Imperial College London
Department of Life Sciences

Title

Xiang Li
CID: 01606449
Email: xiang.li419@imperial.ac.uk

A thesis submitted in partial fulfilment of the requirements for the degree of
Master of Science/Research at Imperial College London
Formatted in the journal style of the Potato Journal
Submitted for the MRes/MSc in XX

Abstract

Contents

List of Tables

List of Figures

1 Introduction

Motion-sensitive cameras, also referred to as camera traps, are cameras set up strategically at field locations to sample animal populations unobtrusively. They are non-invasive, detect various species, and record many thousands of animal detection per deployment. Compared to other wildlife monitoring methods, camera traps have proven to be the most effective and cost-efficient approach for most species [Bowler et al., 2017]. They were first introduced in 1956 and has become an increasingly widespread and standardized survey tool with a 50% annual growth since 1995 when Karanth demonstrated their usefulness in population ecology [Burton et al., 2015] [Rowcliffe and Carbone, 2008] [Kays, 2016] [Gysel and Davis, 1956] [Adams, 2019].

As a primary survey tool, camera traps have been used in a range of conservation and ecological problems, including the production of species inventories and abundance estimation (Rowcliffe and Carbone 2008) [Rowcliffe and Carbone, 2008]. They are typically deployed in arrays of dozens or hundreds of sites, often resulting in a significant amount of photographs per study. For example, one two-year study resulted 2.6 million images from 98,189 detections across six states in the eastern USA [McShea et al., 2016]. While they can take millions of images, extracting knowledge from these camera trap images is traditionally down by humans, which is time-consuming, daunting, and may existing human bias [Fegraus et al., 2011] [Krishnappa and Turner, 2014] [Swinnen et al., 2014].

Meanwhile, although camera traps are often deployed to take images of wildlife, they can be equally triggered by humans. According to a 2018 University of Cambridge study, 90% percent of 235 scientists across 65 countries reported capturing some images of humans in their most recent projects [Sandbrook et al., 2018]. This kind of human data is referred to as human bycatch. In some cases, human bycatch data can outnumber animal pictures. In London hogwatch project, which aimed to study the abundance and distribution of hedgehogs in urban areas, only images taken between 6 pm to 8 am were processed to avoid human bycatch data.

Besides, the deployment of camera traps in and around the study area provides detailed and up-to-date information about the movements and activities of people using that area, which gives a great opportunity to analyze human activity patterns and improves our ability to study and conserve the ecosystem as well. As a result, camera traps are sometimes used explicitly by researchers to take images of people, in order to monitor the interaction of people and wildlife species [Pusparini et al., 2018] [Hossain et al., 2016].

Most recently, computer vision and machine learning have shown outstanding performance in image classification and image detection. In particular, image classification has been demonstrated to be dominated by convolutional neural networks (CNNs) in recent ImageNet Large Scale Visual Recognition Challenges (ILSVRC) [Krizhevsky et al., 2012] [Raleigh and Dowd, 2015]. The advances in CNNs have enabled ecologists to automate the process of identifying, counting, and describing animals and humans in camera trap images significantly [Yousif et al., 2019] [Willi et al., 2019] [Villa et al., 2017]. There is also a toolkit where CNNs are used for animal and human pose estimation in camera trap data [Graving et al., 2019].

There are several open-source libraries for human pose estimation, which are pre-trained and ready to use. These open-source software are normally trained using public datasets. The Openpose library, for example, is an open-source 2d real-time pose estimation tool trained using public human pose data set [CMU-Perceptual-Computing-Lab]. However, images taken from camera traps are rarely perfect, and many images contain humans that are too far away, too close, or only partially visible. Consequently, the performance of such pre-trained model on camera trap data is doubtful.

In this paper, to shed light on the performance of pre-trained model on camera trap data, the Openpose, an open-source 2d real-time pose estimation tool, was used to identify human images and extract human pose data from survey data taken in London hogwatch project, Hampstead Heath. The performance of Openpose as a human classifier was compared with a CNN human classifier trained using survey data. We further clustered the detected poses based on spectral theory.

