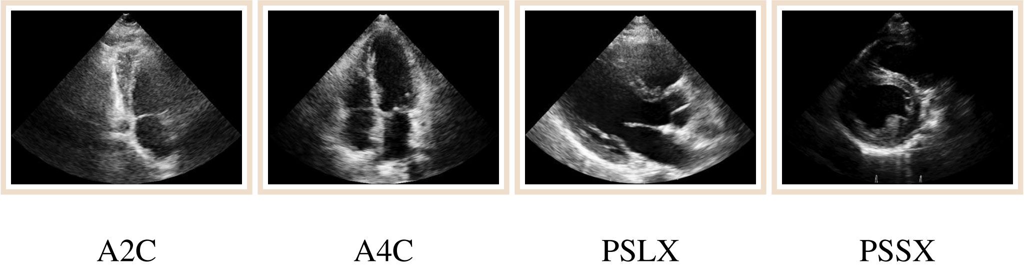
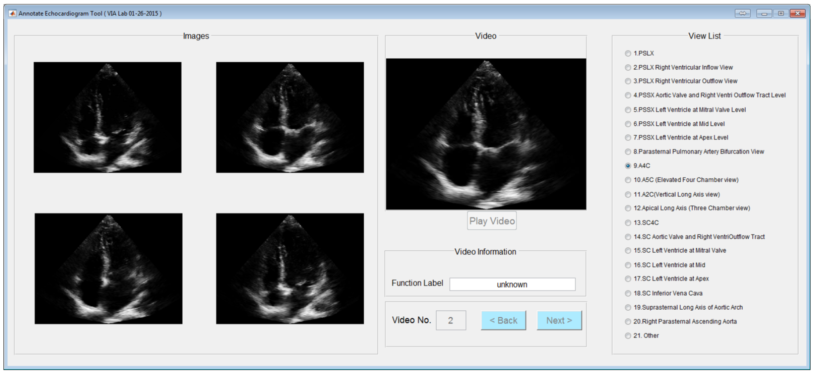


Figure 2.5: Sample echocardiograms and the view labels used for view classification.

To ensure the reliability of the view labels, the viewpoints of all echocardiogram scans are annotated by an experienced echocardiography technician. We design an GUI to assist the annotation of the echocardiogram data. The screenshot of the annotation tools is shown in Figure 2.6.



Annotate Echocardiogram Tool (VIA Lab 01-26-2015)

Function Label: unknown

Video No. 2

View List

○ 1.PSLX
○ 2.PSLX Right Ventricular Inflow View
○ 3.PSLX Right Ventricular Outflow View
○ 4.PSSX Aortic Valve and Right Ventri Outflow Tract Level
○ 5.PSSX Left Ventricle at Mitral Valve Level
○ 6.PSSX Left Ventricle at Mid Level
○ 7.PSSX Left Ventricle at Apex Level
○ 8.Parasternal Pulmonary Artery Bifurcation View
● 9.A4C
○ 10.A5C (Elevated Four Chamber view)
○ 11.A2C(Vertical Long Axis view)
○ 12.Apical Long Axis (Three Chamber view)
○ 13.SC4C
○ 14.SC Aortic Valve and Right VentriOutflow Tract
○ 15.SC Left Ventricle at Mitral Valve
○ 16.SC Left Ventricle at Mid
○ 17.SC Left Ventricle at Apex
○ 18.SC Inferior Vena Cava
○ 19.Suprasternal Long Axis of Aortic Arch
○ 20.Right Parasternal Ascending Aorta
○ 21. Other

Figure 2.6: User interface of the tools we designed for echocardiogram viewpoint annotation.

For a given patient, each view corresponds to a separate echocardiogram video clip, whose duration ranges from 3 to 10 heartbeat cycles. Because the data was collected in a real-world clinical setting, not all four views were collected. In some cases, the echocardiograms were corrupted or otherwise unusable. Table 2.1 shows the number of echocardiogram video in each view we used in the experiment.

Table 2.1: Number of echocardiogram videos in the data set

Views	**A2C**	**A4C**	**PSSX**	**PSLX**
# Count	59	65	8	52

2.4.2 Implementation Details

For our approach, we start with the C3D convolutional neural network, which is initially trained on the Sports-1M dataset [25]. The finetuning process is shown in Figure 2.7. We modify the output layer for four-class (A2C, A4C, PSSX, PSLX) prediction.

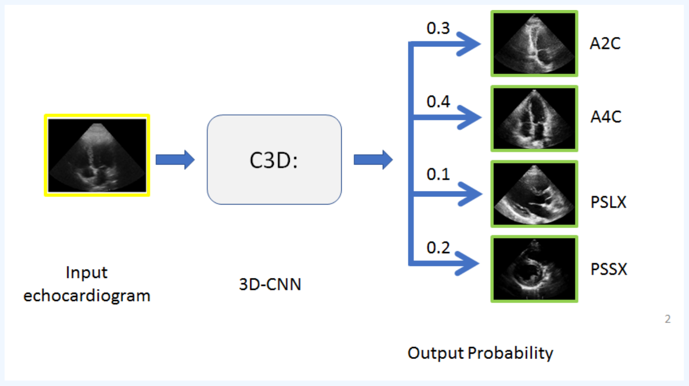


Figure 2.7: The finetuning of 3D-CNN for fine-grained echocardiogram classification.

The original C3D network takes as input 16 sequential color frames of video; we modify the input layer to take a sequence of echocardiograms. Each frame is resized to 128×171 to make sure it matches the input size of C3D. Using the labeled echocardiogram training data, we fine-tune the output layer of the network to this task. With a learning rate of 0.001, training to convergence takes roughly 3000 iterations and 6 hours. For each learning experiment using the labeled echocardiograms collected from the Philips machine, data from 50 subjects were used for training and 10 subjects were used for testing.

2.4.3 Sequence Length

The C3D network was designed for generic video analysis of short clips. For this problem, we evaluated video clips of varying length. Figure 2.8 shows the accuracy on a 4-way discrete classification task as a function of the input clip length. For each input size, we repeated the classification task 5 times. Based on the results, we selected $N = 2$ frames as the input length for the subsequent experiments, which roughly corresponds to ~ 0.1 seconds of real time and provides a balance between accuracy and real-time performance.

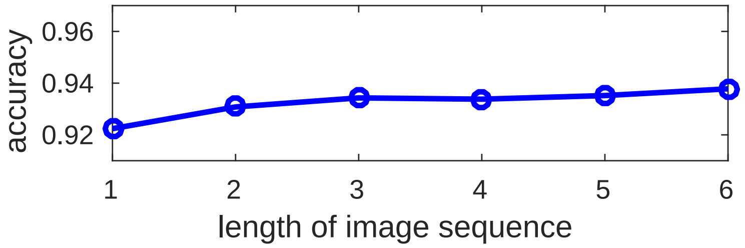


Figure 2.8: Classification accuracy as a function of input image sequence length.

2.4.4 Echocardiogram View Classification

We evaluated the C3D-based model on a 4-way (A2C, A4C, PSSX, PSLX) classification task and compare against the following methods:

- **Baseline** Each input image is represented using the HOG [11] feature descriptor and classified using kernel logistic regression (KLR). HOG features are computed in 20×20 pixel cells and 4×4 blocks are used to normalize gradients, resulting a 2304-dimensional feature vector. The radial basis kernel ($\sigma = 0.01$) is used for KLR.

- **Wu2013** This recently-developed method for view classification [53] uses the GIST feature descriptor [35] and SVM for classification. GIST was computed in 4×4 blocks and, within each block, 8 orientations for each of the three filters of different scales are computed, resulting in a 384-dimensional feature vector. The SVM uses the radial basis kernel ($\gamma = 64$) and regularization parameter, $C = 10$.

- **CNN** For this CNN baseline, we fine-tuned the popular CaffeNet [24] to echocardiogram image classification. Compared to C3D, this network is based on single-frame input and mainly 2D convolution operations. The CNN training settings were the same as for the C3D network.

Table 2.2: Accuracy of views prediction on testing data

Method	Baseline	Wu2013	CNN	**3D-CNN**
Accuracy	73.2%	80.7%	88.5%	**94.1%**

The prediction accuracy on testing data are reported in 2.2. The CNN approaches outperform the other methods, with 3D-CNN achieving ∼6% higher accuracy than CNN. Figure 2.9 shows the confusion matrix for the 3D-CNN method on this task. Except for a small amount of confusion between A2C and PSSX, most of the confused predictions were between typically challenging pairs of viewpoints similar in visual appearance (e.g., A2C vs. A4C).

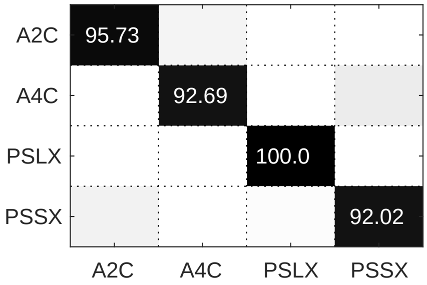
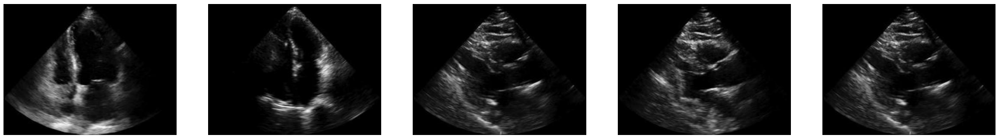


Figure 2.9: Confusion matrix for 3D-CNN method on 4-way echocardiogram view classification.

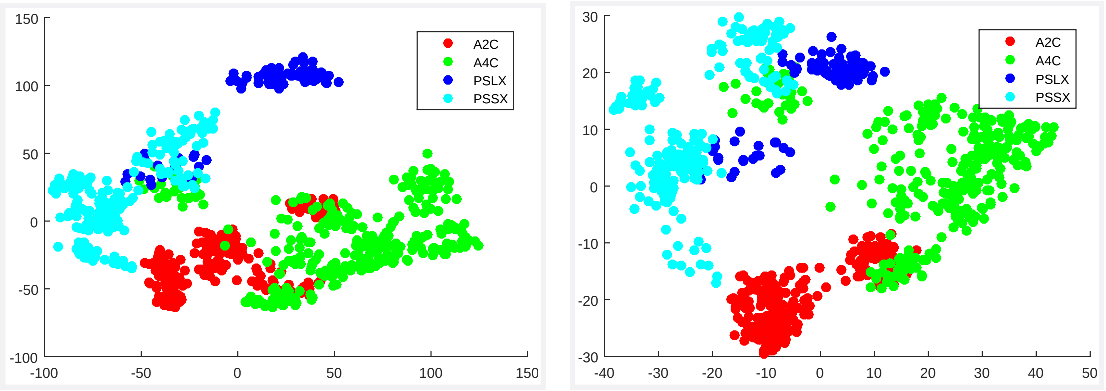
We select a few representative testing example to analyze. Figure 2.10 shows example test images and the output classification for each approach. In the first example, the view of the right atrium (bottom left) is heavily corrupted by image noise. This results in two methods incorrectly predicting the two-chamber (rather than 4) apical view. The second and third images are challenging due to probe positioning and imaging settings, respectively.



BASE	A2C	A4C	A4C	PSLX	PSLX
Wu2013	A4C	A2C	PSLX	PSLX	PSLX
CNN	A2C	A4C	PSLX	PSLX	PSLX
3D-CNN	A4C	A4C	PSLX	PSLX	PSLX

Figure 2.10: Results (correct = green, incorrect = red) on representative examples for the 4-way classification task.

To further investigate the model learned by the 3D-CNN network, we visualize the high-level feature representations of the test examples. Figure 2.11 shows the 2-dimensional multidimensional scaling (MDS) embedding of the 4096-D fc6 (fully connected layer 6) and fc7 (fully connected layer 7) of the 3D-CNN network where the color of each point represents the ground truth label. Both layers show the type of intra-class similarity and inter-class differences the enable discrimination. At the fc7 layer, most of the examples form disjoint clusters.



(a) fc6 layer

(b) fc7 layer

Figure 2.11: 2D MDS embedding of echocardiogram features extracted using fc6 and fc7 layers in 3D-CNN.

2.4.5 Echocardiogram Localization

Echocardiogram localization is a generalization of the classification problem. For this task, we collected non-standard data using a SonoSite M-Turbo Ultrasound Machine. To build this data set, the technician moved the probe to non-standard positions “in-between” the standard views. This data was obtained from three volunteers, and the stream from each volunteer consists of roughly 1800 frames of video. Each frame of this data was also annotated by the domain expert as one of the 4 standard views or “non-standard”.

For the two CNN approaches, we train the network for discrete classification and apply barycentric interpolation as described in Section 2.3. The probe pose is parametrized by the location sensor location in 3D and the rotation of the imaging plane in a reference co-ordinate system. The pose parameters of the four canonical viewpoints (A4C, A2C, PSLX, PSSX) were obtained by manual alignment with the probe positions from an expert technician at the respective positions. Figure 2.12 and Figure 2.13 show representative results for echocardiogram localization and probe pose estimation using CNN and 3D-CNN. While hardware-based probe trackers have been proposed in the literature (e.g., [19]) and commercially, these experiments were performed with the type of sensor typically found in medical settings, without tracking capabilities. Therefore, the results were evaluated qualitatively. In Figure 2.12, the probe was moved smoothly between the standard A2C and PSSX views. As the probe moves, new structures are visible in the bottom portion of the image, however the prediction from the CNN model remains unchanged.

