Parallel Massive Dataset Cleaning

Jianmeng Yu

4th Year Project Report

Computer Science

School of Informatics

University of Edinburgh

2018

Abstract

This project applies the decision algorithm[1] developed by Matt Pugh in 2015 on a massive parallel scale, to remove the large amount of False Positive fish detections in the Fish4Knowledge (F4K) dataset[2], without losing too many True Positives.

Also, according to Qiqi Yu's assessed runtime[3], the cleaning process will take more than 1000 days to complete on a 40-core machine. Simply putting the process onto parallel scale will not be sufficient, optimization of the code is also essential for making the processing more feasible.

This document describes the detail of various approach to reduce unnecessary work during pre-processing, improve the cleaning algorithm, and evaluating efficiency of different implementations of the machine learning techniques used. A more detailed roadmap this project is provided in the Chapter 1.

Acknowledgements

I would like to thank my project supervisor, Prof. Fisher, for his constant, patient support throughout the year. Without his expert knowledge in the field, it would be impossible for me to navigate through all of the data source and prior work of the Fish4Knowledge project.

I would also like to thank Mr. Matthew Pugh for finding out time answering my questions on the project, and precious advices on the implementation of his algorithms.

I must also extend gratitude to my friends, and my family back in China, for all their help and encouragement during my study.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Jianmeng Yu*)

Table of Contents

Chapter 1

Introduction

This project applies the previous work of Matthew Pugh[1] and Qiqi Yu[3] on massive parallel scale (detail in Chapter 2), while trying to reduce the computational cost of the cleaning algorithm. The main goal of this project is to produce a cleaned subset of a 1.6 TB dataset for future researchers.

1.1 Document Structure

Chapter 2 discussed backgrounds of the difficulties, and some of the solutions involved.

Chapter 3 described the details of the data sources, storage and preprocessing used in the cleaning algorithm.

Chapter 4 describes the first stages of the cleaning: early detection removal, feature extraction, preprocessing for classification in the next stage.

Chapter 5 discusses the final classifiers used in the cleaning, with evaluation of the results and comparison between different algorithms.

Chapter 6 contains the conclusions and possible future work needed for the project.

1.2 Fish4Knowledge Data Set

The Fish4Knowledge (F4K) project, funded by EU's Seventh Framework Programme (FP7), studies environmental effects by analysing raw videos and extract information

about observed fishes from it, so researchers could use it for studies without much programming skills.

The project acquired video data collected by Taiwan Ocean Research Institute, they set up 9 cameras in different coral reef areas such as Nanwan National Park (NPP), Lanyu, and Houbi Lake (HoBiHu) in Taiwan. After 5 years of filming, the project recorded about 524,000 10-minute video clips, with a total size of 91 TB, and approximately 1.4 billion fish detection were found in the videos, we call this the F4K Original Data Set (FDS).

In attempt to reduce the dataset, F4K project developed and applied a species recognition algorithm, which extracts all detections as 100x100 RGB images and it's description files, reducing to approximately 839 million detections. These summary files have a combined size of 1.6 TB, this is called Reduced FDS (RDS), more detailed composition of these are described in Chapter 3.

The above reduction get rids of some of the False Positives (object that are not fish, recognized as fish) from FDS, unfortunately, there are still a lot of False Positives, a classification schema is created to identify the detections.

1.3 Classification Schema

According to previous work of Matthew, 10 different classes were used to mark the training dataset, and later used in different classifiers for fitting. Fig 1.1 shows manually picked example from each of these classes.

The classes can be divided into 3 main categories:

I Not A Fish - These detection are marked for removal in future.

1 Compression Artefact - During the process of recording video, some bits were dropped during transmission of the compressed video. These detection usually have a rigid square shape.

2 Illumination Artefact - Lighting changes recognized as fishes, some are caused by refraction of turbid water, some are caused by light reflecting planktons.

3 Background Vegetation - Some of the video are captured with dynamic background, where the swaying plants are recognized as fishes.

4 Others - Everything else, this includes large floating matters, empty contours created by previous algorithms.

5 Unknown - Because of reasons like lighting and stretched video frames, it's uncertain the detection is fish or not.

II A Fish - These frames are useful for future researchers.

6 Good Boundary - With clear ocean as background, these fishes have good boundaries, and is useful for future species recognition.

7 Partial Fish - Mostly good detection boundary, but part of the fish is cut-off for various reasons.

i Fishes cut by frame boundaries.

ii Fishes are covered by vegetation or other fishes.

iii The fish is too big for the 100x100 boundary.

