Ecological Informatics 55 (2020) 101021

Contents lists available at ScienceDirect

Ecological Informatics

journal homepage: www.elsevier.com/locate/ecolinf

Zilong: A tool to identify empty images in camera-trap data

Weideng Wei[1], Gai Luo[1], Jianghong Ran, Jing Li*

Key Laboratory of Bio-Resources and Eco-Environment, Ministry of Education, College of Life Science, Sichuan University, No.24 South Section 1, Yihuan Road, Chengdu 610065, China

ARTICLE INFO

Keywords:
Image classification
Non-machine learning algorithm
Software
Wildlife management

ABSTRACT

The use of camera traps to research and monitor wildlife results in a large number of images. Many of the images are the result of a false trigger, resulting in an empty photo. Manually removing empty images is time-intensive and costly. To increase image processing efficiency, we present a non-machine learning algorithm to identify empty images in camera-trap data, and developed freely available software, Zilong. We applied Zilong to 53,598 camera-trap images from 24 sites and compared the results to a CNN-based (Convolutional Neural Network) R package MLWIC (Machine Learning for Wildlife Image Classification). Zilong correctly identified 87% of animal images and correctly identified 85% of empty images, while MLWIC identified 65% and 69%, respectively. Our results suggest that Zilong performed better than MLWIC on identifying empty images. Zilong performed well for most of sites (22/24), with reduced performance identifying empty images when there was vegetation swinging significantly in front of camera (2/24). By using Zilong, wildlife researchers can reduce time and resources required to review camera-trap images.

1. Introduction

The use of camera traps is a very popular non-invasive technique for studying wildlife (Burton et al., 2015), and camera trap has been proposed as an effective tool for inventorying and monitoring terrestrial vertebrates, especially for elusive species (Janečka et al., 2011; O'Brien and Kinnaird, 2008; Tan et al., 2017). In recent decades, the use of motion- and infrared-triggered camera traps has increased dramatically (Garrote et al., 2011; Niedballa et al., 2016). Such widespread applications of camera traps are generating a profusion of data, especially a large accumulation of images (Swanson et al., 2015), which are too many for researchers to manually view and classify. However, among these images, a large proportion of them are empty images (containing no animals; Willi et al., 2019). False trigger by non-animal events such as vegetation swinging significantly or fluctuation of light, is one of the main reasons causing empty images. Identifying and removing these empty images manually is a serious problem for research teams, and such burden has even constrained studies (Tabak et al., 2018). In addition, when identifying the empty images manually, staring at a screen where nothing moves would strain the eyes and diminish the observer's attention to a point where he or she may miss an event (Jumeau et al., 2017). Hence, it is an urgent priority to develop an effective method to identify empty images automatically, which will reduce the labour of research team heavily.

In previous studies, Yu et al. (2013) converted animal images to grayscale images, and then extracted SIFT (Scale Invariant Feature Transform) and cLBP (cell-structured Local Binary Pattern) features of these images to train an SVM (Support Vector Machine) model, and at last used the trained model to identify species. Figueroa et al. (2014) proposed a Pixel by Pixel method to judge the presence or absence of ocelots in camera-trap images. Price Tack et al. (2016) used Canny edge descriptor to judge the presence or absence of deer, pigs, and raccoons in camera-trap images. More recently, CNN (convolutional neural network) has been proposed as a potential method to identify empty images and animal species. Norouzzadeh et al. (2018) and Willi et al. (2019) used CNN-based methods to identify empty images and species in camera-trap images. Especially, Tabak et al. (2018) published a CNN-based R package MLWIC (Machine Learning for Wildlife Image Classification), which could be used to identify empty images. However, training machine-learning models requires large amounts of annotated images and expensive computer hardware (e.g. GPUs). Moreover, the trained models may not perform well on images which are beyond their benchmark datasets.

In this study, we proposed a non-machine learning method to identify empty images in camera-trap data, and developed a software named Zilong in C++. We also compared our software performance

* Corresponding author at: College of Life Science, Sichuan University, No.24 South Section 1, Yihuan Road, Chengdu 610065, China.
E-mail address: ljtjf@126.com (J. Li).