2 Materials and Methods

2.1 Study area

We used images taken from the survey of Hampstead Heath, which is a part of the London Hogwatch project[Carbone and Cates, 2018]. The survey took place over a four period of four months, from April to July 2018. Two different camera traps, Reconyx and Browning Strike Force Pro, were placed as close as possible to ensure even coverage of the greenspace. There were 150 sites that covered the entirety of the Heath. The Heath Hands provided volunteers to assist with camera set-up and collection. Hampstead Heath, also simply referred to as Heath locally, is a large, ancient London Heath covering 320 hectares (790 acres). The main habitat types that can be distinguished in Hampstead Heath are: Despite corrupted images, 418,723 images from 136 sites were used in human pose estimation. Besides, we hold a sub dataset contain 57,634 images with calibrated labels. Since these images were calibrated, We could use the label to distinguish human and non-human images. We further chose 9481 human and non-human images each to form a test data set. This test data set was used as ground truth when we testify the accuracy of the Openpose classifier, which will be discussed in the following session.

The survey was originally used in monitoring the abundance and distribution of major hedgehog populations in the capital.

2.2 Openpose

Pose estimation Human estimation has primarily focused on finding body parts of individuals. Inferring the pose of multiple people in images presents a unique set of challenges. First, each image may contain an unknown number of people that can appear at any position or scale. Second, interactions between people induce complex spatial interference, due to contact, occlusion, or limb articulations, making association of parts difficult.

The traditional approach is to employ a person detector and perform a single-person pose estimation for each detection. Theses top-down approaches directly leverage existing techniques for single-person pose estimation but suffer early commitment. If the person detector fails as it is prone to do when people are in a complex environment or are partially observable, there is no resource to recovery.

The Openpose library represents the first bottom-up representation of association scores via Part Affinity Fields(PAFs), a set of 2D vector fields that encode the location and orientation of limbs over the image domain [CMU-Perceptual-Computing-Lab]. It is capable of multi-person 2D pose detection, including body, foot, hand, and facial keypoints. Compared to the other two state-of-art, well-maintained, and widely used multi-person pose estimation libraries, OpenPose has significant computational advantages. The run-time comparison among Openpose, Mask R-CNN [He et al., 2017], and Alpha-Pose [Fang et al., 2017] is illustrated in Fig. 1. .

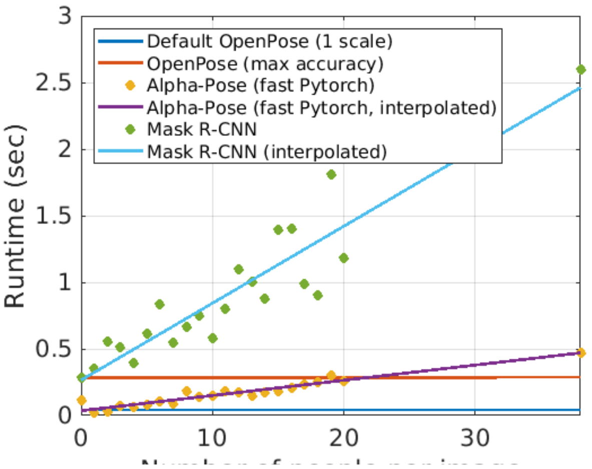


Figure 1: Inference time comparison between OpenPose, Mask R-CNN, and Alpha-Pose (fast Pytorch version).This was all performed on a system with a Nvidia 1080 Ti and CUDA 8 [Cao et al., 2018].

In this project, We used the BODY-25 model provided in the Openpose library to detect and extract human poses from the Hampstead survey data. The model was pre-trained using the combination of 2 public human pose dataset, COCO and MPI. It has the capability of extracting up to 25 human keypoints. We chose the Json file format to save the Openpose output. i.e., for each image file, the Openpose outputs a JSON file contain a "people" array of objects. Each object represents a detected human pose and has an array. containing the body parts locations and detection formatted as x1,y1,c1,x2,y2,c2,... where x and y are the coordinates, and c in (0,1) represents the confidence score.