8 Bad Boundary - The fish is clearly captured, but the boundary extracted is useless for research.

III A Fish, but not useful - These frames detects fishes correctly, but misleading information may be extracted, it's unsure these frames should be kept or not.

9 Other Errors - like compression artefact are found in the image.

10 Multiple Fish with shared contour.

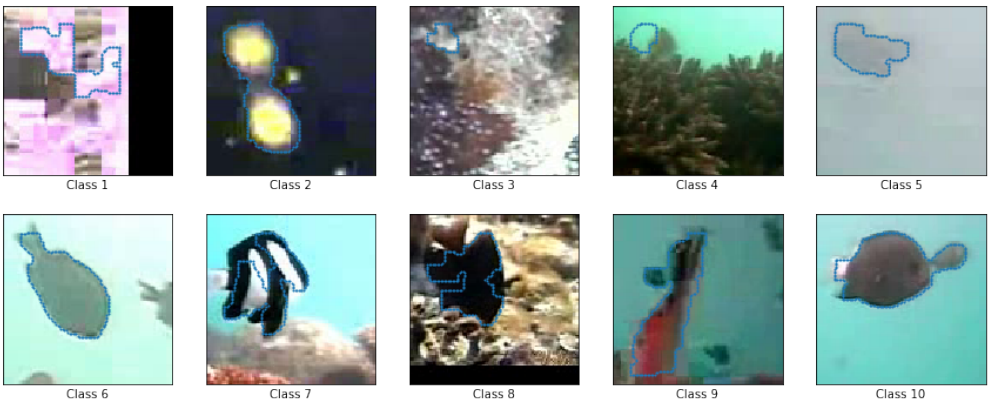

Figure 1.1: Example Detection From Each Class

1.4 Pipeline Classifier

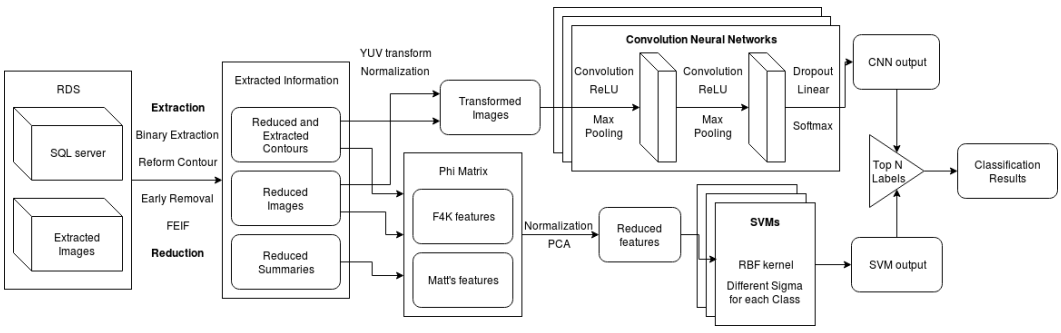

Figure 1.2: Pipeline Classifier designed by Matthew, modified

Chapter 2

Background

2.1 Big Data

chui bi

chui bi chui bi

2.2 Message Passing Interface (MPI)

Chapter 3

Data Source

from each 10 minute clip, and compiles it into a different avi file and an associated text file for description.

Under limitations of disk space and access speed, loading a large SQL database dump file into server and performing 400,000 queries is very unnecessary and time consuming, hence making it the slowest part of the cleaning. Since each record needed for the cleaning are independent, an alternative is to use python script with stdin/out pipeline to parse and partition the dump file directly into usable csv files.

3.1 Extracted Images

3.2 SQL dump file

3.2.1 Stdin based Script

Chapter 4

Preprocessing

4.1 Early Video Removal

During the feature extraction tests, it is discovered that loading a 40,000 frame video and extract features from it would take about 8 GB of memory space, if such video is processed on a node with RAM less than 8 GB, it will cause serious thrashing, rendering the node unresponsive.