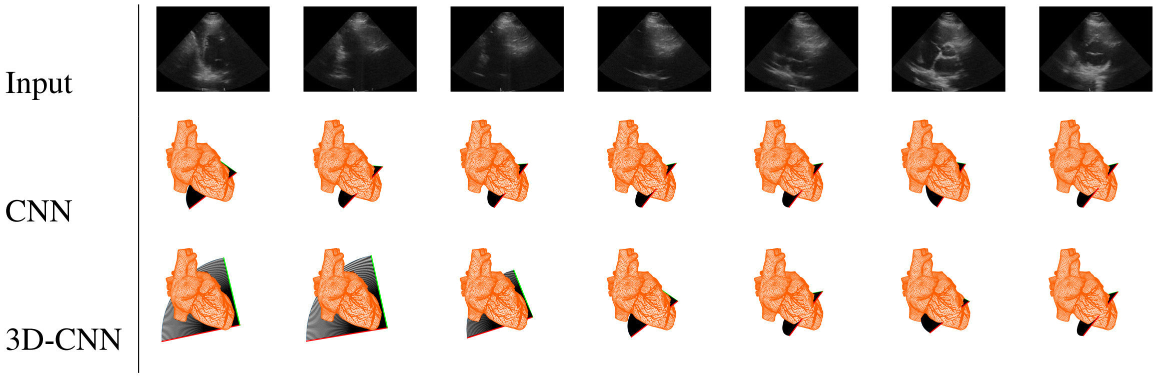


Figure 2.12: Probe pose estimation for non-standard views between A2C and PSSX. In each column, the visualization shows the probe pose prediction for the echocardiogram. The first row are the input moving echocardiograms (from left to right); while the estimated probe poses using CNN and 3D-CNN are shown in second and third row, respectively.

Similarly, in Figure 2.13, the probe was moved gradually between the A4C and PSLX views. 3D-CNN provides smoothly changing pose estimates for these sequences. In both cases, the quality of the prediction follows the trend of the discrete classification results, with 3D-CNN showing estimates, which most closely matched the probe motion applied by the technician.

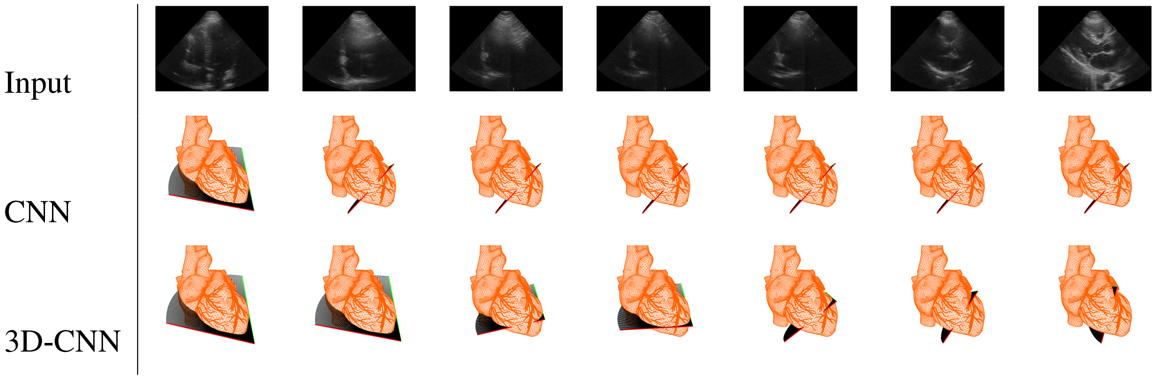


Figure 2.13: Probe pose estimation for non-standard views between A4C and PSLX. In each column, the visualization shows the probe pose prediction for the echocardiogram. The first row are the input moving echocardiograms (from left to right); while the estimated probe poses using CNN and 3D-CNN are shown in second and third row, respectively.

Currently, most ultrasound training follows the apprentice model where an experienced technician provides guidance to a novice. Most of this guidance is provided using general rules of thumb (e.g., “start at the 4th or 5th intercostal space”, “orient the probe to 3 o’clock”, “rotate towards the patient’s shoulder”, “tilt the probe until the relevant structures are visible”) that must be adapted to each patient’s body shape.

Our application prototype is based on probe pose estimation from echocardiograms. Figure 2.14 shows a screenshot of the application with the echocardiogram output displayed alongside a visual representation of the estimated probe orientation. For a selected view (e.g., A4C, PSLX), the arrows indicate the direction the transducer should be repositioned based on the difference between the estimated pose and the reference pose. In a similar manner to human trainers, the arrows will instruct the user to rotate the head or the tail of the probe. In a teaching setting, such an application could be used to increase the amount of practice time a technician-in-training receives. For a new echocardiographer, this application may help in finding usable images faster, thus decreasing the amount of time a patient is exposed to the ultrasound as well as the patient’s comfort level. For the experienced echocardiographer, such an application could be used in cases involving difficult body habitus that is the result of surgery, size of the patient, or additional signal interference introduced by lung diseases.

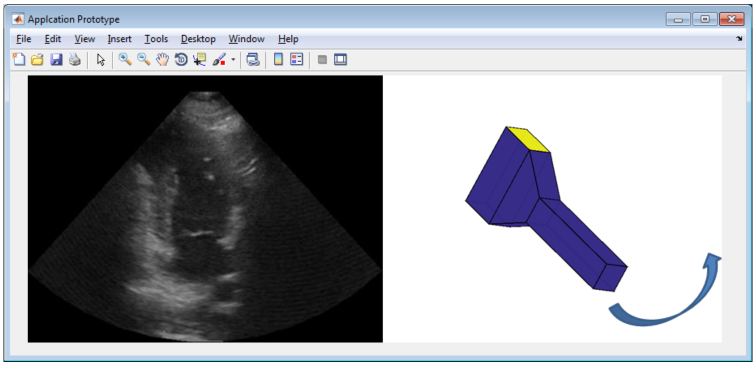


Figure 2.14: A screenshot of the application prototype with the echocardiogram output displayed alongside a visual representation of the estimated probe orientation. For a selected view (e.g., A4C, PSLX), the arrows indicated the direction the transducer should be repositioned.

2.5 Summary

In this chapter, we present a method for fine-grained echocardiogram view classification from videos. Our method is based on finetuning 3D CNN. This problem represents the class of tasks having small scale of data but clean and balanced labels. The proposed method employs pretrained model previously trained on large but unrelated data set. We finetune the model with limited echocardiogram data we collected. We augment the training data through many techniques. We compare the method against several other methods. The experiment result shows that the proposed method work best on echocardiogram classification. We demonstrate the effectiveness of 3D CNN on solving video analysis task with small amount of labeled training data.

CHAPTER 3: Autoencoder with One-Class Loss for Rare Event Video Detection

Echocardiograhy is one of the most reliable ways to diagnose human heart diseases. Automated evaluation of heart healthiness from echocardiogram is a challenging task, largely because there are large variations in the impaired hearts and very limited learning examples. Figure 3.1 shows three healthy echocardiograms and three impaired echocardiograms. As can be seen from the figure, the difference between healthy and impaired echocardiograms is barely noticeable. Therefore, the impaired echocardiogram detection task belongs to the fine-grained classification realm. However, the data imbalance between normal and impaired hearts pose another huge challenge to the problem. In this chapter, we frame this task as an anomaly detection problem between two types of highly similar videos. We develop a 3D convolutional autoencoder with one-class layer. We use only normal echocardiograms to train a model and use it to predict the healthiness of the unseen echocardiogram videos. This method solves a fine-grained video classification problem given unbalanced dataset. We evaluate the performance of the proposed method with real-world dataset collected in clinical settings. The experimental results show that our method outperforms the traditional methods.

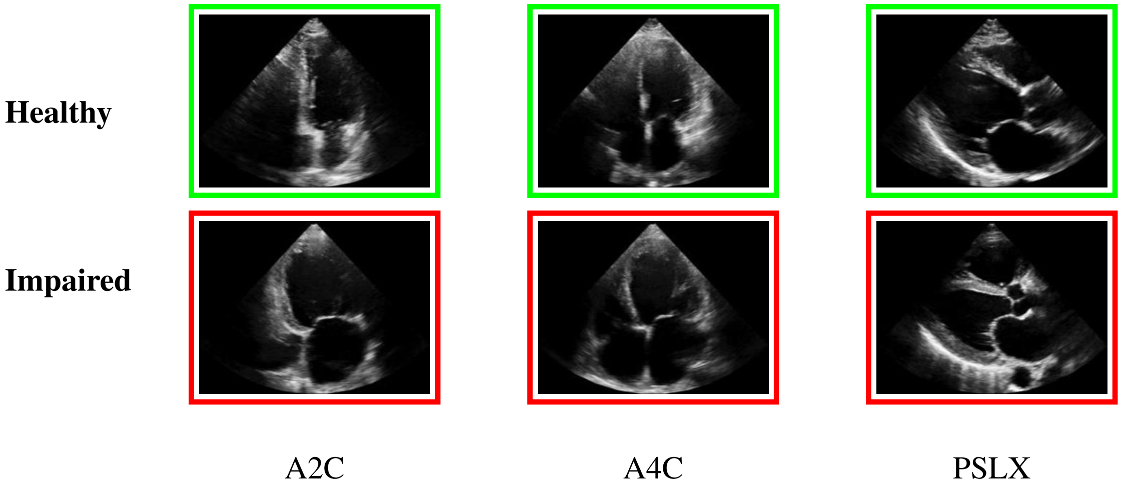


Figure 3.1: Three echocardiograms from healthy human hearts are in the top row (enclosed by green rectangle), another three echocardiograms with heart disease are shown in the bottom row (enclosed by red rectangle). The three columns correspond to A2C, A4C, and PSLX viewpoints, respectively.

3.1 Problem Statement

In many real-world video classification problems, videos are not equally distributed in all categories. In its simplest form, the dominant class contains most of the data and the rest of data belong to the minor class. Data imbalance makes classifier hard to train. Since the minor class contains few training examples, the learned classifier tends to classify all unknown data in the testing into majority class, as illustrated by Figure 3.2. In real-world problems such as diagnosis of diseases, if a potential heart disease is diagnosed as healthy it could lead to fatal consequences. On the contrary, if a heart disease can be detected in early development of disease, timely treatment can prevent heart disease from developing into more deadly stages.

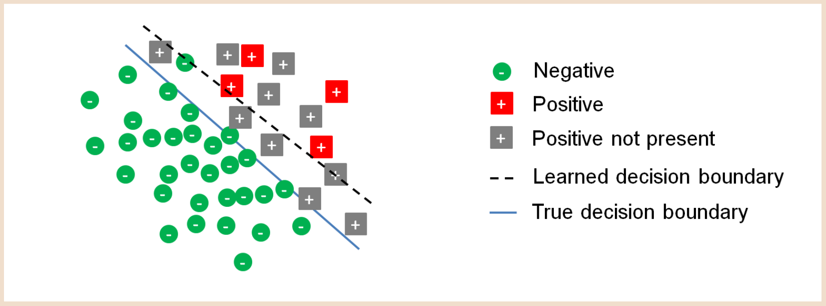


Figure 3.2: Data imbalance makes classifier hard to train.

This type of data imbalance issue is not uncommon in computer vision and general pattern recognition area. In conjunction with fine-grained difference between classes, problems get even harder to solve. For instance, echocardiogram function label classification, is challenging due to insufficient impaired echocardiogram examples. There are several possible reasons for the scarcity of impaired echocardiograms. Firstly, patients with heart diseases are less common than patients with healthy heart both in real world and in our data set. Secondly, patients with severe heart diseases tend to less actively share their echocardiogram with third party for research purpose. These factors together make impaired echocardiogram underrepresented in the collected dataset. (In our dataset, impaired echocardiogram videos is less than 5%).

3.2 Related Work

There have been extensive methods to address the data imbalance problem in generic pattern recognition area, such as resampling [8]. Typically, the resampling method reduces the dominant data by randomly sampling a fraction of it. It can also enlarge the minor data using multiple sampling with replacement. Other approaches introduces hard-negatives mining [15, 33] to iteratively discard easy examples that are relative far from the true decision boundary. In the end, only the *hard* examples that are close to the true decision boundary are used for final classifier training. Note that there is a common limitation of above methods: simple random resampling or hard-negatives mining mitigate the problem to some extent but it can't eradicate data imbalance.

3.2.1 Anomaly Detection

Anomaly detection [7] trains on mostly one class of data for the purpose of identifying object not belonging to this class. Any data does not belong to the training class will be viewed as anomaly. Among many anomaly detection algorithms, one-class support vector machine (OCSVM) [41, 46] learns a separating hyperplane by using only the majority class, as illustrated by Figure 3.3. OCSVM is widely used in pattern recognition problems when one single class is dominant in the dataset. Wu et al. [54] employs OCSVM to classify normal and impaired echocardiograms.