[1] Co-first authors.

https://doi.org/10.1016/j.ecoinf.2019.101021
Received 3 April 2019; Received in revised form 22 October 2019; Accepted 22 October 2019
Available online 31 October 2019
1574-9541/

with a recent published CNN-based R package MLWIC by using the same test datasets.

2. Material and methods

2.1. Image processing algorithm in Zilong software

Our algorithm is based on the following assumptions: the camera is fixed, and if any activity triggered the camera, a sequence of images will be recorded within a short period of time. Thus, these images share the same background, but they would be different with the movement of animals. We defined pixel value difference among these images as *colour change*. However, in foggy weather, the flowing fog may also cause *colour change*, but fog cannot change edge information of the background. Therefore, in images photographed in foggy weather, the edge information would only change with the presence of animals, and we defined it as *edge change*.

An intuitive Pixel by Pixel method has been proposed to get *colour change* information from two continuous shooting images, but it is too sensitive to get a good processing result (Figueroa et al., 2014). Let *p1* and *p2* be any two continuous shooting images, and *f* be a single-channel image that contains *colour change* from *p1* and *p2*. Here, we present an improved pixel by pixel algorithm which is implemented as the following steps:

1) Perform mean-shift (Comaniciu and Meer, 1999) on *p1* and *p2* to get *s1* and *s2*:

$$s1 = \text{meanShift}\,(p1)$$

$$s2 = \text{meanShift}\,(p2)$$

2) Get the absolute value from the difference of the three-channel pixel value between *s1* and *s2*, and generate a three-channel image *d*:

$$d^n = \text{absDiff}\,(s1^n, s2^n),\ n \in \{r, g, b\}$$

where *r*, *g*, and *b* are red, green, and blue channels of a full colour image, respectively.

3) Set a threshold on the three-channel pixels of *d* to obtain result *f*:

$$f(x,y) = \begin{cases} 255, \text{ if } d^r(x,y) > \alpha,\ d^g(x,y) > \alpha,\ d^b(x,y) > \alpha \\ 0, \text{ otherwise} \end{cases}$$

where *x* and *y* are column number and row number of a pixel, and α is threshold.

Let *e* be a single-channel image that contains *colour change* from *p1*and *p2*. We used the following steps to generate *e*:

1) Convert *p1*and *p2* into grayscale images *g1*and *g2*, respectively:

$$g1 = \text{rgb2gray}\,(p1)$$

$$g2 = \text{rgb2gray}\,(p2)$$

2) Calculate the edges of *g1* and *g2* by Sobel algorithm (Kaehler and Bradski, 2016) to get *l1* and *l2*:

$$l1 = \text{Sobel}\,(g1)$$

$$l2 = \text{Sobel}\,(g2)$$

3) Set a threshold on the absolute value from the difference of pixel values between *l1* and *l2* to get the single channel binary image *m*:

$$m(x,y) = \begin{cases} 255, \text{ if } |l1(x,y) - l2(x,y)| > \beta \\ 0, \text{ otherwise} \end{cases}$$

where *x* and *y* are column number and row number of a pixel, and β is the threshold.

4) Carry out erode operation (Coster and Chermant, 2001) on *m* to get final result *e*:

$$e = \text{erode}\,(m)$$

If number of pixels whose value is 255 in *f* is larger than user-input criteria *c1*, Zilong would consider that there is a *colour change* in *p1*and *p2*; if length of longest contour in *e* is greater than user-input criteria *c2*, Zilong would consider that there is an *edge change* in *p1*and *p2*. As for two no-fog continuous shooting images, Zilong would regard them as animal images if they show *colour change*; as for two foggy-weather continuous shooting images, Zilong would regard them as animal images if they simultaneously show both *colour change* and *edge change* (Figs. 1 and 2).

2.2. Two test datasets

The first is no-fog image dataset from the Snapshot Serengeti (SS) project, which is one of the world's largest camera-trap projects published to date (Swanson et al., 2015). Totally, 29,109 images from Season 1 of SS project were used to test. The second contains 24,489 images from 10 cameras that were placed in the Mt. Gongga, China, with the elevation ranging from 3900 m to 5100 m. This dataset contains lots of foggy weather images because of the alpine climate. Image annotations of the no-fog dataset were obtained from SS project, and annotations of foggy weather dataset were from our previous project (Luo et al., 2019). The datasets are all consist of three continuous shooting images, meaning that three consecutive photographs would be recorded, if any activity triggered the camera. Animal information of the two datasets can be found in Table S2 (supplementary materials).