2.3 Spectral Clustering

Graph structure has proven computationally cumbersome for pattern analysis since the correspondences must be established between the nodes of structures, which are potential of different size before they can be converted to pattern vectors. In this project, we turned to the spectral decomposition of the Laplacian matrix to overcome this problem. In this section, we illustrate how we used the spectral clustering to cluster different poeses. For each human skeleton data extracted using the Openpose library, we constructed a line graph to compute the angle between a pair of limbs. Pattern vectors were constructed from the eigenvectors of the Laplacian matrix. The pattern vectors are embedded into a pattern-space using Principal Component Analysis (PCA). And finally, we applied K-means to clustering points in the pattern-space.

The outline of this section is as follow: Section3.1 details the construction of the spectral matrix. In section 3.2, we show how symmetric polynomials can be used to construct permutation invariants from the spectral matrix elements. And In section 3.3, the complete algorithm of human pose recognition based on graph spectra is given.

2.3.1 Spectral Graph representation

Although human skeleton data extracted by the BODY-25 model contain up to 25 key points, we only used the first 15 key points for clustering. These first 15 key points represent the nose, neck, right shoulder, right elbow, right wrist, left shoulder, left elbow, left wrist, middle hip, right hip, right knee, right ankle, left knee, and left ankle. Consequently, we can use any two adjacent key points to represent a human limb. We labeled each human skeleton data with its site number and time when the image is taken for analysis. Each human skeletal structure was model as a fixed undirected skeletal graph $G = (V, E, W)$ with the vertex set $V_s = v_1, v_2 ... v_n$ corresponding to tracked body joints. The edge set consisted of undirected edges with unity weights, which were specified in W. E was decided based on knowledge about the human skeleton as follows: v_i was connected to v_j with a unity weight only if there exists a physical limb directly connecting the i-th and j-th body joint. In this way, the constructed line graph G captured the physical connectivity between body parts. Since we intended to use angular information to cluster poses, the adjacency matrix is defined to be:

$$A_{i,j} = \begin{cases} \theta_{i,j}, & \text{if } (i,j) \in E \\ 0, & \text{otherwise} \end{cases}$$

where $\theta_{i,j}$ is the angle between two adjacent limbs. The angle is calculated by:

$$\theta_{i,j} = arccos\left(\frac{\overline{v_{i2}v_{i1}} \cdot \overline{v_{j1}v_{j2}}}{|\overline{v_{i2}v_{i1}}| \cdot |\overline{v_{j1}v_{j2}}|}\right)$$

where two adjacent limbs are represented by $\overline{v_{i2}v_{i1}}$ and $\overline{v_{j1}v_{j2}}$. The nearby nodes of limbs in this case are v_{i2} and v_{j1}.

Furthermore, the Laplacian matrix of the graph was given by $L = D - A$ where D was diagonal node degree matrix whose elements were the number of edges that exit the individual nodes.

$$D_{i,i} = \sum_j A_{i,j}$$

And the normalised graph Laplacian matrix $\mathcal{L}$ is defined as:

$$\mathcal{L} \equiv D^{-\frac{1}{2}} L D^{-\frac{1}{2}} = I - D^{-\frac{1}{2}} W D^{-\frac{1}{2}}$$

The matrix L has eigenvalues which are all either positive or zero since it is positive semidefinite. In addition, eigendecomposition of $\mathcal{L}$ can be shown to be:

$$\mathcal{L} = U \Lambda U^T = \sum_{i=1}^{N} \lambda_i u_i {u_i}^T$$

where λ_i and u_i are the n eigenvalues and eigenvectors of the symmetric matrix L. The spectral matrix then has the scaled eigenvectors as columns and is given by:

$$\Psi = \left(\sqrt{\lambda_1} U_1 | \sqrt{\lambda_2} U_2 | ... | \sqrt{\lambda_n} U_n\right)$$

The matrix Ψ can be treated as a complete representation of the graph in the sense that we can use Ψ to construct the original Laplacian matrix using the relation $L = \Psi\Psi^T$.