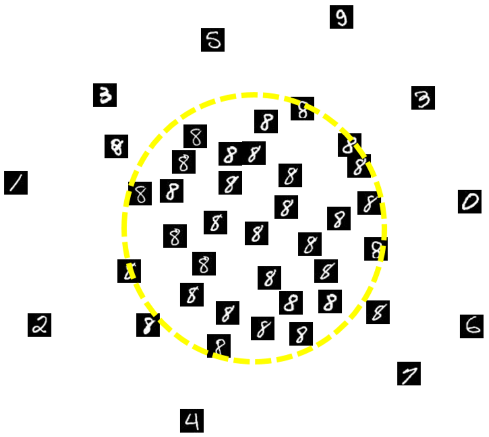


Figure 3.3: Illustration of one-class SVM for anomaly detection. In the figure, handwritten digit images from MNIST data set are shown. The digit "8" is the dominant class and all the other digits are minor class (anomaly). One-class SVM learns a hyperplane separating the dominant class from the minor class. Most data in the dominant class are encompassed by the separating hyperplane.

3.3 Method

To detect abnormal video event, we build on 3D convolutional autoencoder with a one-class SVM layer. Figure 3.4 illustrates our model architecture. The encoding layers in the autoencoder are consist of 3D convolutional kernels, and the decoding layers are consist of 3D deconvolutions (or upsamplings) kernels. We use a one-class SVM layer to add regularization on the on learned feature of the autoencoder. The total loss is a combination of reconstruction loss between the input and output, and the loss of the one-class layer.

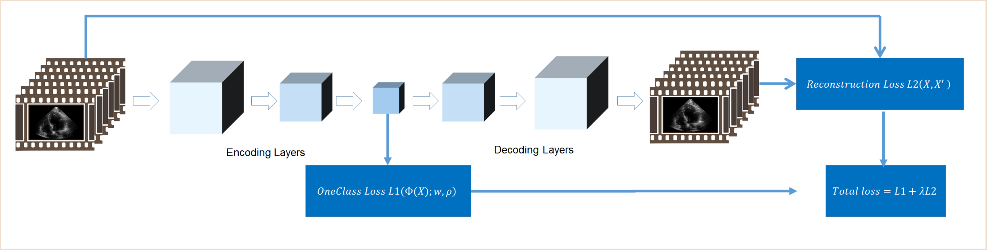

Encoding Layers
Decoding Layers
Reconstruction Loss $L2(X, X')$
OneClass Loss $L1(\Phi(X); w, \rho)$
Total loss $= L1 + \lambda L2$

Figure 3.4: 3D convolutional autoencoder with an one-class SVM layer

For reconstruction loss, mean squared error (MSE) and cross entropy loss are typically used in image-related and video-related problems. For one-class layer, we employ the loss function in OCSVM, which was proposed by Schölkopf et al. [41] for novelty detection. The loss function can be expressed as follows:

$$\begin{aligned} L_{ocsvm} &= \frac{1}{2}\|\mathbf{w}\|^2 + \frac{1}{vn}\sum_{i=1}^{n}\xi_i - \rho \\ \text{subject to } (\mathbf{w}^\top \Phi(x_i)) &\geq \rho - \xi_i, i = 1, \ldots, n, \\ \xi_i &\geq 0, i = 1, \ldots, n, \end{aligned} \tag{3.1}$$

where, Φ is a function that maps the input into feature space, $v \in (0, 1)$ a hyperparameter controlling the maximum percentage of anomalies, $\mathbf{w}$ and ρ are parameters learned from the training set.

We implement the OCSVM layer using Keras [9], a popular deep learning library. Once the model is trained, the encoding layers of the autoencoder is used as a feature extractor in the testing stage. Given a testing video x_{test}, we first extract feature from it using the encoder layers. Then the decision function of OCSVM, $\mathbf{w}^\top \Phi(x_{test}) - \rho$, computes the signed distance of x_{test} to the hyperplane that separates normal data and anomaly in the feature space. If distance is positive, the testing video is classified into normal class. Otherwise, testing video is classified as anomaly.

3.4 Result and Evaluation

We evaluate our method with two datasets.

- **Handwritten digit recognition** We use handwritten digit images from MNIST dataset [31] to validate the initial idea of combining reconstruction loss and one-class loss. In this experiment on toy data, we use 2D convolutions instead of 3D convolutions.

- **Impaired Echocardiogram Detection** We employ the proposed autoencoder with one-class layer to classify the function label of echocardiogram. We also implement multiple other methods for comparison.

In both experiments, we use area under the receiver operating characteristic curve (AUROC) to evaluate multiple algorithms. In most classification problems, precision and recall are inversely related. By adjusting the decision threshold, one could tune any method's performance from the highest recall (the lowest precision) to the highest precision (the lowest recall). Therefore the performance of two methods cannot be compared using one pair of precision-recall value. AUROC provides an unbiased way to compare different algorithms.

3.4.1 Handwritten Digit Recognition

3.4.1.1 Data

The MNIST dataset is a large handwritten digit database that consists of 70,000 digit images, 60,000 images of training data and 10,000 images of testing data. It is widely used as a benchmark to evaluate the performance of various machine learning and computer vision algorithms. The dataset contains 10 common digits written by human. The size of the grayscale images is 28×28. A few examples of images in MNIST are shown in the Figure 3.5.

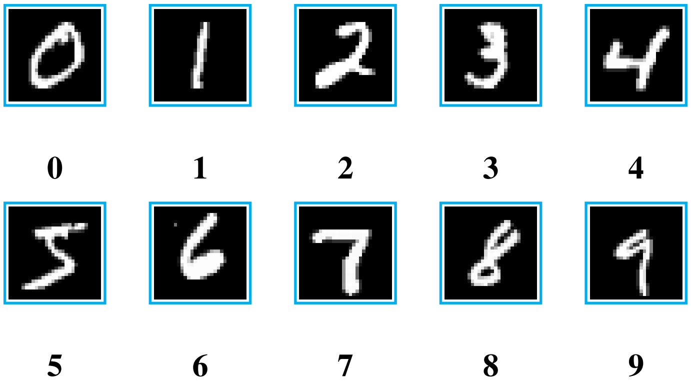


Figure 3.5: Examples of handwritten digit images in the MNIST database

3.4.1.2 Experiment setup

The experiment steps are explained as follows. First, we random select one digit (such as “0”) as normal data, and treat the remaining digits (such as “1” to “9”) as anomaly. Second, we create the training set and testing set for the experiment. The training dataset contains all normal digits from the 60,000 original MNIST training images set, and testing data set contains both normal digits and abnormal digits from the 10,000 original MNIST testing images.

Since anomaly data is uncommon in many real application scenarios, to emulate the situation in real-world problems, we explicitly control the ratio between normal and abnormal data in the testing set to make sure it is close to 80:20.

For abnormal digit detection, we train a 2D autoencoder with convolutional autoencoder with OCSVM layer. The network architecture is shown in the following Figure 3.6. The only part that differs from the proposed method for impaired echocardiogram detection is the 2D convolutions instead of 3D convolutions. The input size of the network is 28x28, corresponding to the original dimension of MNIST images. After three stages of convolutions and max pooling, the feature map's size is reduced to 1x32, followed by a fully connected layer. The OCSVM layer is attached to the end of fully connected layer.

During the training phase, we alternately switch between two phases, Phase 1 and Phase 2. In Phase 1, the autoencoder optimization phase, we focus on optimizing the weights in autoencoder layers, so the learning rate of One-Class SVM layer is set to a smaller value. (Through our experiment, we choose 10%.) In Phase 2, the One-Class SVM optimization phase, we mainly optimize OCSVM layer, thus we set the learning rate of autoencoder layers to a smaller rate. We repeat two optimization phases multiple times until the loss converges. We choose is AdaDelta's [55] as the automatic learning rate deceasing policy. The initial learning rate is set to 1.0. The implementation of AdaDelta is provided by Keras [9] library.

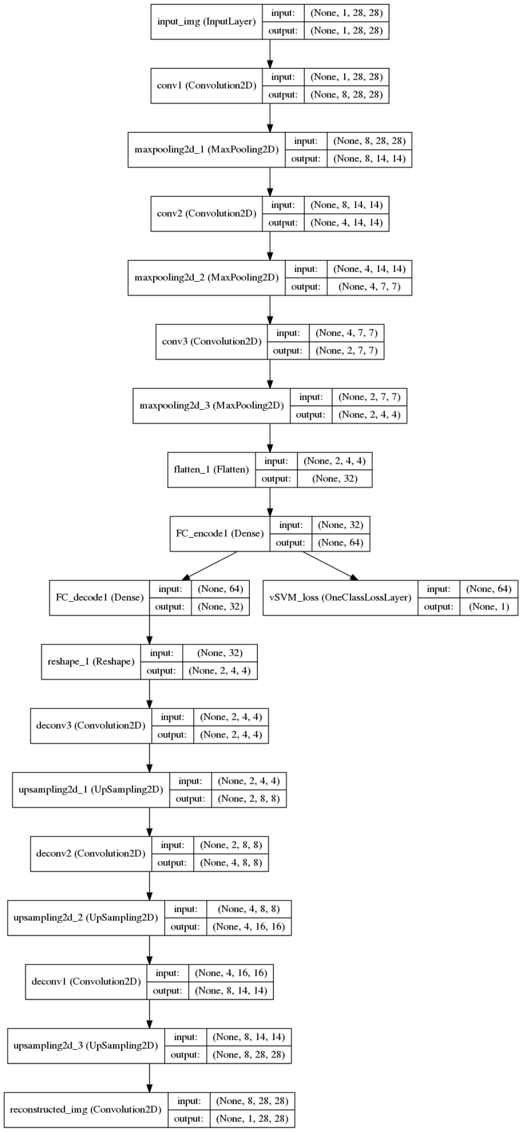

input_img (InputLayer) input: (None, 1, 28, 28) output: (None, 1, 28, 28)
conv1 (Convolution2D) input: (None, 1, 28, 28) output: (None, 8, 28, 28)
maxpooling2d_1 (MaxPooling2D) input: (None, 8, 28, 28) output: (None, 8, 14, 14)
conv2 (Convolution2D) input: (None, 8, 14, 14) output: (None, 4, 14, 14)
maxpooling2d_2 (MaxPooling2D) input: (None, 4, 14, 14) output: (None, 4, 7, 7)
conv3 (Convolution2D) input: (None, 4, 7, 7) output: (None, 2, 7, 7)
maxpooling2d_3 (MaxPooling2D) input: (None, 2, 7, 7) output: (None, 2, 4, 4)
flatten_1 (Flatten) input: (None, 2, 4, 4) output: (None, 32)
FC_encode1 (Dense) input: (None, 32) output: (None, 64)
FC_decode1 (Dense) input: (None, 64) output: (None, 32)
vSVM_loss (OneClassLossLayer) input: (None, 64) output: (None, 1)
reshape_1 (Reshape) input: (None, 32) output: (None, 2, 4, 4)
deconv3 (Convolution2D) input: (None, 2, 4, 4) output: (None, 2, 4, 4)
upsampling2d_1 (UpSampling2D) input: (None, 2, 4, 4) output: (None, 2, 8, 8)
deconv2 (Convolution2D) input: (None, 2, 8, 8) output: (None, 4, 8, 8)
upsampling2d_2 (UpSampling2D) input: (None, 4, 8, 8) output: (None, 4, 16, 16)
deconv1 (Convolution2D) input: (None, 4, 16, 16) output: (None, 8, 14, 14)
upsampling2d_3 (UpSampling2D) input: (None, 8, 14, 14) output: (None, 8, 28, 28)
reconstructed_img (Convolution2D) input: (None, 8, 28, 28) output: (None, 1, 28, 28)

Figure 3.6: Network architecture of the autoencoder used in MNIST experiment

3.4.1.3 Result

Table 3.1: AUROC of the anomaly digit recognition

Normal Digit	Abnormal Digit	AUROC
“0”	non-“0”	0.9736
“1”	non-“1”	0.9988
“2”	non“2”	0.9313
“6”	non“6”	0.9809

We repeat the experiment 10 times, using each digit as the normal data. Since there are 10 different experimental results, we select four of them to report. The ROC curves of anomaly digit recognition in four experiments are shown in Figure 3.7 and corresponding AUROC values are reported in Table 3.1. The average AUROC of four experiments is 0.9711, indicating the proposed anomaly detection mostly performs well. The lowest AUROC we obtain is when digit “2” is selected as normal data, and used to detect non-“2” digits. In the testing, abnormal digit such “5” is confused with of “2”. The experiment result also shows that the difficulty of anomaly detection of handwritten digits depends on the similarity between normal digit and abnormal digits.