2.3. Verification experiment

We used both Zilong and MLWIC (Tabak et al., 2018) to identify empty images in the two datasets. Table S1 (supplementary materials) shows all the parameters set in Zilong software. According to the image annotations, we then evaluated the performances of Zilong and MLWIC. Zilong and MLWIC are all tested on a same Ubuntu 16.04.5 LTS computer (with 56 threads and 512GB RAM).

3. Results

Based on image annotations, 76% of the no-fog dataset images are empty, and in the foggy weather dataset, 70% of images are empty. After processing these test images, Zilong retained average 87% of the

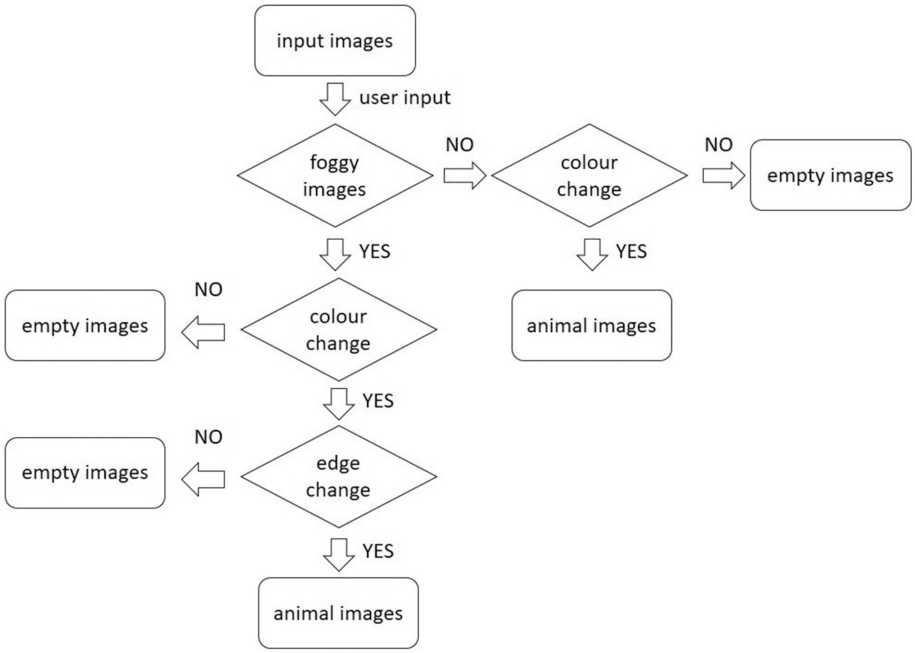

Fig. 1. Image processing flow in Zilong software.