2.3.2 Symmetric polynomials and feature vectors

Since the columns and rows of the adjacency matrix and the Laplacian matrix are indexed by the node order, they are modified by the node order. However the topology of a graph is invariant under permutations of the node orders. Thus, we intended to use the elementary symmetric polynomials to construct invariants from the elements of spectral matrix. The elementary symmetric polynomials provided spectral features which are invariant under node permutations and utilize the full spectral matrix.

A symmetric polynomial is a polynomial $P\left(x_1, x_2, ..., x_n\right)$ in n variables. A symmetric polynomial is invariant under node permutation, which means if any of the node index are interchanged, the same polynomial is obtained [Wilson et al., 2005]. The elementary symmetric polynomials (S) is a special set of symmetric polynomials that form a basis set for symmetric polynomials. Any symmetric polynomial can be expressed as a polynomial function of the elementary symmetric polynomials. For a set of variable $x_1, x_2, ...x_n$ the elementary symmetric polynomials can be defined as:

$$S_n\left(x_1, x_2, ..., x_n\right) = \prod_{i=1}^{n} x_i$$

The power symmetric polynomial functions (P) defined as:

$$P_n\left(x_1, x_2, ..., x_n\right) = \sum_{i=1}^{n} {x_i}^n$$

The elementary symmetric polynomials can be efficiently computed using the power symmetric polynomials using the Newton-Girard formula:

$$S_r = \frac{(-1)^{r+1}}{r} \sum_{k=1}^{r} (-1)^{k+r} P_r S_{r-k}$$

where the shortcut S_r is used for $S_r\left(x_1, x_2, ..., x_n\right)$ and P_r is used for $P_r\left(x_1, x_2, ..., x_n\right)$.

We intended to use the elementary symmetric polynomials to construct invariants from the elements of spectral matrix. The polynomials can provide spectral features which are invariant under node permutations of the node in a graph and utilize the full spectral matrix. These features are constructed as follows: each column of the spectral matrix Ψ forms the input to the set of spectral polynomials. For example, the column $\{\Psi_{1,i}, \Psi_{2,i}, ..., \Psi_{n,i}\}$ produces the polynomials $S_1\left(\Psi_{1,i}, \Psi_{2,i}, ..., \Psi_{n,i}\right)$, $S_2\left(\Psi_{1,i}, \Psi_{2,i}, ..., \Psi_{n,i}\right)$,..., $S_n\left(\Psi_{1,i}, \Psi_{2,i}, ..., \Psi_{n,i}\right)$. The value of each of these polynomials is invariant to the node order of the Laplacian. We can construct a set of n^2 spectral features using the n columns of the spectral matrix in combination with the n symmetric polynomials. Each set of n features for each spectral mode contains all the information about the mode up to a permutation of components. This means that it is possible to reconstruct the original components of the mode, given the values of features only.

We took only the first ten coefficients as the rest of the coefficients approach to zero because of the product terms appearing in the higher-order polynomials.

2.3.3 overall algorithm for human pose recognition based on graph spectra

Step one: For each full human skeleton data, the linear graph is constructed. The angles between adjacent limbs are calculated.

Step two: For each linear graph, the Laplacian matrix is calculated by

Step three: For each Laplacian matrix are calculated, the eigenvalues and eigenvectors are calculated

$$\mathcal{L} = U \Lambda U^T = \sum_{i=1}^{N} \lambda_i u_i {u_i}^T$$

Step four: For each Laplacian matrix, the spectral matrix on the basis of its eigenvalues and eigenvectors are calculated.

$$\Psi = \left(\sqrt{\lambda_1} U_1 | \sqrt{\lambda_2} U_2 | ... | \sqrt{\lambda_n} U_n \right)$$

Step five: For each Laplacian matrix, elementary symmetric polynomials are calculated on basis of the spectral matrix. The columns form a vector that characterized the structure of the model.

Step six: The graph feature vectors are embedded into two dimensional pattern-space by performing the principle component analysis (PCA) for visualization.

