The Automation of Camera Trap Distance Sampling with Machine Learning for the Estimation of Population Density and Abundance

Author:
TOM RAYNES

DEPARTMENT OF COMPUTER SCIENCE

UNIVERSITY OF BRISTOL

A thesis submitted to the University of Bristol in accordance with the requirements of the degree of MSC COMPUTER SCIENCE

SEPTEMBER 2025

Abstract

[Abstract]

Acknowledgements

[Acknowledgements]

Author’s declaration

I declare that the work in this dissertation was carried out in accordance with the requirements of the University’s Regulations and Code of Practice for Research Degree Programmes and that it has not been submitted for any other academic award. Except where indicated by specific reference in the text, the work is the candidate’s own work. Work done in collaboration with, or with the assistance of, others, is indicated as such. Any views expressed in the dissertation are those of the author.

SIGNED: .... DATE: ....

Table of Contents

List of Tables

List of Figures

1 Introduction

Manual distance sampling bottleneck

Aim to automate distance sampling of a large dataset and use estimated distances to achieve accurate estimates for population activity

WCF dataset

How abundance and density is calculated using distances

2 Background

2.1 WCF Dataset

2.2 Methods and Models

2.2.1 Deep Learning

2.2.2 Mega Detector

2.2.3 Segment Anything

2.2.4 Depth Anything

2.2.5 Dense Prediction Transformers

2.2.6 Calibrating Distances

2.3 Estimating Activity

Distance estimation [1]

3 Experimantal

3.1 Calibration Frame Preparation

For each camera location of the dataset, a set of calibration frames (generally fifteen, location dependant) and corresponding binary masks were prepared. The frames were extracted from location-specific reference videos supplied with the dataset consisting of a person standing, facing the camera, at known distance intervals and holding a sheet of A4 paper labelled with the corresponding distance.

3.1.1 Frame Extraction

In an effort to streamline the frame extraction process, an extractor program was created (Figure 3.1). This tool enabled reference videos to be dragged and dropped into a GUI window allowing for the easy navigation between individual frames to extract the optimal one for each distance. With this software, the next/previous frames are accessed with the 'left'/'right' arrow keys, the frame index position is moved ahead/behind 20 places with the 'd'/'a' keys and the frame found at a specific timestamp is accessed with the 's' key followed by entering a time (in seconds) into an input bar. The active frame (at the current index) can be saved to disk with the 'enter' key followed by entering a filename in an input bar.