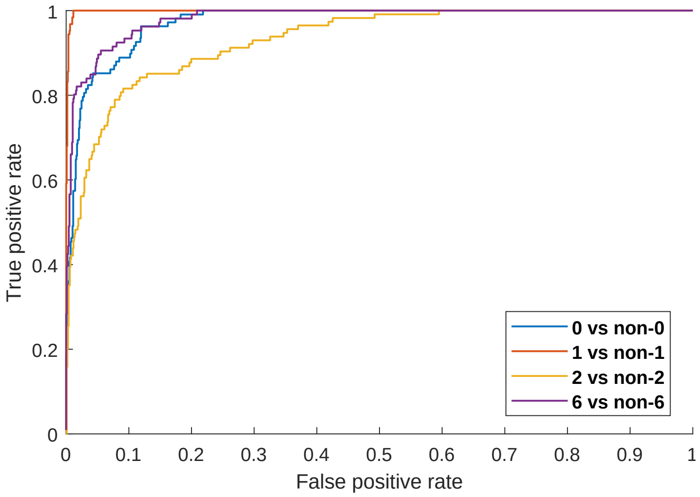
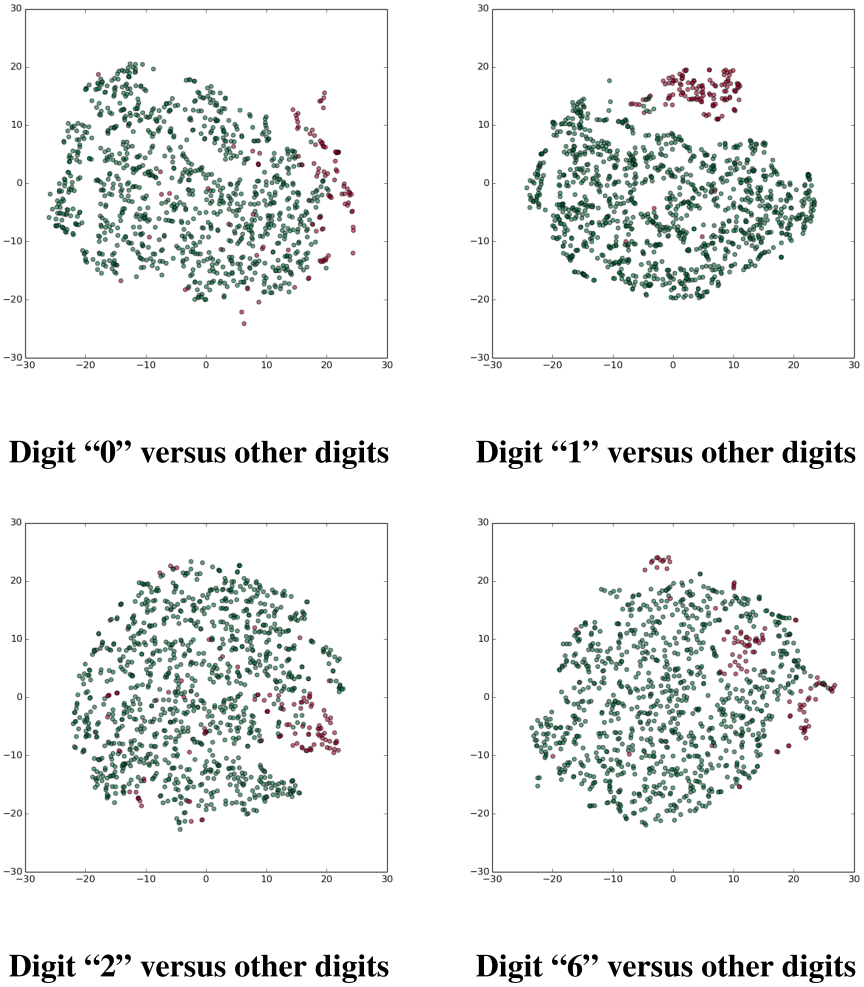


Figure 3.7: The ROC curve of experiment on MNIST data set

We also investigate the learned features of autoencoder. We use t-Distributed Stochastic Neighbor Embedding (t-SNE) [50, 49], a nonlinear dimensionality reduction tool, to visualize the learned features of testing data. The 2D embeddings of learned feature representation of testing data in four experiments are shown in Figure 3.8. We can find that in 2D embedding space, nearly all abnormal data (in red color) distribute in the outer edge of normal data.



Digit “0” versus other digits **Digit “1” versus other digits**

Digit “2” versus other digits **Digit “6” versus other digits**

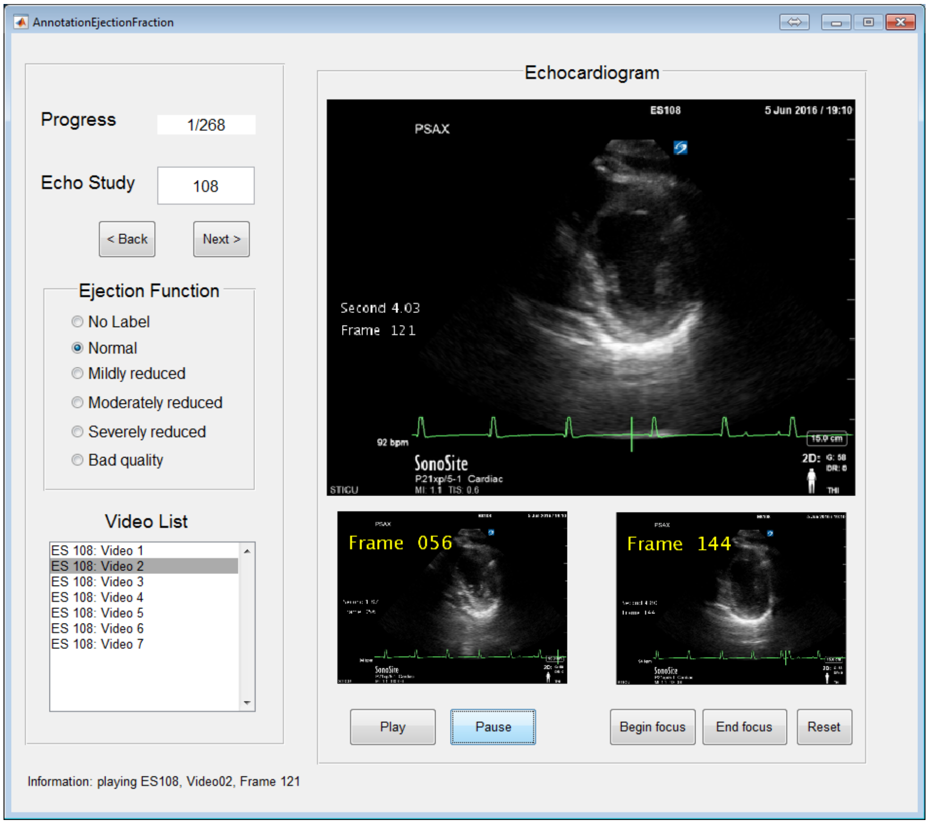
Figure 3.8: 2D t-SNE Embedding of handwritten digit images’ learned features. Normal digits are represented by green points and red points mean anomaly digits. We can find that in 2D embedding space, nearly all abnormal data (red) distribute in the outer edge of normal data.

3.4.2 Impaired Echocardiogram Detection

In this section, we describe the experiment setting for the impaired echocardiogram detection problem and report the comparison result.

3.4.2.1 Data

We collect echocardiogram data from around 200 patients who need accurate diagnosis based on echocardiogram in clinical settings. The echocardiography ultrasound machines we used are SonoSite M-Turbo and SonoSite X-Porte. Four different viewpoints of echocardiogram scans were collected from : apical 2-chamber (A2C), apical 4-chamber (A4C), parasternal long axis (PSLX) and parasternal short axis (PSSX), which are most commonly used views in echocardiography. The function labels of all patients are annotated by six experienced echocardiography technicians from three categories: $Normal$, $Impaired$, and $Uncertain$. The screenshot of the annotation tool we design is shown in Figure 3.9.



Progress 1/268

Echo Study 108

Ejection Function

- ○ No Label
- ● Normal
- ○ Mildly reduced
- ○ Moderately reduced
- ○ Severely reduced
- ○ Bad quality

Video List

- ES 108: Video 1
- ES 108: Video 2
- ES 108: Video 3
- ES 108: Video 4
- ES 108: Video 5
- ES 108: Video 6
- ES 108: Video 7

Information: playing ES108, Video02, Frame 121

Figure 3.9: The screenshot of the tools we designed for echocardiogram function annotation.

We use Cohen's kappa to find the annotators that deviates a lot from the other annotators and eliminate all the annotations provided by the outlier annotator. Then we obtain the ground truth function labels using majority voting based of the remaining annotator's labeling. We discard the echocardiogram with $Uncertain$ ground truth labels, in order to reduce the ambiguity of function label in the experimental echocardiogram. Only the echocardiogram with $Normal$ and $Impaired$ ground truth labels are retained for the experiments. After the filtering, we keep 125 normal and 17 impaired patients' echocardiogram.

Each original echocardiogram video contains variable number of heartbeat cycles, usually between 3 to 10. We extract the ECG signal from the raw echocardiograms using digital image processing. The ECG signal is utilized to split the original echocardiogram into multiple short video clips. Each video clip starts with diastole (relaxation of heart muscles), it is followed by systole (contraction of heart muscles), ending right before next diastole. The numbers of echocardiogram video clips for all views are shown in Table 3.2. The purpose of this procedure is to align all echocardiogram in temporal domain, and it also guarantees a short video clip contains exactly one single heartbeat. The video clips are the immediate input of all experiments in the following.

Table 3.2: Number of echocardiogram video clips in the experiment

Echocardiogram View	# Normal	# Impaired	**# Total**
A2C	1658	277	1935
A4C	1731	318	2049
PSLX	1880	274	2154
PSSX	1772	242	2014

3.4.2.2 Experiment setup

We randomly split the data into five disjoint folds by patients (human subjects) and perform five-fold cross validations. For each validation, we run four experiments, each for one viewpoint from A2C, A4C, PSLX, and PSSX, independently. Therefore, we have 5x4=20 sets of experimental results for every single method.

The network architecture for impaired echocardiogram detection (only the encoder part) is shown in Figure 3.10. In total it contains four convolutional stages, 6 convolutional layers and 4 max pooling layer. The decoder is symmetric to the encoder, except that the convolution operation is replaced by deconvolutions. The input size of 3D convolutional neural network is 3x16x128x128, the size of learned feature representation is 512x8x8x8, meaning there are 512 channels of 8x8x8 cubes.

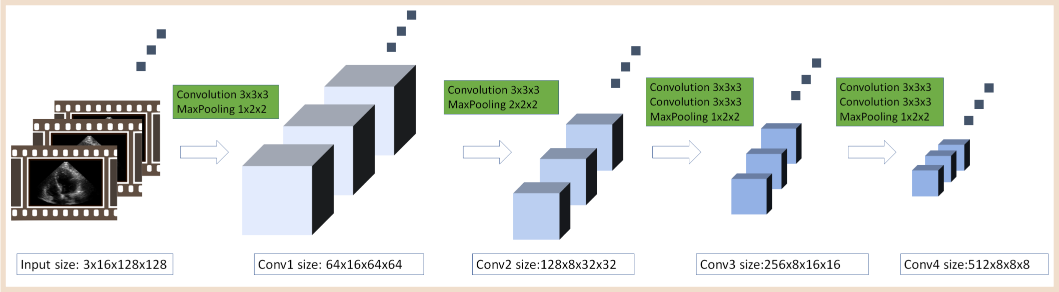


Figure 3.10: The encoder structure used in impaired echocardiogram detection experiment

As mentioned above, we train the model with only normal echocardiogram data, and test on both normal and impaired echocardiogram data. We implement the OCSVM using Keras [9], a deep learning library. All experiments are conducted on a workstation equipped with two Nvidia K-40c GPUs. Our method is compared against six baseline methods. In all experiments, we random select 10% of training data as validation set, and use it to select best hyperparameters. The detailed experimental setup of all methods are explained as follows.

- **3D autoencoder:from scratch** This is our proposed method, the 3D autoencoder with OCSVM layer. In this experiment, the entire model is trained from scratch, i.e, the weights are random initialized. The training elapsed time is 5 to 10 times longer than the training method using finetuning since it needs more iterations to converge.

- **3D autoencoder:fine-tune** This method is nearly the same as the method above. The only difference is the optimization process. For this method, the encoder's weights are initialized from the C3D [47] network's convolutional weights. However the decoder's weights and OCSVM layer's weight are still randomly initialized since there is no pretrained model for it.

- **3D CNN:fine-tune** This is supervised 3D CNN for the purpose of binary classification, "$Impaired$" vs "$Normal$" . It is also fine-tuned from the C3D model. In order to mitigate the imbalance of training samples, we augment the "$Impaired$" echocardiograms by resampling and adding various transformations so that the numbers of both classes are are to 50:50. This is the only supervised method while all other methods are unsupervised.

- **3D-CNN feature+linear OCSVM** This method extracts feature from the echocardiogram using pretrained C3D model and train anomaly detection using OCSVM with linear kernel. We extract the features from conv5 and fc6 layers. An earlier comparison experiments shows that fc6 features is better. Therefore we only use fc6 feature through all the experiment comparisons. We use grid search to determine the optimal hyperparameter.

- **3D-CNN feature+RBF OCSVM** This method is similar to *3D-CNN feature+linear OCSVM*, except that we choose radial basis function (RBF) kernel for OCSVM. We also use grid search to determine the optimal hyperparameter based on validation set.

- **VGG16 feature+linear OCSVM** This method is also similar to *3D-CNN feature+linear OCSVM*. We extract feature from the echocardiogram using VGG16 [43] instead of C3D. We feed all frames of the input video to VGG16 model. Then we concatenate all the fc1 features to form one single descriptor. The reason we choose VGG16 model over AlexNet [29] is that it performs better than AlexNet in image classification benchmarks. A linear kernel is used in this method's OCSVM .

- **VGG16 feature+RBF OCSVM** This method is similar to *VGG16 feature+linear OCSVM*, except it uses RBF kernel in OCSVM. We still use grid search to select the optimal hyperparameters of OCSVM.

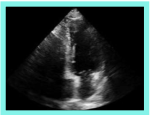
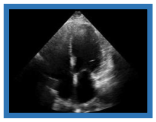
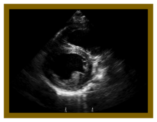
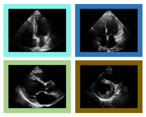
- **HOG+linear OCSVM** This method is similar to *VGG16 feature+linear OCSVM*, except it uses conventional histogram of oriented gradients (HOG) feature [11]. The cell size of HOG is 16×16 in pixels, the block size is 2×2 in cells.