Step seven: The method of K-means is applied to clustering the points. Silhouette index, an internal clustering validation index, was used to evaluate the performance of K-means and choose the number of clusters.

3 Results

3.1 Results as human detector

The size of images varied. To ensure every images would be processed by the Openpose library and the CNN human classifier, I first compressed all the photos larger than 2.5 megabytes to 0.85 times the original size.

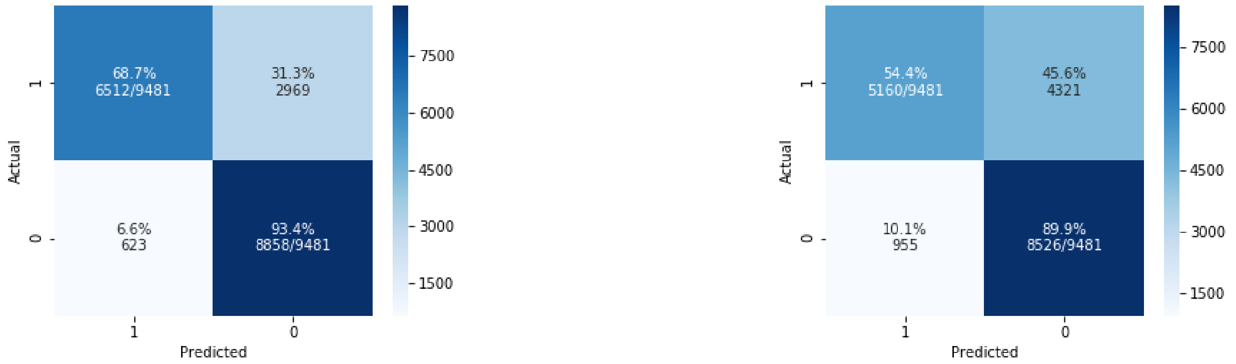
We first compared the confusion matrix for both BODY-25 model and CNN human classifier on the test data set. Different from the CNN human classifier which used 0.5 as a threshold to determine whether an image contains people or not, the BODY-25 model output a series of possible human keypoint along with their confidence score ranging from 0 to 1. To find under which number of human key points and probability we consider for getting the higher accuracy and recall, We compared the the performance of different settings. The statistics are shown in table 1

Table 1: Accuracy, precision, recall and f1 score comparison of Openpose results as human detector on test data using different threshold

probability	number of points	accuracy	precision	recall	f1 score
0.4	1	0.7104	0.7810	0.5847	0.6688
0.4	2	0.7123	0.8064	0.5588	0.6601
0.5	1	0.7217	0.8438	0.5442	0.6617
0.5	2	0.7182	0.8658	0.5164	0.6469
0.6	1	0.7209	0.8939	0.5013	0.6424
0.6	2	0.7130	0.9072	0.4745	0.6231

The BODY-25 model achieved highest accuracy when we used $p = 0.5$ and $nr = 1$ as the threshold to accept an image as human image (accuracy = 72.17%, precision = 84.38%, recall = 54.42%, f1 score = 66.17%).

The confusion matrices on test data set for CNN human classifier results and the BODY-25 model results are shown in Fig. 2. The accuracy, precision, recall and f1 score for the human classifier were 81.06%, 91.29%, 68.68% and 78.38% respectively.



(a) confusion matrix for the human classifier results

(b) confusion matrix for the BODY-25 model from the Openpose library

Figure 2: Confusion matrix on test data set. Row indicates predicted results. Column indicate ground truth. 0 indicates non-human images and 1 indicates human images

In total, We processed 418,723 images across 136 sites in total using both the BODY-25 model from the Openpose library and CNN human classifier. The time distribution of human images classified by the BODY-25 model and the human classifier is decipted in Fig. 3.