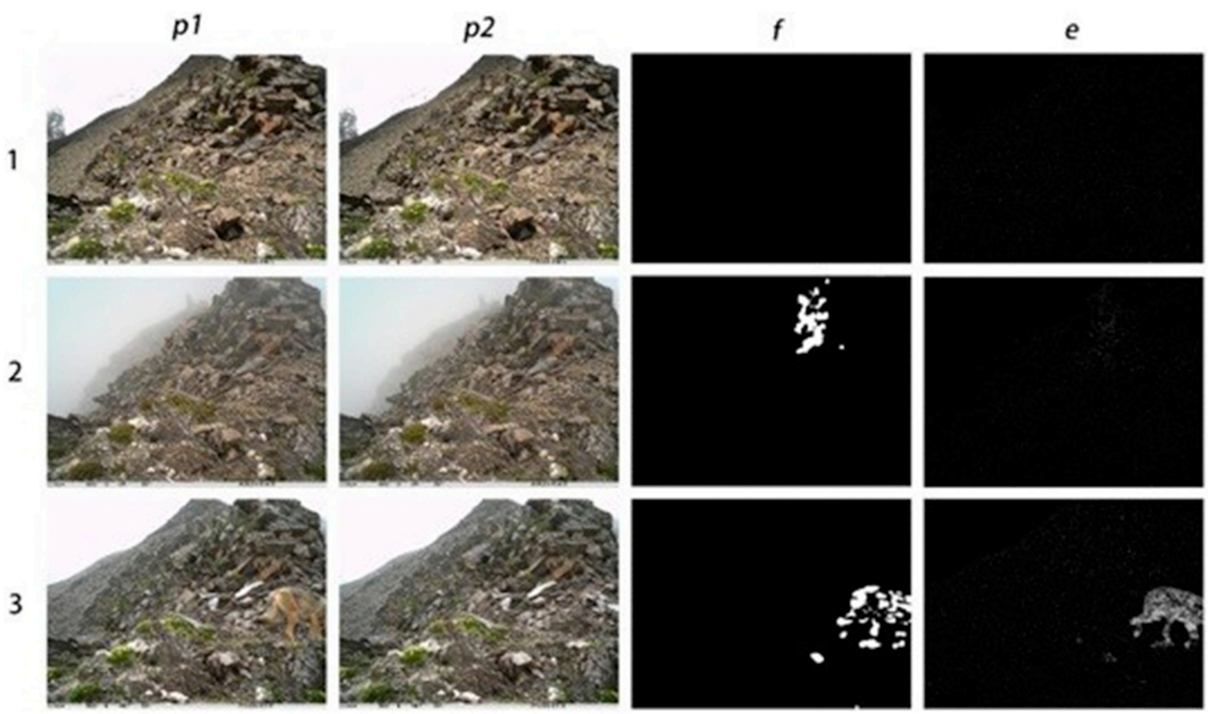

Fig. 2. Processing effect charts. (1) Empty continuous shooting images (*p1* and *p2*) with their *f* and *e* results. (2) Empty continuous shooting images in foggy weather (*p1* and *p2*) with their *f* and *e* results. (3) Continuous shooting images containing a wolf (*p1* and *p2*) with their *f* and *e* results. *f*: colour change. *e*: edge change.

animal images, and correctly identified average 85% of the empty images. However, MLWIC only got 65% and 69%, respectively (Table 1). In addition, MLWIC interrupted when processing images in S1_H02_R2, Gongga_1, Gongga_2 and Gongga_3, because these folders contain several damaged image files, while Zilong can handle this situation by skipping these images.

Further, for Zilong's results, we explored which types of animal images were incorrectly identified as empty images, and which types of

Table 1
Comparison in empty image identification performance between Zilong and MLWIC.