3.4.2.3 Result

The mean AUROC values of all methods on impaired echocardiogram detection task are reported in Table 3.3. As can be seen in the table, the proposed method, 3D autoencoder with OCSVM layer using fine-tuning optimization obtains the highest AUROC. The proposed method using random initialization acquires second best AUROC, though the difference between them is very small. We discuss more results to compare these methods in details as follows.

Figure 3.11 shows the same results for easy comparison.

Table 3.3: Mean AUROC over all cross-validation folds for each view

Method	Mean AUROC over all cross-validation folds				
	A2C 	A4C 	PSLX 	PSSX 	**All** 
3D autoencoder from scratch	0.7039	0.6789	0.6720	0.6155	**0.6676**
3D autoencoder fine-tune	0.7077	0.6966	0.7170	0.6505	**0.6930**
3D-CNN fine-tune	0.5958	0.5725	0.6758	0.5058	**0.5875**
3D-CNN feature linear OCSVM	0.3336	0.3784	0.3435	0.4148	**0.3676**
3D-CNN feature RBF OCSVM	0.4125	0.5155	0.4879	0.5182	**0.4835**
VGG16 feature linear OCSVM	0.2020	0.2692	0.2770	0.3339	**0.2705**
VGG16 feature RBF OCSVM	0.5021	0.4581	0.5601	0.5515	**0.5180**
HOG linear OCSVM	0.4662	0.4793	0.4954	0.5192	**0.4900**

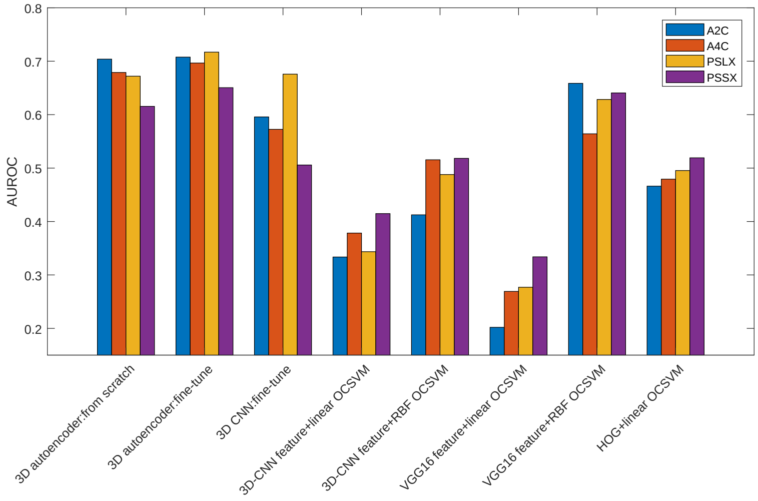


Figure 3.11: Comparison of AUROC of all methods

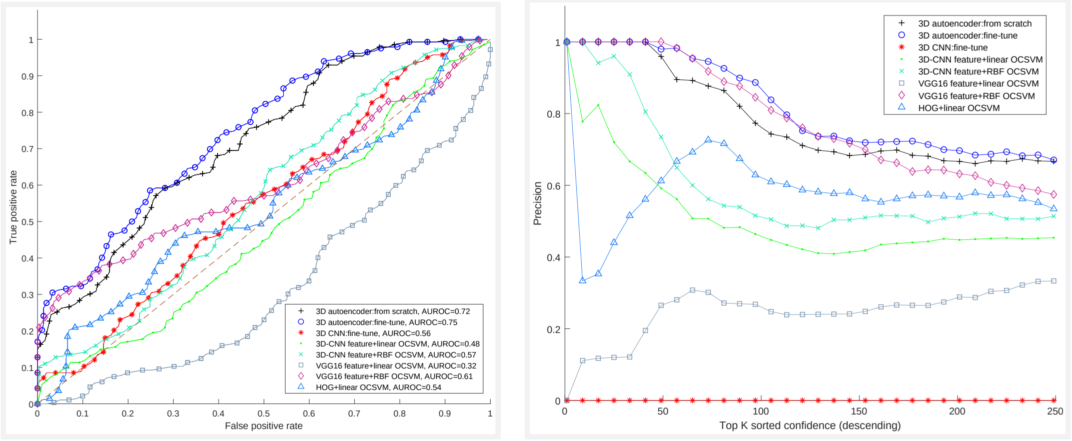
Unsupervised anomaly detection vs supervised classification. *3D-CNN:fine-tune* is the only supervised learning method in all methods. It utilizes very limited impaired echocardiogram training examples by resampling and adding various transformations. In actual training, the percentage of impaired echocardiogram is below 5%. Neural network optimization is very sensitive to data imbalance in dataset. We augment the minor class to make sure the both classes, “Normal” and “Impaired”, have approximately equal number of training examples. The mean AUROC of *3D-CNN:fine-tune* is 0.5875, which is 0.10 lower than the two proposed methods that belongs to unsupervised learning. The difference might be caused by the high variation of “impaired” echocardiogram. It also suggests that simple resampling and augmentations with transformation mitigate the data imbalance problem to some extent but it can’t recover the original data distribution since minority data is underrepresented in the dataset.

3D CNN vs CNN In general, the methods based on 3D CNN perform better than those methods using CNN. It supports that 3D CNN better models the spatial-temporal features than fusion of 2D image features.

Linear OCSVM vs RBF OCSVM Two methods both train OCSVM with linear kernel, while two other methods utilize OCSVM with RBF kernel. We observe that in both cases, the RBF kernel perform better than linear kernel. The OCSVM layer we implement in Keras framework is equivalent to a linear kernel. This observation may suggest that RBF OCSVM layer within 3D autoencoder network has the potential to improve the performance. However, to this end, we find it is challenging to implement it.

A2C vs A4C vs PSLX vs PSSX We notice that not all view of echocardiogram yield equal AUROC for impaired human heart detection. From Figure 3.11 and Table 3.3, we observe that the detection performance based on viewpoint “PSSX” is noticeably lower than other three viewpoints. It could be explained by the fact that “PSSX” is actually consist of three subcategories view points, the “PSSX Apex” level, “PSSX Mid” level and “PSSX Mitral” level. Therefore, it increases the difficulty for autoencoder to reconstruct all subsubcategories in PSSX viewpoint.

To further compare different methods, we randomly select one specific experiment, *A4C* view is selected and run on the fifth cross-validation. We plot ROC curve in Figure 3.12 (a). In this particular experiment, 3D autoencoder with OCSVM layer and fine-tuned by C3D weights, again obtains the highest AUROC, 0.7463. The second highest AUROC is 0.7174, obtained by similar method where optimization starts with random initialization.



(a) ROC curves

(b) Curves of precision vs top query

Figure 3.12: ROC curves of impaired echocardiogram detection and precision of top query curve in one random cross-validation fold

In Figure 3.12 (b), we show the curves of impaired echocardiogram detection precision with respect to the top query number for all methods, from largest to smallest. This is a measurement of the detection quality of most impaired patients. Our proposed methods with two different optimization processes still acquire the best precision in top queries.

3.5 Summary

In this chapter, we frame the task of automatic echocardiogram healthiness grading as an anomaly detection problem. We propose a 3D convolutional autoencoder with OCSVM loss. We use only healthy (normal) echocardiogram data to train a model and predict the healthiness of new echocardiogram. We evaluate the performance of the proposed method with real-world dataset which are collected in clinical settings. We also implement multiple baseline methods as comparisons. The experimental results show that our method outperforms the traditional methods. It suggests that the concept of combining 3D convolutional autoencoder and outlier detection framework could be a potential method to solve the video analysis task that has only limited yet unbalanced data set.

CHAPTER 4: Rare Event Localization and Classification

For the study of social insects like ant, interactions between individual ant are an important aspect of behavior analysis. Current approaches to the automated analysis of insect behavior from video are mainly limited to tracking the single-insect activities. In this work, we present an automated system to localize and classify fine-grained pairwise insect behaviors such as trophallaxis and grooming. Our method consists of two steps. First we generate video proposal regions by utilizing the feature map in 3D-CNN. Second, we classify the proposals into predefined ant behavior categories. Experimental results show that our approach outperforms baseline methods on testing videos from ant colonies recorded in a real laboratory setting.

4.1 Problem Statement

Many insect colonies including bee and ant, demonstrate substantial social behaviors. The study of social behaviors is important for understanding the social interaction in multi-agent system such as human society [21]. In this chapter, we aim to solve fine-grained pairwise ant localization and classification problem. A pairwise ant behavior is defined as a continuous interaction between the same two ants. In the case of ant behavior studies, videos are usually recorded on top of ant colonies for hours. The detailed analysis of all the ant behaviors often rely on the manual review of hours of video playback. This human review is costly and time-consuming.

The primary challenge for automatic identification of these ant behaviors from video is the visual similarity of the behaviors. In a typical setup, a colony contains tens to hundreds of ants. On average, each ant is roughly 80 pixels in size, and the movement of ant behaviors can be as small as a few pixels. Figure 4.1 shows a video containing multiple simultaneous ant behaviors in one colony, as well as two enlarged *grooming* and *trophollaxis* behaviors for closer comparison. As can be seen from the figure, the difference between *grooming* and *trophollaxis* is barely noticeable.

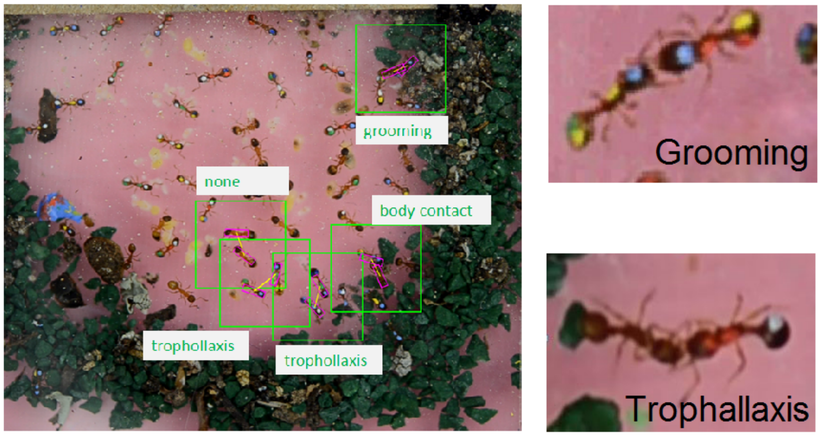


Figure 4.1: In the left of the figure, we show one frame containing several concurrent pairwise ant behaviors in the same ant colony. Two visually-similar behaviors are shown in the right, the grooming and trophollaxis.

Behavior localization in video analysis aims to know the spatial and temporal position of multiple events in a video. In the simplest case, we use a tightest 3-dimensional bounding volume to contain each event in 2D+T space. The bounding volume is represented by a vector of six elements: $(x_1, y_1, t_1, x_2, y_2, t_2)$, where (x_1, y_1) and (x_2, y_2) are the coordinates of top left corner and the bottom right corner, t_1 and t_2 are start and completion time of the event, respectively.

In the following, we review related methods in the area of automated insect analysis, describe our approach to detecting pairwise ant behaviors, and evaluate the performance of our approach on ant colony video collected from a biological research lab.

4.2 Related Work

We review the related work in two perspectives the automated insect analysis and proposal generation for video analysis.

4.2.1 Automated Insect Analysis

Existing approaches to solve this problem mainly focus on tracking of the ant's location [3, 14, 27], which track location of tens and hundreds of ants. While the above approaches have alleviated some of the burden of manual analysis, tracking is only the first step for ant behavior analysis, the ultimate goal however is to understand the insect behaviors. Recent methods directly recognize the behavior for all insects [3]. Our proposed method differs from them since we bypass the tracking of individual ants, we build a pipeline to directly localize and classify the pairwise ant behavior. Balch et al. [2] develop a method to identify common ant interactions. However, it still relies on the precomputed automated tracking location of ants.

Several other methods recognize group behaviors and non-contact behaviors. Balch and Khan [4] consider behavior of the overall group. Wittman and Gotelli [51] propose a method to model the contactless behavior of ant such as "chasing" using on Markov chain model. These two methods have different concentrations, our problem focuses on must-touch pairwise ant behaviors.

Our approach follows recent efforts toward fine-grained classification for object categorization [56] and action recognition [34] and, to the best of our knowledge, represents the first method to efficiently classify fine-grained, pairwise insect interactions from video.

4.2.2 Video Volume Proposal

Compared to region proposal generation for object localization task from images, the video proposal generation for videos behavior recognition draws less attention, partially because of its large computation time and there exists no widely used dataset for comparison. Current work to generate video proposals usually takes advantage of existing image proposal generation method. By concatenating image proposals in multiple frames we obtain video proposals. The Selective Search [48], Edge Boxes [60] are commonly used region proposal methods for object localization. Recently, convolutional neural network is modified to generate video region proposals, such as R-CNN [17], deep proposal [16] and region proposal network [38]. The common limitation of these approaches for video proposal generations is that the edges of resulting video proposals do not align with actual regions with behaviors, since the smoothness between frames is not considered. Gkioxari and Malik [18] propose a method called action tube to generate video proposal. Optical flow is used to re-rank the image proposals. However, there are two weaknesses in this method. First, it assumes there is only one action or behavior in the video, so it cannot apply to video with multiple concurrent behaviors. Second, optical flow images of all frames have to be computed separately and it is time-consuming. Our method is motivated by [16]. We extend the method of generating proposal by taking advantage of convolutional features to the case of 3D-CNN.