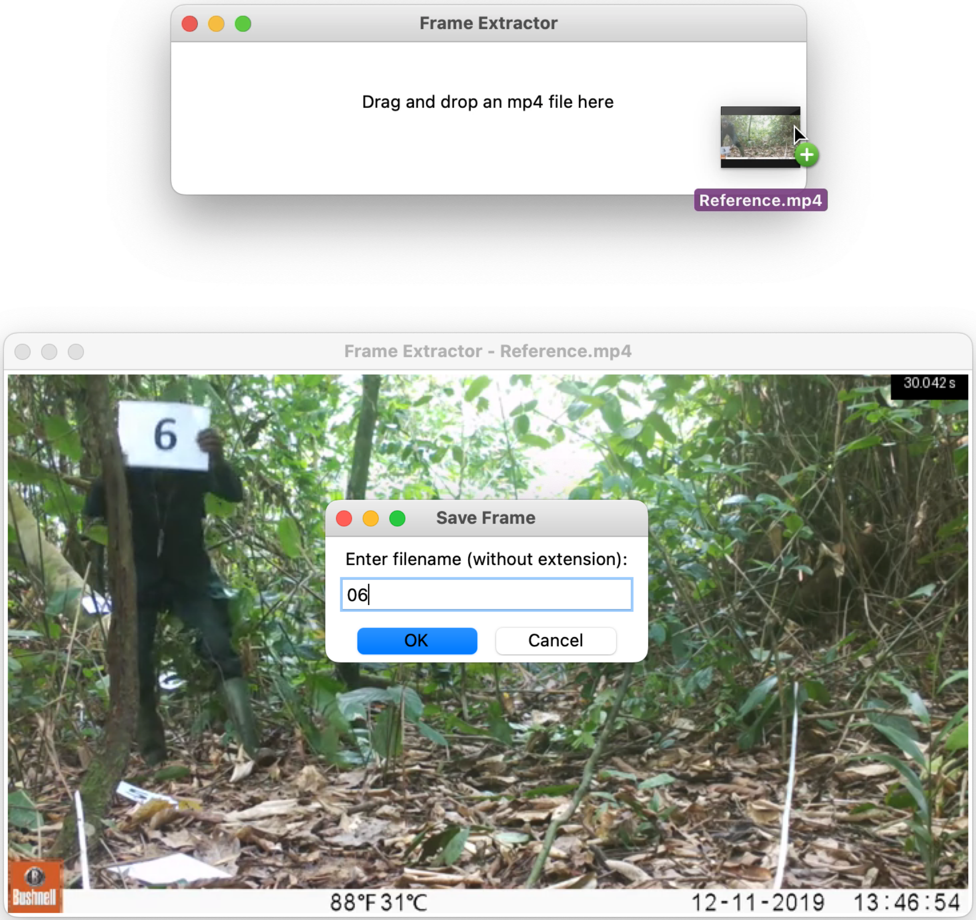

Figure 3.1: Frame extractor program showing the drag/drop (top) and save frame (bottom) features

3.1.2 Frame Mask Creation

For each of the extracted calibration frames, a corresponding binary mask was created. This task can be completed manually using image manipulation software (e.g. Photoshop, GIMP, etc...) where the segmentation boundary is manually traced then filled. This approach, however, is rather labour-intensive, constituting somewhat of a bottleneck, and therefore it is desirable to automate the process.

To assess the feasibility of automation, a two-stage detection/segmentation pipeline was created. Here, a raw calibration frame is first processed with YOLOv5 detector [2] to generate a bounding box enclosing the frame landmark. The bounding box is then passed to Segment Anything Model which predicts a segmentation mask for the landmark.

Examples of the manual and automated masks along with the corresponding raw calibration frames are shown below.

3.2 Detection Frame Extraction

3.2.1 Frame Identification

2.2.1.1 Manual Sample

Converting date/time from csv to a timestamp for each video

2.2.1.2 Automated Sample

Extracting frames from all videos sampled every 2 seconds

3.2.2 Frame Extraction

python script to extract based on timestamps

3.3 Distance Estimation

DPT/BBOX, DPT/SAM, DA/BBOX, DA/SAM

Code adjustments for blue crystal

Variable calibration runs

Depth map diagrams

3.4 Activity Estimation

Refining activity script based on distribution of estimated distances?

4 Analysis

4.1 Analysis of Distance Estimates

4.1.1 Model / Manual Distance Comparison

4.1.2 Error Analysis

4.1.3 Qualitative Analysis

close/far failure cases, sweet spot

4.1.4 Effects of Varying Calibration

4.2 Analysis of Activity Estimates

4.2.1 Manual Sample Activity Analysis

Single chimp frame distances supplemented with manual distances

4.2.2 Automated Sample Activity Analysis

5 Evaluation of Methodology

Improvement over fully manual approach?

Bottlenecks, calibration frame preparation, available compute

Time saved using frame extractor, preserves exact pixels

Limitations of the environment

Calibration and detection frame image resolution

Depth estimation point consistency (the animal itself has depth)

Error in reference videos (person has depth, not always standing up straight)

Better calibration (polynomial/not just linear, more calibration frames)

6 Conclusion

6.1 Further Work

References

[1] T. Haucke, H. S. Kühl, J. Hoyer, and V. Steinhage, "Overcoming the distance estimation bottleneck in estimating animal abundance with camera traps," *Ecological Informatics*, vol. 68, p. 101536, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1574954121003277

[2] R. Khanam and M. Hussain, "What is yolov5: A deep look into the internal features of the popular object detector," 07 2024.