Image group	Foggy weather	Number of images	Number of animal images	Number of empty images	Number of animal images identified by Zilong	Number of animal images identified by MLWIC	Number of empty images identified by Zilong	Number of empty images identified by MLWIC	Zilong animal images identification accuracy rate	MLWIC animal images identification accuracy rate	Zilong empty images identification accuracy rate	MLWIC empty images identification accuracy rate	Zilong processing time (seconds)	MLWIC processing time (seconds)
S1_C04_R3	NO	618	297	321	246	200	372	418	78%	40%	95%	75%	3.6	52.3
S1_D02_R1	NO	1818	540	1278	546	328	1272	1490	96%	53%	98%	97%	10.6	48.5
S1_D03_R5	NO	2370	27	2343	1659	2329	711	41	100%	96%	30%	2%	12.1	55.0
S1_D06_R2	NO	1770	963	807	747	548	1023	1222	72%	54%	93%	96%	11.2	43.5
S1_D12_R1	NO	1221	51	1170	66	32	1155	1189	88%	47%	98%	99%	10.6	52.7
S1_F02_R4	NO	2154	141	2013	165	118	1989	2036	87%	41%	97%	97%	18.3	50.0
S1_G01_R2	NO	369	288	81	267	131	101	238	87%	39%	81%	77%	2.6	15.9
S1_H02_R2	NO	1230	942	288	903	–	327	–	89%	–	80%	–	6.9	–
S1_H10_R1	NO	5646	30	5616	39	358	5607	5288	70%	27%	99%	94%	35.1	98.5
S1_I07_R1	NO	2346	1626	720	1494	1572	852	774	86%	78%	88%	59%	13.2	47.7
S1_I13_R1	NO	1791	855	936	912	634	879	1157	88%	55%	83%	83%	10.9	34.4
S1_K03_R1	NO	1080	894	186	846	524	234	566	91%	56%	83%	91%	6.0	23.7
S1_L03_R1	NO	510	396	114	372	268	138	242	91%	55%	92%	56%	3.1	14.3
S1_T12_R1	NO	6186	48	6138	60	21	6126	6165	81%	44%	99%	100%	40.4	220.0
Gongga_1	YES	3615	3	3612	270	–	3345	–	100%	–	93%	–	155.7	–
Gongga_2	YES	2010	972	1038	672	–	1338	–	68%	–	99%	–	32.6	–
Gongga_3	YES	1785	1584	201	1530	–	255	–	94%	–	82%	–	64.3	–
Gongga_4	YES	3021	1329	1692	1566	2937	1455	84	90%	98%	78%	3%	237.7	113.5
Gongga_5	YES	5100	696	4404	4755	3429	345	1671	95%	92%	7%	37%	391.3	171.6
Gongga_6	YES	1467	747	720	738	1014	729	453	86%	94%	87%	57%	104.5	58.0
Gongga_7	YES	822	450	372	483	429	339	393	91%	74%	81%	74%	29.8	35.5
Gongga_8	YES	1803	300	1503	306	321	1497	1482	78%	72%	95%	93%	72.6	67.0
Gongga_9	YES	3231	228	3003	273	1017	2958	2214	88%	92%	98%	73%	53.0	68.1
Gongga_10	YES	1635	903	732	951	1524	684	111	95%	93%	87%	7%	63.8	60.0

A "-" represents an incomplete outcome, because an error occurred when processing these images with MLWIC.

Table 2
The manual checking result of falsely identified images by Zilong.

Image group	False positive images of Zilong result		False negative images of Zilong result	
	Unclassified animal images	Empty images	Stay still animal images	Animal images
S1_C04_R3	12(100%)	0(0%)	57(86%)	9(14%)
S1_D02_R1	18(75%)	6(25%)	9(50%)	9(50%)
S1_D03_R5	0(0%)	1632(100%)	0(0%)	0(0%)
S1_D06_R2	39(76%)	12(24%)	264(99%)	3(1%)
S1_D12_R1	3(3%)	87(97%)	0(0%)	3(100%)
S1_F02_R4	12(11%)	96(89%)	0(0%)	12(100%)
S1_G01_R2	15(100%)	0(0%)	15(42%)	21(58%)
S1_H02_R2	48(84%)	9(16%)	51(52%)	48(48%)
S1_H10_R1	12(67%)	6(33%)	3(33%)	6(67%)
S1_I07_R1	69(74%)	24(26%)	201(89%)	24(11%)
S1_I13_R1	75(49%)	78(51)	66(69%)	30(31%)
S1_K03_R1	24(80%)	6(20%)	51(65%)	27(35%)
S1_L03_R1	6(67%)	3(33%)	6(100%)	0(0%)
S1_T12_R1	6(33%)	12(67%)	6(100%)	0(0%)
Gongga_1	0(0%)	267(100%)	0(0%)	0(0%)
Gongga_2	3(33%)	6(67%)	243(79%)	66(21%)
Gongga_3	27(75%)	9(25%)	0(0%)	3(100%)
Gongga_4	336(91%)	33(9%)	84(97%)	3(3%)
Gongga_5	0(0%)	4092(100%)	15(45%)	18(55%)
Gongga_6	21(23%)	72(77%)	63(62%)	39(38%)
Gongga_7	9(13%)	60(87%)	9(23%)	30(77%)
Gongga_8	3(4%)	69(96%)	30(45%)	36(55%)
Gongga_9	9(13%)	63(87%)	6(22%)	21(78%)
Gongga_10	90(94%)	6(6%)	21(44%)	27(56%)

empty images were incorrectly identified as animal images. Referring to image annotations of the test datasets, we extracted these incorrectly identified images (Table 2). We define animal images to be positive and empty images to be negative. In false positive images of the Zilong result, we found that average 49% of these images are actually animal images. However, such images are labelled as empty in the annotation, because it is hard for researchers to identify the species of the animal (e.g. Fig. 3a). The rest are indeed no-animal images, and most of them are generated by vegetation swinging significantly in front of the cameras (e.g. Fig. 3b). In false negative images of the Zilong result, we found that 55% of them are images containing very still animals, but most of these individuals were detected by Zilong when they moved (e.g. Fig. 3c); the rest are animal images that Zilong failed to identify (e.g. Fig. 3d).