4.3 Method

Our task is to localize and classify the fine-grained ant behavior in video, especially for video with multiple fine-grained behaviors. The overall model including both localization and classification can be shown in Figure 4.2. Our pipeline contains two major steps: 1) generate video proposals from given testing video; and 2) classify each video proposal using a trained 3D-CNN classifier. While the second step is straightforward, we put more emphasis on describing the video proposal generation pipeline in both training stage and testing stage.

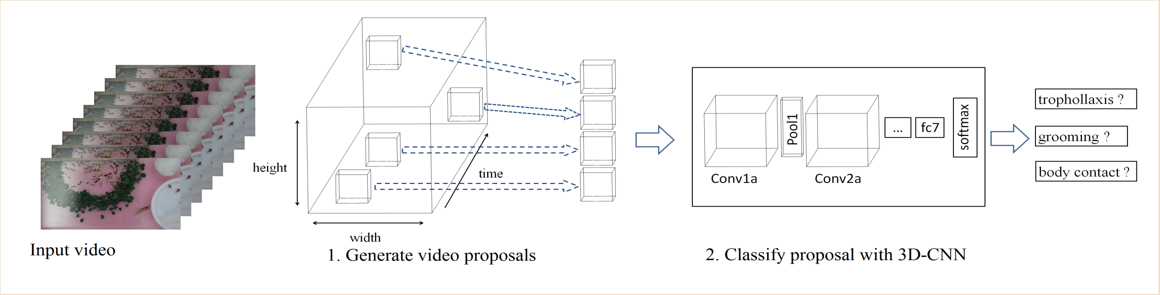


Figure 4.2: Overview of the pipeline to localize actions in video

4.3.1 Training Video Proposal Generation

Our training process consists of the training of behavior-specific 3D-CNN classifier and the training of behavior/non-behavior SVM detector. The first step is similar to the fine-tuning of the C3D model in Chapter 2, given the cropped video containing ground-truth ant behaviors. We denote this behavior-specific 3D-CNN classifier by $\mathbf{C}_{3dcnn}$, which is used to extract features at all convolutional layers in the next step.

The second step is to learn multiple binary behavior/non-behavior detectors. We first extract the convolutional features using $\mathbf{C}_{3dcnn}$ from all original training videos at all convolutional layers. Using the 3D bounding volumes of ground-truth behaviors in the original 2D+T space, we compute the corresponding 3D bounding volumes of ground-truth behaviors at all convolutional layers. Figure 4.3 depicts the training process of multiple behavior detectors based on SVM.

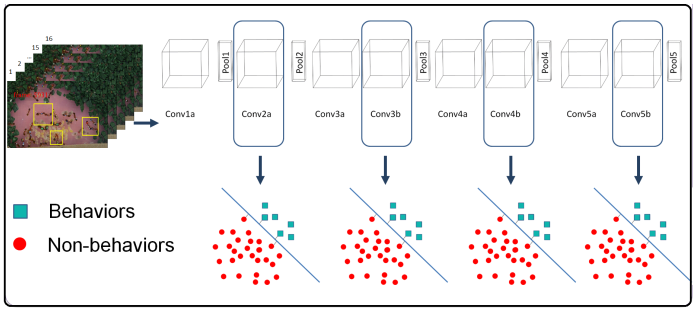


Figure 4.3: Training pipeline for 3D proposals

Next we train a set of SVM classifiers to classify behaviors and non-behaviors for all convolutional layers. The positive class of training data for SVM is the $\mathbf{C}_{3dcnn}$ features inside the ground-truth bounding volumes. Like the problem of object localization in images, negative class is not explicitly defined. We randomly sample non-behavior regions outside the ground-truth bounding volumes, as well as the region that overlaps with the ground-truth bounding volumes but the intersection over union (IoU) is lower than a predefined threshold.

Since we obtain much more non-behavior data than behavior data, i.e. the data is quite unbalanced. We use hard-negative mining [15, 33] to train SVM. A large portion of non-behavior data can be easily recognized since they contain few or no motion, thus they can be discarded after several iterations of the SVM training. The behavior/non-behavior classifiers focuses on the training samples that are hard to classify.

Hard-negative mining starts the training with all behavior data and a small portion of non-behavior data. It then tests its classification performance on the remaining non-behavior data. The falsely classified non-behavior data will be added to the training set. It repeats this steps until the training set stops changing. This also increases the training speed of SVM since the number of effective training instance is lower than all original train set in each iteration. The output of the training process is a set of behavior/non-behavior classifiers, each for one convolutional layer. A more detailed algorithmic description is shown in Algorithm 4.1 as follows.

Algorithm 4.1: Training pipeline of video proposal generation

```
Input  : $\mathbf{V}_{train} = \{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_m\}$, training video set
         $n$, number of convolutional layers in 3D-CNN
Output: $\mathbf{C}_{svm} = \{\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_n\}$, classifiers for video proposal generation
$\mathbf{U}_{train}$←get all cropped ground-truth ant behaviors using annotation;
$\mathbf{C}_{3dcnn}$←train a behavior-specific 3D-CNN with $m$ conv layers using $\mathbf{U}_{train}$;
$\mathcal{P} = \{\mathbf{P}_1, \mathbf{P}_2, \ldots, \mathbf{P}_n\} \leftarrow$ Initialize a set of set to store feature map of behaviors;
$\mathcal{N} = \{\mathbf{N}_1, \mathbf{N}_2, \ldots, \mathbf{N}_n\} \leftarrow$ Initialize a set of set to store feature map of
  non-behaviors;
for $i \leftarrow 1$ to $m$ do
    Feed $\mathbf{v}_i$ to $\mathbf{C}_{3dcnn}$;
    $\mathbf{F} = \{\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_n\} \leftarrow$ Compute convolutional features of $\mathbf{v}_i$;
    for $j \leftarrow 1$ to $n$ do
        $B \leftarrow$ Get the bounding volume of ground-truth behaviors of $\mathbf{v}_i$ at $\mathbf{f}_j$;
        $\mathbf{f}_{inside} \leftarrow$ Get $\mathbf{v}_i$'s features inside $B$ at layer $j$;
        $\mathbf{P}_j \leftarrow \mathbf{P}_j \cup \mathbf{f}_{inside}$;
        $\mathbf{f}_{outside} \leftarrow$ Randomly sample $\mathbf{v}_i$'s features outside $B$ at $\mathbf{f}_j$;
        $\mathbf{N}_j \leftarrow \mathbf{N}_j \cup \mathbf{f}_{outside}$;
    end
end
for $j \leftarrow 1$ to $n$ do
    $\mathbf{c}_j \leftarrow$ Train a binary SVM classifier to detect behaviors at conv layer $j$, using
      data $\mathbf{P}_j$ as positive and $\mathbf{N}_j$ as negative. Hard negative mining is used in the
      training;
end
return $\mathbf{C}_{svm} = \{\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_n\}$
```

4.3.2 Video Proposal Generation in Testing

We notice that size of the convolutional feature maps at different layer become smaller, while the information the convolutional feature maps contains become condenser. By analyzing feature maps from the most coarse layer (very last convolutional layer) to the most fine-grained layer (very first convolutional layer), we can localize the desired video with better precision layer by layer. In testing, the video proposal generation pipeline operates as follows. First, we feed the testing video to $\mathbf{C}_{3dcnn}$, our trained behavior-specific 3D-CNN. The convolutional feature maps of a given video are computed at all convolutional layers. Second, we utilize the rich information in 3D feature maps to reversely localize the positions and frames where ant behaviors possibly exist. The proposal generation process is illustrated by Figure 4.4.

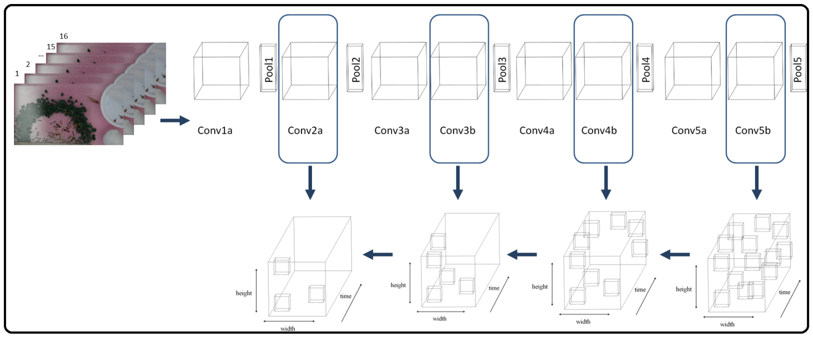


Figure 4.4: Testing pipeline with 3D proposals

As can be seen in the figure, the video proposals are generated from coarse to fine graininess, re-ranked by the response of the behavior/non-behavior classifier, $\mathbf{C}_{svm}$, in each layer. Only the sub-regions that have high response score will be preserved for the refinement of spatial-temporal position in the next step. The first convolutional layer in 3D-CNN has the highest resolution, so it localize the boundary of behaviors better than the second convolutional layer. One advantage of our method is that no optical flow is computed during both training and testing.

Algorithm 4.2: Testing pipeline of video proposal generation

```
Input  : $\mathbf{C}_{3dcnn}$, trained a behavior-specific 3D-CNN
         $\mathbf{C}_{svm} = \{\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_n\}$, trained behavior detection classifier
         $\mathbf{v}_{test}$, the testing video
         $n$, number of convolutional layers in 3D-CNN
Output: $\mathrm{B}$, a set of bounding volumes
Feed $\mathbf{v}_i$ to $\mathbf{C}_{3dcnn}$;
$\mathbf{F} = \{\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_n\} \leftarrow$ Compute convolutional features of $\mathbf{v}_{test}$;
$\mathrm{B} \leftarrow$ Get the size of $\mathbf{f}_n$;
for $j \leftarrow n$ to $i$ do
    $\mathrm{B} \leftarrow$ Use $\mathbf{c}_j$ to keep only high response volume at $\mathbf{f}_j$;
end
return $\mathrm{B}$
```

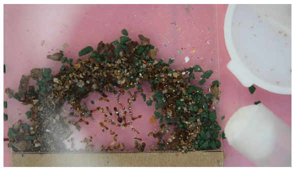
4.4 Result and Evaluation

In this section, we first describe the real ant dataset we used as well as the experiments setup. After introducing two baseline methods, we report the final comparison of performance between our methods and the baseline methods, followed by the analysis of representative testing examples.

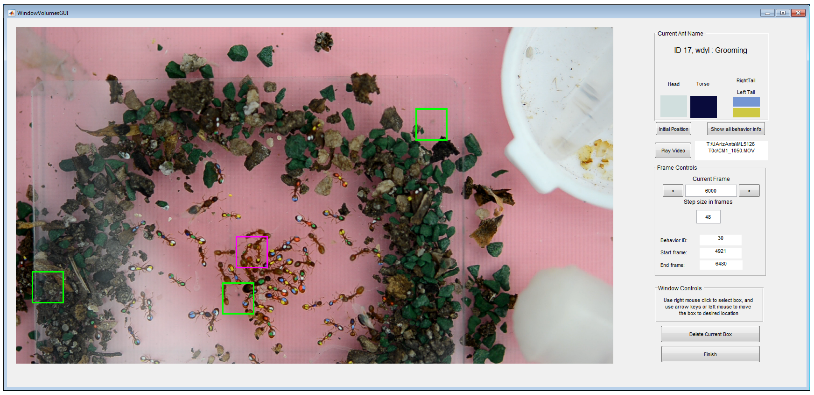
4.4.1 Data

The experimental ant behavior data set consists of 8 videos, recorded at four different ant colonies using 24 frames per second. The average length of the videos is 5 minutes, 7200 frames. The spatial resolution of the videos is 1920×1080 and there are about 50 moving ants per video on average. There are six videos in the training set, and the remaining two videos will be used for testing. Table 4.1 shows the screenshots of all videos.

Table 4.1: Snapshot of videos in the experiment

Data	Screenshot of video	
Train		
		
Test		

The positions and orientations of all ants are computed using a recent automated tracking algorithm [14]. However, the tracked positions of ants are not always accurate, partially due to the small sizes and similar appearances of ants. To make the ant position reliable, we develop a GUI application to correct and adjust the ant location frame by frame. The screenshot of the annotation tool is shown in Figure 4.5.



ID 17, wdyl : Grooming

T:\UArizAnts\ML5126 T0c\CM1_1050.MOV

Current Frame: 6000

Step size in frames: 48

Behavior ID: 30

Start frame: 4921

End frame: 6480

Use right mouse click to select box, and use arrow keys or left mouse to move the box to desired location

Figure 4.5: User interface of the ant's behavior adjustment tools we've written.

The pairwise behaviors in all videos are manually annotated and reviewed. We set the spatial window size of bounding volume based on the average size of two interacting ants. Since the average size of ant in our data set is roughly 80 pixels, we set the spatial window size to be 150x150 pixels. Three types of behaviors are considered in the experiments, **Trophallaxis**, **Grooming**, and **Other Behavior**). Table 4.2 shows that total frames of ground-truth ant behaviors in two categories.