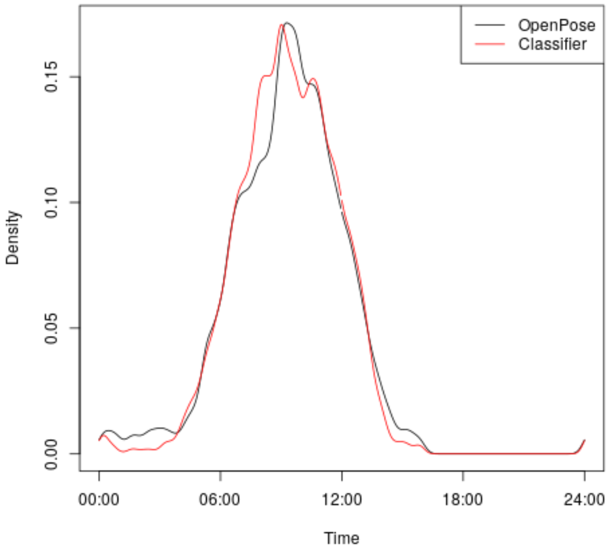


Figure 3: Density of human images against time. The black line represents density of human images classsified by the BODY-25 model provided by the Openpose, and the red line represents density of human images classified by a CNN human classifier.

Overall, there were $115,488$ images held different classification results, which was 27.6% of total images. The distribution is shown in Fig. 4.

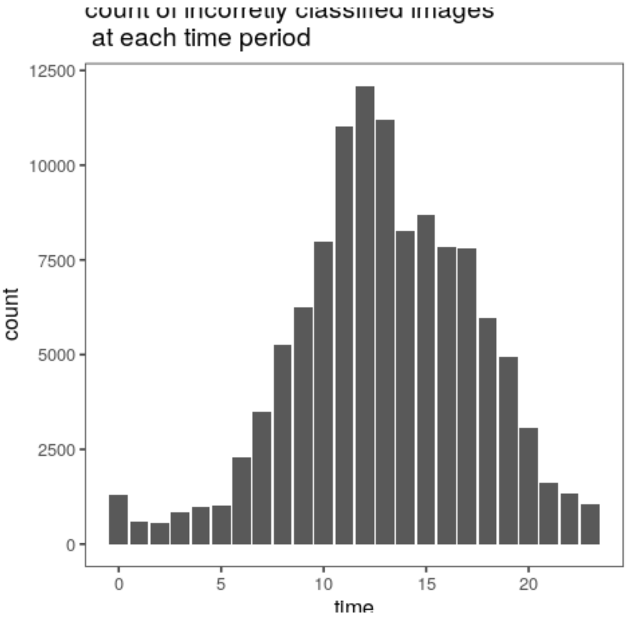


Figure 4: Distribution of images held opposite classification results from BODY-25 model and CNN human classifier.

222 The site distribution is shown below:

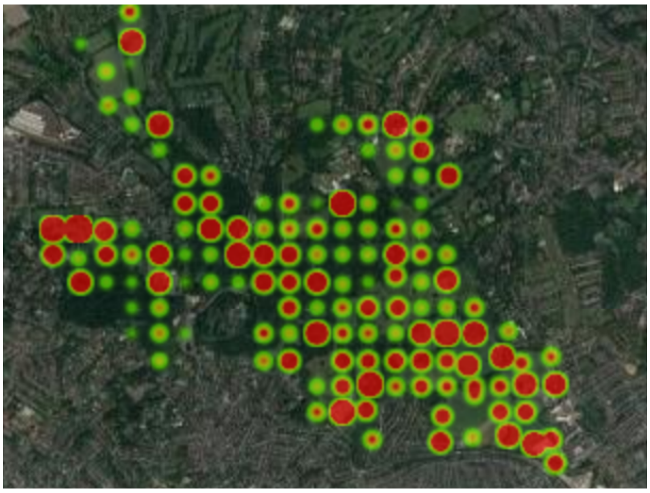


Figure 5: Distribution of human images classified using BODY-25 model over sites.