4. Discussion

In recent years, camera trap has become a common tool for wildlife research, and produced lots of empty images. Manually filtering out these empty images costs a lot of time and energy. In this study, we proposed a non-machine learning algorithm to identify empty images in camera-trap data, and developed a software named Zilong. Compared to a CNN-based R package MLWIC (Tabak et al., 2018), Zilong performed better both on identify empty images and processing time. By using Zilong, wildlife researchers can more cost-effectively process camera-trap images.

One limitation of Zilong is that it will not identify an animal if it remains completely still in multiple, consecutive images. But these individuals will be detected by Zilong when they moved in or out of camera sensing scope (e.g. Fig. 3c), and researchers can easily go back to review potential photos of still animals based on the time stamps of the images. Therefore, the limitation might not affect downstream analyses which need clear images of animals. According to previous wildlife monitoring projects (O'Brien et al., 2003; Li et al., 2010; Singh and Macdonald, 2017), a same individual photographed for multiple times within 30 min is regarded as only one animal event. Moreover, individuals usually would not keep very still for a long time. Therefore, using Zilong to processing images would retain most animal events and animal images.

Vegetation swinging significantly in front of the camera would cause false trigger of cameras and thus generate empty images (most of the images in S1_D03_R5, Gongga_5 are empty due to vegetation swinging significantly in front of cameras). This kind of false trigger would delete the memory of cameras quickly. So far, Zilong cannot correctly identify this type of empty images. Therefore, we have several recommendations on deploying camera traps:

1) Clean up the vegetation which may swing significantly in front of the camera lens, or avoid putting cameras at these places.
2) Avoid the camera lens to face the direction of the sunrise, because dramatically changing of light may also cause false trigger of camera.
3) Take high-resolution images (e.g. 4000 × 3000), because high-resolution images retain more details of objects of images, which would also help Zilong perform better.

Zilong provides a tool for wildlife researchers to reduce time and resources required for manually reviewing camera-trap images, and allows them to focus their research questions. Zilong is a freely available command line tool to identify empty images in camera-trap data, and it has no run-time dependency. It can run easily both on Windows or Linux, and can be executed in R (R Core Team, 2018). Moreover, Zilong processing result may not affect downstream analyses, because it would retain most animal events and animal images. While Zilong cannot identify animal species, it is very suitable for filtering out empty images before using deep learning model to classify animal species (because empty images will significantly affect the trained model's performance). We believe that Zilong provides wildlife researchers a cost-effective means to review images without the need for computationally and data intensive machine-learning approaches.

5. Software availability

Zilong software is freely available at https://github.com/0106WeiWeiDeng/Zilong under BSD License. A detailed documentation of Zilong is provided both in Github website and supplementary materials.

Supplementary data to this article can be found online at https://doi.org/10.1016/j.ecoinf.2019.101021.