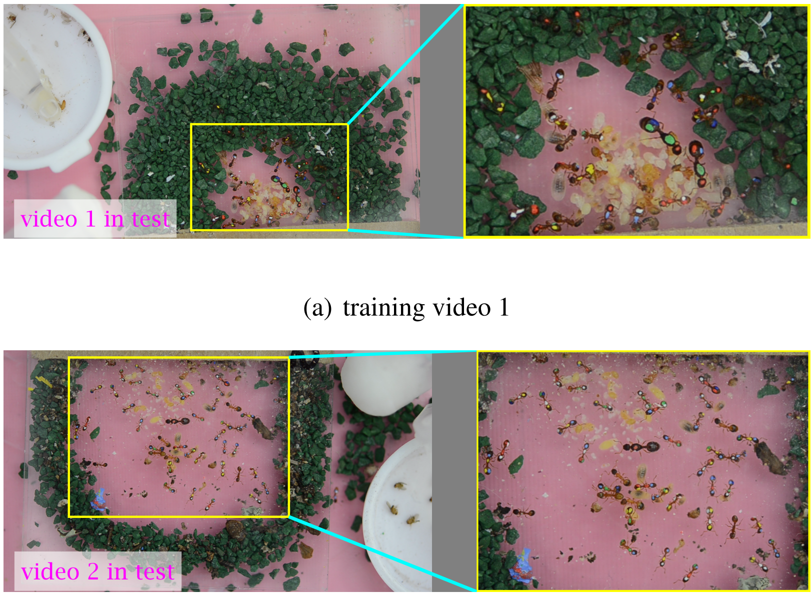
Table 4.2: Statistics of ground-truth ant behaviors

Statistics	**Trophallaxis**			**Grooming**		
	# behaviors	# frames	mean length	# behaviors	# frames	# mean length
Train	10	15850	1585.0	24	9912	413.0
Test	5	10004	2000.8	5	1958	391.6

4.4.2 Experiment Setup

A large portion of original video is the ant colony's surrounding where no ant moves. To accelerate the processing, we crop the input video, keeping the region in the center of the colony. This process does not skip any ant behaviors and can speedup the computation.

The cropping are illustrated in Figure 4.6. After this step, the resolution of the cropped videos is roughly 600×800.



(a) training video 1

(b) training video 2

Figure 4.6: The preprocess step to crop ant videos

4.4.3 Baseline Methods

We compare our proposal generation method against two baseline methods including Selective-Search [48], and Action Tubes [18].

The selective search is widely used as the primary method to generate region proposals because it is simple and fast. It repeatedly divides the input image into multiple sub-regions based on low level cues like RGB color until the region is small enough or pure enough. We apply the selective search to each frame of the ant video, and aggregate all region proposals across all frames. The intersection of region proposals in multiple frames will be viewed as the final proposal for output. The pipeline of applying selective search is shown in Figure 4.7.

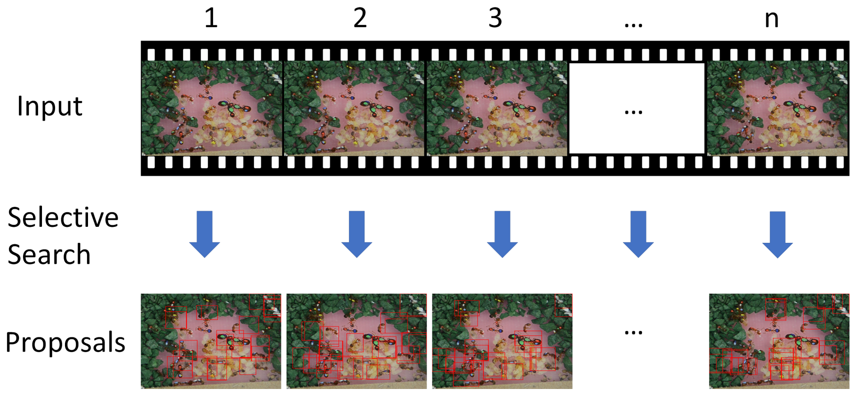


Figure 4.7: The pipeline of generating volume proposals based on selective search

The action tube method is an extension of selective search. It considers the motion smoothness in video by taking advantages of the optical flow of original image. In the proposal aggregation step, it employs dynamic programming to fuse the final output proposal. Figure 4.8 displays the overflow video proposal generation of action tube .

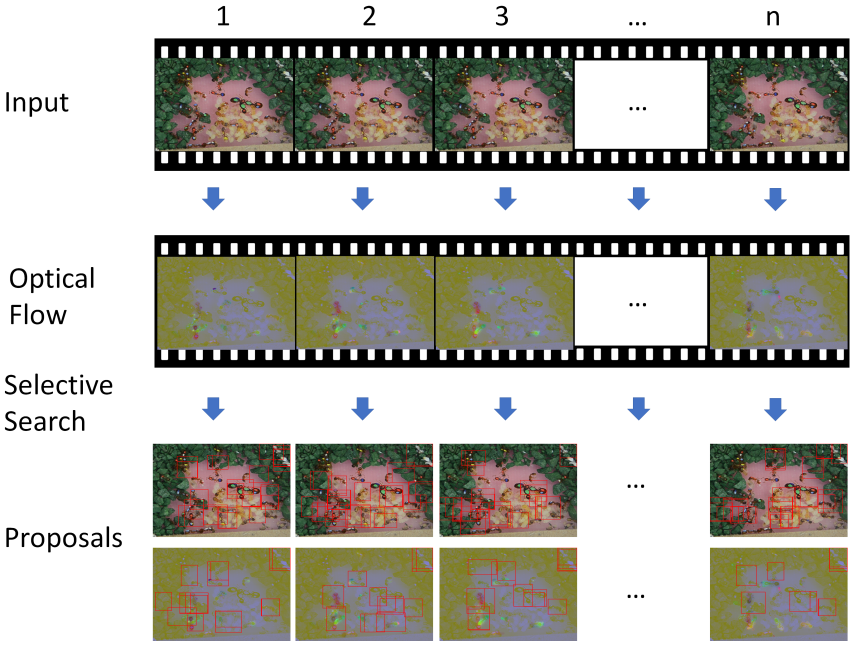


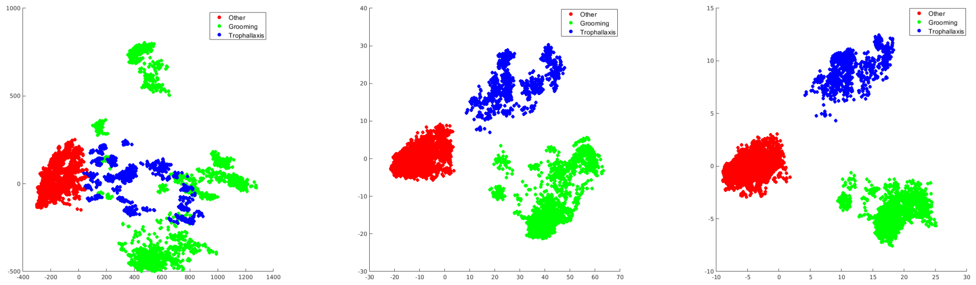
Figure 4.8: The pipeline Action Tubes of generating volume proposals

4.4.4 Result

Behavior-specific classifier As described, our pipeline to localize and classify contains two steps, the video proposal generation and video proposal classification. The behavior-specific classifier is used to classify behavior proposals in the final step of three methods. We train the behavior-specific classifier using the ground-truth data of the ant behaviors, which cropped from original videos. The behavior-specific 3D-CNN classifier is fine-tuned from the C3D model [47]. We change the number of output to 3 classes and retrain fc7 and fc8 layers in C3D using a smaller learning rate.

Since we have limited training examples and pairwise ant behaviors are highly similar, we augment the input videos to increase the training set. For each training video, the transformations we apply include, horizontal mirroring, vertical mirroring, 90 degree rotation, random white noise, brightness change. After using transformation, the training set is enlarged about 50 times.

After training, we plot the 2D MDS embedding of the training data in Figure 4.9. As shown in the figure, three classes of behaviors are well separated in fc7 and fc8 feature space. It suggests the finetuning of the model to classify different ant behaviors is effective.



(a) conv5b (b) fc7 (c) fc8

Figure 4.9: 2D MDS embedding of training data's features, extracted using finetuned 3D-CNN for behavior-specific classification. As shown in the figure, three classes of behaviors are well separated in fc7 and fc8 feature space.

4.4.4.1 Comparison of ant behavior proposal generation

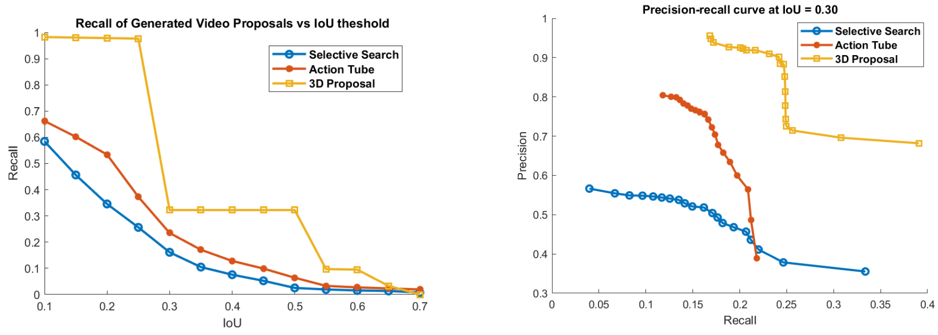
Computation Time The computation time of three proposal generation generation methods in testing is reported in Table 4.3. It can be seen that selective search is the fastest among three methods. The action tubes is the slowest because it requires separate computation of optical flow for all frames of the original video. Our proposed method is not the fastest or the slowest. Our method does not need to compute optical flow since the temporal smoothness has been captured by 3D-CNN. It is worthwhile noting that selective search runs on CPU while the other two methods rely on GPU to accelerate the CNN computation.

Table 4.3: Comparison of computation time in the experiment (in hours)

Computation time	Optical Flow	Proposal Generation	Total
Selective Search	0	0.5	0.5
Action Tubes	15	2	17
3D Proposals	0	9	9

Quality of ant behavior localization To this end, since we use the same behavior-specific classifier for all three methods, the performance of pairwise ant behavior localization is mainly determined by the quality and quantity of the region proposals. In the testing stage, one common problem is there are multiple overlapped proposals. Non-maximum suppression (NMS) is a common technique used in edge detection and object detection to remove redundant detections. We also apply NMS to keep only one candidate region with maximum classifier score and IoU score.

The number of output proposals must be fixed for fair comparison. In Figure 4.10 (a), we present the curves of recall of 1000 generated region proposals. The horizontal axis is the spatial IoU threshold of video volume proposal and the ground-truth behavior. If the actual IoU is equal to or greater than the threshold, the video proposal is considered a hit. Otherwise, the video proposal is a miss. It is not surprise that selective search is the lowest among three methods, because it does not take the motion into consideration. Overall, our proposed method obtains the highest recall and action tube gets the second highest recall. Our proposed method gets nearly 100% recall rate when IoU is less than 0.28.



(a)

(b)

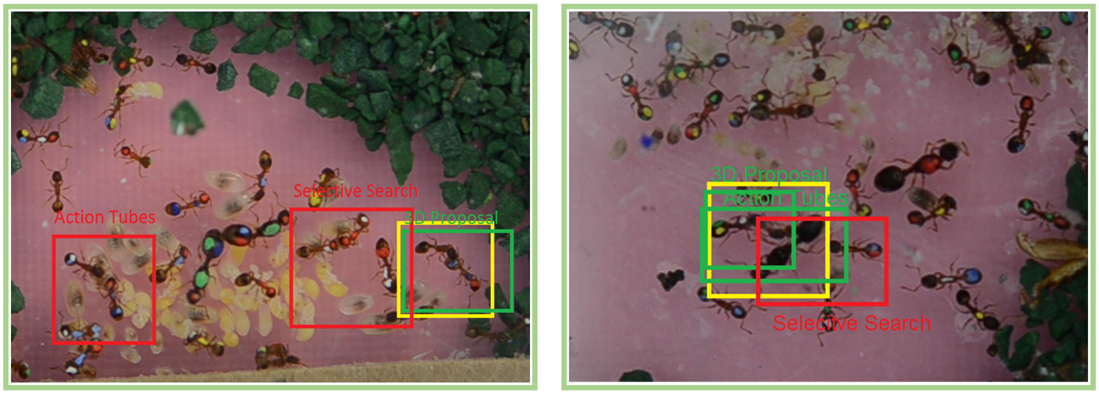
Figure 4.10: (a) Recall of generated proposals vs IoU threshold;(b) Precision-recall curve

The quality of ant behavior proposals is also reported using precision-recall curve. When fixing the IoU threshold by letting IoU = 0.3, we compare the precision of three methods, with respect to the recall rate. In Figure 4.10 (b), we show the precision-recall curves of three methods in the ant behavior localization and classification task. The horizontal axis is the recall rate. In general, our proposed method obtains the highest precision when recall=0.2. It's also clear that it obtains the largest area under the precision-recall curve. It suggests that utilizing 3D-CNN feature map helps localize better behavior proposals. In Table 4.4, we compute the the precision, recall and F-score of behavior classification using the default behavior-specific classifier.