223 3.2 Spectral clustering results

224 133,568 full skeletons across 134 sites are detected. number of skeletons per images varied from 1 to 11. Distribtution illustrated as below: We intuitively use 45 seconds as unified time interval for a sequence of images. x

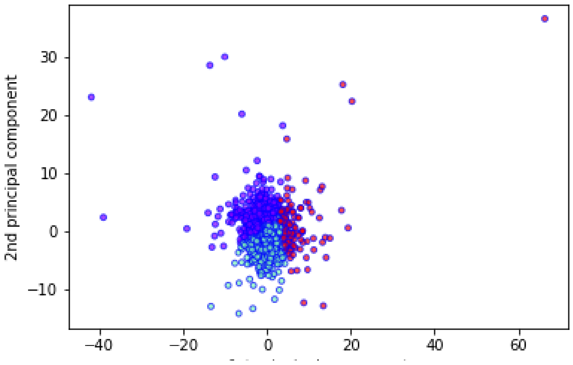


Figure 6: The data representation in 2D-dimentional space using the method of principal components.

Table 2: silhouette score for K-means on different number of clusters

number of clusters	silhouette score
2	0.6294
3	0.6441
4	0.5969
5	0.5991
6	0.5860

example image from different clusters : example of false positive and false negative results in test data set.

4 Discussion

The Openpose library, as a pre-trained human pose estimation tool, allows biologists to extract human poses from camera trap data without the need for a training set, thus reducing the workload of labeling key points of the human body. However, it has lower accuracy and recall on our test data set compared to a CNN human classifier trained using camera trap data. This may be caused because of the characteristics of camera trap images. First, the light condition varies significantly over a day. Second, the background environment of camera trap images is often cluttered. However, images in the public human pose datasets the Openpose library used to train the model, COCO and MPI, are mostly taken in the day time.

In addition to used the BODY-25 model as a human classifier, we used human poses it extracted and used K-means for clustering. The premise of our human action recognition approach was to treat every camera trap image as a still image. However, the major active research topic in computer vision and pattern recognition is motion-based human action recognition. Only recently, researchers have begun to focus on still image-based action recognition [Guo and Lai, 2014]. In traditional video-based action recognition, the low-level features extracted from space-time volume can be used directly for action recognition, e.g., the spatiotemporal interest point (STIP) based features [Laptev, 2005]. In still images, since there are no motions, we could only use high-level cues such as the human body, body parts, action-related objects, and the whole scene or context. In particular, We used angular information and separated complete human skeleton data into 3 clusters since it had the highest silhouette score (64.41%). These three clusters were not distinctly separated. This is because the person presented in a camera trap image is most likely partially occluded, at varying distances or very close to the camera. In this project, only percentage of skeleton data detected had at least 15 human keypoint and thus can be used to construct a Laplacian matrix. These fully detected human poses are likely to be detected at the same sites, with similar postures. As a result, they hold similar angular information and thus cannot be distinctly seperated in lower dimension space.

5 Conclusion

6 Code and Data Availability

To see all scripts and data used in this project as well as generated plottings, please visit github.

References

William M Adams. Geographies of conservation ii: Technology, surveillance and conservation by algorithm. *Progress in Human Geography*, 43(2):337–350, 2019.

Mark T Bowler, Mathias W Tobler, Bryan A Endress, Michael P Gilmore, and Matthew J Anderson. Estimating mammalian species richness and occupancy in tropical forest canopies with arboreal camera traps. *Remote Sensing in Ecology and Conservation*, 3(3):146–157, 2017.

A Cole Burton, Eric Neilson, Dario Moreira, Andrew Ladle, Robin Steenweg, Jason T Fisher, Erin Bayne, and Stan Boutin. Wildlife camera trapping: a review and recommendations for linking surveys to ecological processes. *Journal of Applied Ecology*, 52(3):675–685, 2015.

Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Openpose: realtime multi-person 2d pose estimation using part affinity fields. *arXiv preprint arXiv:1812.08008*, 2018.

Chris Carbone and Rachel Cates. London hogwatch hampstead heath camera-trap survey april-july 2018. 2018.

CMU-Perceptual-Computing-Lab. Cmu-perceptual-computing-lab/openpose. URL `https://github.com/CMU-Perceptual-Computing-Lab/openpose`.

Hao-Shu Fang, Shuqin Xie, Yu-Wing Tai, and Cewu Lu. Rmpe: Regional multi-person pose estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2334–2343, 2017.