While risking the chance of thrashing, these video took longer time to process, and most importantly, they are usually filled with False Positive detections. As discussed in Chapter 3.1, if a camera recorded 30,000 detection in 10 minute, means that in every frame of the original video, an average of 10 detection is extracted. By looking into these ”outlier” videos, some patterns were found:

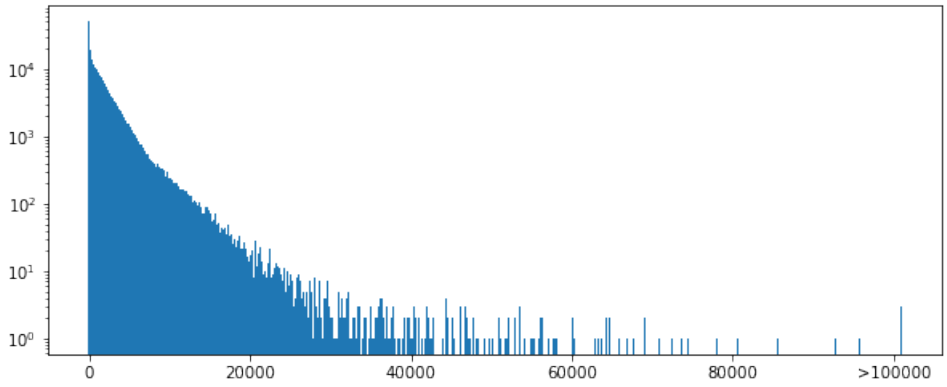

Figure 4.1: Log Scale Histogram of Detection Length

- Both Camera at Lanyu site are night-vision cameras, when they film during the night, a lot of light reflecting planktons close to the camera are recognised as fishes. Videos filmed during night have an average of 6974 detections. A 5% of the total detection comes from such videos, with the high False Positive rate, these videos can be safely excluded from cleaning.
- Videos full of compression/transmission errors, mostly happens at NPP-3 site camera 2 during June 2012 to August 2012. The camera seems to be falling down and change angles every few days. Even if there are no such errors, most of the detections are from moving background vegetation.
- One outlier video with 200,000 detection, consists of lots of repeating frames, possibly caused by previous extraction processes.

There are also some good video with high detections:

- Videos from NPP-3 site camera 3, at January 2010. These videos are captured at a higher frame rate, resulting in more detections. They usually contains lots of good detections.
- Videos filled with moving vegetation, or refraction of sunlight. They usually contains lots of good detections.

Using the above patterns, if we remove all the videos recorded in the night, videos with 40,000 or more frames, and video recorded with above characteristics and 20,000 or more frames. About 8% of the detections can be rejected without need to extract them, saving approximately 200 days of computational time.

4.2 Frame Edge Indicator Function

what the fuck man

4.3 Feature Extraction

4.3.1 F4K and Matt Feature

4.3.2 Translation Attempt

4.3.3 PCA Analysis

4.4 Image Processing For CNN

Chapter 5

Classifiers

Chapter 6

Conclusion

Bibliography

[1] Matthew Pugh. Removing false detections from a large fish image data-set. Msc dissertation, The University of Edinburgh, 2015.

[2] Fish4knowledge homepage. http://fish4knowledge.eu/. Accessed: 2018-01-08.

[3] Qiqi Yu. Adding temporal constraints to a large data cleaning problem. Msc dissertation, The University of Edinburgh, 2016.

Appendix A

Sample of Lengthy Videos

The videos below are some sample frames

Figure A.1: Sample frames from a video filmed at night.

Figure A.2: Sample frames from a corrupted video.

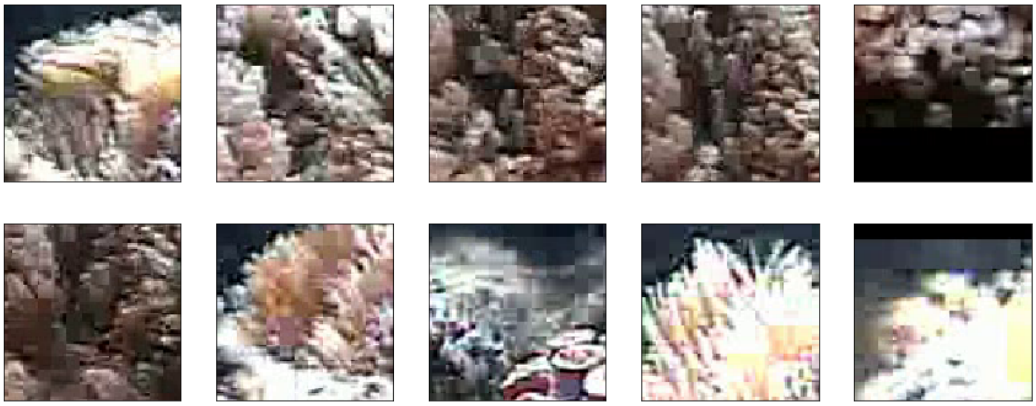

Figure A.3: Sample frames from a video facing ground.

Figure A.4: Sample frames from a video with abnormal amount of fish.