Table 4.4: Precision, recall and F-score of detecting ant behaviors using fixed number of volume proposals (IoU=0.3)

Method	Precision	Recall	F-score
Selective Search	0.55	0.08	0.14
Action Tubes	0.76	0.16	0.26
3D Proposals	**0.89**	**0.24**	**0.38**

Error analysis We select two representative test examples to show the difference of three proposal generation methods. For selective search, it does not detect the true behavior regions. Several possible reason could explain its limitation. Firstly, selective search is tuned on general natural images, so it's not designed and not able to finetune for application-dependent problems, such as ant videos. Secondly it relies only on appearance of input image, does not consider motion information. These reasons make it less robust when adapting to a different yet challenging problem. While action tube improves the localization performance by taking into consideration motion information, it uses generic optical flow. Our proposed method is trained only with the ant-behaviors, so its ability to localize ant behavior should be more specialized. This is another advantage of our method. It can be tuned to be application-dependent video problems.



(a) Test example 1

(b) Test example 2

Figure 4.11: Two test examples of volume proposals generated by three methods. (**Yellow**) Ground-truth ant behavior. (**Green**) Correct localization. (**Red**) Wrong localization.

4.5 Summary

In this chapter, we introduce a pipeline to localize fine-grained pairwise ant behaviors from ant videos. We build a method to generate the proposals by reversely looking up the information in convolutional feature maps of all convolutional layers, from coarse grain to fine grain. The first step is to compute all the convolutional feature maps through forward propagation. The second step is to train a set of SVM classifiers at all convolutional layers in order to rank the proposal's quality. After the video volume regions are generated, we use a separate behavior-specific 3D-CNN model to classify them into the desired behavior categories. To evaluate the effectiveness of our method, we implement two baseline methods based on selective search and action tube. The experiment results show that our method yields better localization performance than two baseline methods.

CHAPTER 5: SUMMARY

Machine learning and computer vision has advanced very fast in the past years. After the revival of CNN, many challenging real-world problems existing for many years have been solved, including large-scale object classification, object localization and semantic segmentation. Some of them have achieved super-human performance in Dog Breed classification task. While the result is still debatable, it is no doubt that the computer vision algorithms building on top of CNN is quickly closing the gap between human's intelligence and the artificial intelligence in computer vision area. It is still an active research area to apply CNN-related algorithms on task with imbalanced yet small amount of data, due to the huge data hungry property of CNN. This dissertation attempts to address three problems.

In chapter 1, we briefly introduce that concepts and fundamentals that are closest to this work. We discuss the basic components of classical 2D CNN, 3D CNN.

In chapter 2, we develop a method to classify fine-grained echocardiogram view based-on 3D-CNN. This problem represents the class of tasks having small scale of data but clean and balanced labels. The proposed method employs pretrained model previously trained on large but unrelated data set. We finetune the model with limited echocardiogram data we collected. We augment the training data through many techniques. We compare the method against several other methods. The experiment result shows that the proposed method work best on echocardiogram classification. We demonstrate the effectiveness of 3D CNN on solving video analysis task with small amount of labeled training data.

In chapter 3, we propose an algorithm to detect abnormal video pattern from imbalanced data set. We utilize the model to detect impaired human heart video from a collection of echocardiograms. We build a network architecture that combines 3D convolutional autoencoder and one-class SVM. The model tasks only the majority class of data as training input and can be used to predict the likelihood or confidence score it belongs to majority class when given a new testing video. In the network, encoder aims to learn a feature representation that can be reconstructed later on with decoder. And the one-class SVM layers tries to learn a decision boundary separating most of training data with the origin in the high dimensional feature space. We also implement several baseline methods to compare with. The result shows that our method based upon autoencoder and One-Class SVM obtains the highest detection rate on average. The proof-of-concept work shows the same idea can be applied to solve similar data-imbalance problems.

In chapter 4, we introduce a method to localize ant behavior from videos based on 3D-CNN feature maps. The method takes advantages of the rich information within the 3D convolutional feature maps computed by the forward propagation. By analyzing feature maps from the most coarse layer (very last convolutional layer) to the most fine-grained layer (very first convolutional layer), we can localize the desired video gradually layer by layer. We compare our method with two previously methods and it shows that the proposed method has several advantages.

Bibliography

[1] Guha Balakrishnan, Frédo Durand, and John V. Guttag. Detecting pulse from head motions in video. In *CVPR*, pages 3430–3437, 2013.

[2] T. Balch, F. Dellaert, A. Feldman, A. Guillory, C.L. Isbell, Zia Khan, S.C. Pratt, A.N. Stein, and H. Wilde. How multirobot systems research will accelerate our understanding of social animal behavior. *Proceedings of the IEEE*, 94(7):1445–1463, July 2006.

[3] Tucker Balch, Zia Khan, and Manuela Veloso. Automatically tracking and analyzing the behavior of live insect colonies. *Proc. Conf. Autonomous Agents*, 2001.

[4] Tucker Balch, Zia Khan, and Manuela Veloso. Automatically tracking and analyzing the behavior of live insect colonies. In *Proceedings of the Fifth International Conference on Autonomous Agents*, AGENTS '01, pages 521–528, New York, NY, USA, 2001. ACM.

[5] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

[6] Xiaochun Cao and Mubarak Shah. Camera calibration and light source estimation from images with shadows. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 918–923, 2005.

[7] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM Comput. Surv.*, 41, 2009.

[8] Nitesh V. Chawla. Data mining for imbalanced datasets: An overview. In *The Data Mining and Knowledge Discovery Handbook*, 2005.

[9] François Chollet. Keras. `https://github.com/fchollet/keras`, 2015.

[10] Y. Le Cun, L. D. Jackel, B. Boser, J. S. Denker, H. P. Graf, I. Guyon, D. Henderson, R. E. Howard, and W. Hubbard. Handwritten digit recognition: applications of neural network chips and automatic learning. *IEEE Communications Magazine*, 27(11):41–46, Nov 1989.

[11] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, volume 1, pages 886–893. IEEE, 2005.

[12] Justin Domke and Yiannis Aloimonos. A probabilistic framework for correspondence and egomotion. In *ECCV Workshop on Dynamical Vision*, pages 232–242, 2006.

[13] S. Ebadollahi, Shih-Fu Chang, and H. Wu. Automatic view recognition in echocardiogram videos using parts-based representation. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, volume 2, pages 2–9, 2004.

[14] Thomas Fasciano, Hoan Nguyen, Anna Dornhaus, and Min C. Shin. Tracking multiple ants in a colony. *IEEE Winter Conference on Applications of Computer Vision*, 0:534–540, 2013.

[15] Pedro F. Felzenszwalb, Ross B. Girshick, David A. McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *TPAMI*, 32(9):1627–1645, 2010.

[16] Amir Ghodrati, Ali Diba, Marco Pedersoli, Tinne Tuytelaars, and Luc J. Van Gool. Deepproposal: Hunting objects by cascading deep convolutional layers. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, pages 2578–2586, 2015.

[17] Ross B. Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 580–587, 2014.

[18] Georgia Gkioxari and Jitendra Malik. Finding action tubes. In *CVPR*, pages 759–768, 2015.

[19] David G. Gobbi, Roch M. Comeau, and Terry M. Peters. Ultrasound probe tracking for real-time ultrasound/mri overlay and visualization of brain shift. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pages 920–927, 1999.

[20] David Hall and Pietro Perona. Fine-grained classification of pedestrians in video: Benchmark and state of the art. *CoRR*, abs/1605.06177, 2016.

[21] Bert Hlldobler and Edward Wilson. *The Superorganism: The Beauty, Elegance, and Strangeness of Insect Societies*. W.W. Norton & Company, 1 edition, 2008.

[22] Timothy M. Hospedales, Jian Li, Shaogang Gong, and Tao Xiang. Identifying rare and subtle behaviors: A weakly supervised joint topic model. *TPAMI*, 33(12):2451–2464, 2011.

[23] Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu. 3d convolutional neural networks for human action recognition. *TPAMI*, 35(1):221–231, 2013.

[24] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross B. Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the ACM International Conference on Multimedia*, pages 675–678, 2014.

[25] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Fei-Fei Li. Large-scale video classification with convolutional neural networks. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 1725–1732, 2014.

[26] Shehroz S Khan and Michael G Madden. A survey of recent trends in one class classification. In *Irish conference on Artificial Intelligence and Cognitive Science*, pages 188–197. Springer, 2009.

[27] Zia Khan, Tucker Balch, and Frank Dellaert. An mcmc-based particle filter for tracking multiple interacting targets. *Proc. Eur. Conf. Comput. Vision*, 2004.

[28] Olivier Koch and Seth J. Teller. Wide-area egomotion estimation from known 3d structure. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2007.

[29] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 1106–1114, 2012.

[30] R. Kumar, F. Wang, D. Beymer, and T. Syeda-Mahmood. Echocardiogram view classification using edge filtered scale-invariant motion features. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2009.

[31] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, Nov 1998.

[32] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015.

[33] Tomasz Malisiewicz, Abhinav Gupta, and Alexei A. Efros. Ensemble of exemplar-svms for object detection and beyond. In *ICCV*, pages 89–96, 2011.

[34] Bingbing Ni, Vignesh R. Paramathayalan, and Pierre Moulin. Multiple granularity analysis for fine-grained action detection. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 756–763, 2014.

[35] Aude Oliva and Antonio Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *International Journal of Computer Vision*, 42(3):145–175, 2001.

[36] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2010.

[37] J.H. Park, S.K. Zhou, C. Simopoulos, J. Otsuki, and D. Comaniciu. Automatic cardiac view classification of echocardiogram. In *Proc. IEEE Intl Conf. on Computer Vision*, pages 1–8, October 2007.

[38] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In C. Cortes, N. D. Lawrence,

D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 91–99. Curran Associates, Inc., 2015.

[39] Marcus Rohrbach, Sikandar Amin, Mykhaylo Andriluka, and Bernt Schiele. A database for fine grained activity detection of cooking activities. In *2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, June 16-21, 2012*, pages 1194–1201, 2012.

[40] Tomoaki Saito, Asako Kanezaki, and Tatsuya Harada. IBC127: video dataset for fine-grained bird classification. In *IEEE International Conference on Multimedia and Expo, ICME 2016, Seattle, WA, USA, July 11-15, 2016*, pages 1–6, 2016.

[41] Bernhard Schölkopf, Robert C Williamson, Alexander J Smola, John Shawe-Taylor, John C Platt, et al. Support vector method for novelty detection. In *NIPS*, volume 12, pages 582–588, 1999.

[42] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 568–576, 2014.

[43] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[44] Bharat Singh, Tim K. Marks, Michael Jones, Oncel Tuzel, and Ming Shao. A multi-stream bi-directional recurrent neural network for fine-grained action detection. In

The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2016.

[45] C. Szegedy, Wei Liu, Yangqing Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9, June 2015.

[46] David MJ Tax and Robert PW Duin. Support vector data description. *Machine learning*, 54(1):45–66, 2004.

[47] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Proc. IEEE Intl Conf. on Computer Vision*, 2015.

[48] Jasper R. R. Uijlings, Koen E. A. van de Sande, Theo Gevers, and Arnold W. M. Smeulders. Selective search for object recognition. *International Journal of Computer Vision*, 104(2):154–171, 2013.

[49] Laurens van der Maaten. Accelerating t-sne using tree-based algorithms. *Journal of Machine Learning Research*, 15(1):3221–3245, 2014.

[50] Laurens van der Maaten and Geoffrey E. Hinton. Visualizing non-metric similarities in multiple maps. *Machine Learning*, 87(1):33–55, 2012.

[51] SarahE. Wittman and NicholasJ. Gotelli. Predicting community structure of ground-foraging ant assemblages with markov models of behavioral dominance. *Oecologia*, 166(1):207–219, 2011.

[52] Hao-Yu Wu, Michael Rubinstein, Eugene Shih, John Guttag, Frédo Durand, and William T. Freeman. Eulerian video magnification for revealing subtle changes in the world. *ACM Trans. Graph. (Proceedings SIGGRAPH 2012)*, 31(4), 2012.

[53] Hui Wu, Dustin M. Bowers, Toan T. Huynh, and Richard Souvenir. Echocardiogram view classification using low-level features. In *Proc. IEEE International Symposium on Biomedical Imaging*, pages 752–755, 2013.

[54] Hui Wu, Toan T. Huynh, and Richard Souvenir. Motion factorization for echocardiogram classification. In *IEEE 11th International Symposium on Biomedical Imaging, ISBI 2014, April 29 - May 2, 2014, Beijing, Chin, Beijing, China*, pages 445–448, 2014.

[55] Matthew D. Zeiler. ADADELTA: an adaptive learning rate method. *CoRR*, abs/1212.5701, 2012.

[56] Ning Zhang, Jeff Donahue, Ross B. Girshick, and Trevor Darrell. Part-based r-cnns for fine-grained category detection. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 834–849, 2014.

[57] Zhaoxiang Zhang, Min Li, Kaiqi Huang, and Tieniu Tan. Practical camera autocalibration based on object appearance and motion for traffic scene visual surveillance. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 1–8, 2008.

[58] Zhengdong Zhang, Yasuyuki Matsushita, and Yi Ma. Camera calibration with lens distortion from low-rank textures. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 2321–2328, 2011.

[59] S.K. Zhou, J.H. Park, B. Georgescu, D. Comaniciu, C. Simopoulos, and J. Otsuki. Image-based multiclass boosting and echocardiographic view classification. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, volume 2, pages 1559–1565, 2006.

[60] C. Lawrence Zitnick and Piotr Dollár. Edge boxes: Locating object proposals from edges. In *ECCV*, 2014.