Eric H Fegraus, Kai Lin, Jorge A Ahumada, Chaitan Baru, Sandeep Chandra, and Choonhan Youn. Data acquisition and management software for camera trap data: A case study from the team network. *Ecological Informatics*, 6(6):345–353, 2011.

Jacob M Graving, Daniel Chae, Hemal Naik, Liang Li, Benjamin Koger, Blair R Costelloe, and Iain D Couzin. Deepposekit, a software toolkit for fast and robust animal pose estimation using deep learning. *Elife*, 8:e47994, 2019.

Guodong Guo and Alice Lai. A survey on still image based human action recognition. *Pattern Recognition*, 47(10):3343–3361, 2014.

Leslie W Gysel and Earle M Davis. A simple automatic photographic unit for wildlife research. *The Journal of Wildlife Management*, 20(4):451–453, 1956.

Kaiming He, Georgia Gkioxari, Piotr Dollár, and B Ross. Girshick. mask r-cnn. In *ICCV*, 2017.

Abu Naser Mohsin Hossain, Adam Barlow, Christina Greenwood Barlow, Antony J Lynam, Suprio Chakma, and Tommaso Savini. Assessing the efficacy of camera trapping as a tool for increasing detection rates of wildlife crime in tropical protected areas. *Biological Conservation*, 201:314–319, 2016.

Roland Kays. *Candid creatures: how camera traps reveal the mysteries of nature*. JHU Press, 2016.

Yathin S Krishnappa and Wendy C Turner. Software for minimalistic data management in large camera trap studies. *Ecological informatics*, 24:11–16, 2014.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

Ivan Laptev. On space-time interest points. *International journal of computer vision*, 64(2-3):107–123, 2005.

William J McShea, Tavis Forrester, Robert Costello, Zhihai He, and Roland Kays. Volunteer-run cameras as distributed sensors for macrosystem mammal research. *Landscape Ecology*, 31(1):55–66, 2016.

Wulan Pusparini, Timbul Batubara, FAHRUDIN Surahmat, Tri Sugiharti, Muhammad Muslich, Fahrul Amama, William Marthy, Noviar Andayani, et al. A pathway to recovery: the critically endangered sumatran tiger panthera tigris sumatrae in an 'in danger'unesco world heritage site. *Oryx*, 52(1): 25–34, 2018.

Clionadh Raleigh and Caitriona Dowd. Armed conflict location and event data project (acled) codebook. *Find this resource*, 2015.

J Marcus Rowcliffe and Chris Carbone. Surveys using camera traps: are we looking to a brighter future? *Animal Conservation*, 11(3):185–186, 2008.

Chris Sandbrook, Rogelio Luque-Lora, and William M Adams. Human bycatch: conservation surveillance and the social implications of camera traps. *Conservation and Society*, 16(4):493–504, 2018.

Kristijn RR Swinnen, Jonas Reijniers, Matteo Breno, and Herwig Leirs. A novel method to reduce time investment when processing videos from camera trap studies. *PloS one*, 9(6):e98881, 2014.

Alexander Gomez Villa, Augusto Salazar, and Francisco Vargas. Towards automatic wild animal monitoring: Identification of animal species in camera-trap images using very deep convolutional neural networks. *Ecological informatics*, 41:24–32, 2017.

Marco Willi, Ross T Pitman, Anabelle W Cardoso, Christina Locke, Alexandra Swanson, Amy Boyer, Marten Veldthuis, and Lucy Fortson. Identifying animal species in camera trap images using deep learning and citizen science. *Methods in Ecology and Evolution*, 10(1):80–91, 2019.

Richard C Wilson, Edwin R Hancock, and Bin Luo. Pattern vectors from algebraic graph theory. *IEEE transactions on pattern analysis and machine intelligence*, 27(7):1112–1124, 2005.

Hayder Yousif, Jianhe Yuan, Roland Kays, and Zhihai He. Animal scanner: Software for classifying humans, animals, and empty frames in camera trap images. *Ecology and Evolution*, 9(4):1578–1589, 2019.