Acknowledgements

We thank the editors of ecological informatics and the anonymous

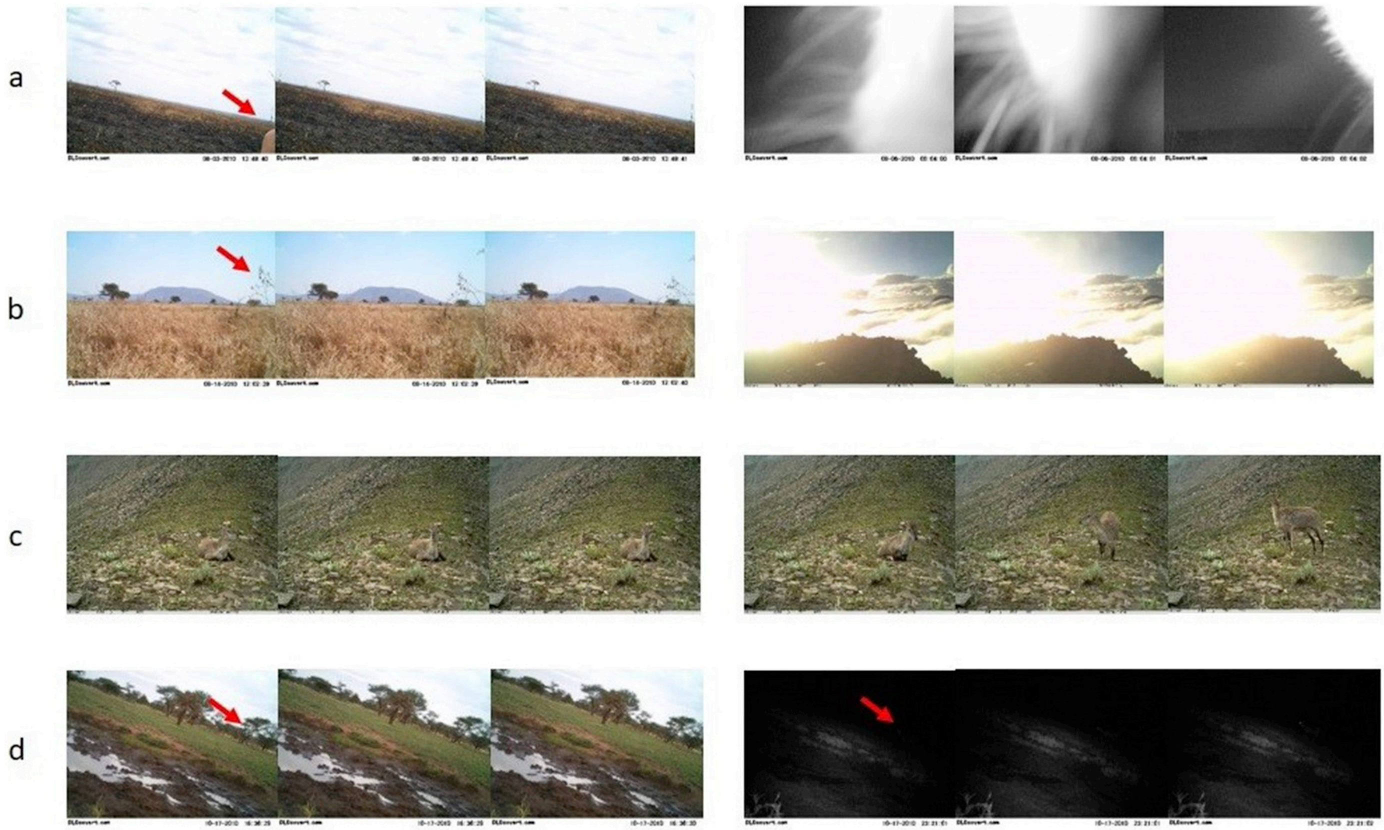

Fig. 3. Examples of falsely identified images. (a) Unclassified animal images at daytime and night. (b) Empty images caused by vegetation swinging, and light change significantly. (c) A still animal (Zilong treats these images as empty), and moved at last in a sequence of images (Zilong detected the individual). (d) Images that Zilong failed to identify: a giraffe far away from camera at daytime, and a deer far away from camera at night.

referees for their helpful comments and suggestions. This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

References

Burton, A.C., Neilson, E., Moreira, D., Ladle, A., Steenweg, R., Fisher, J.T., Bayne, E., Boutin, S., 2015. Wildlife camera trapping: a review and recommendations for linking surveys to ecological processes. J. Appl. Ecol. 52 (3), 675–685.
Comaniciu, D., Meer, P., 1999. Mean shift analysis and applications. IEEE Int. Conf. on Computer Vision. 2, 1197–1203.
Coster, M., Chermant, J.L., 2001. Image analysis and mathematical morphology for civil engineering materials. Cem. Concr. Compos. 23 (2–3), 133–151.
Figueroa, K., Camarena-Ibarrola, A., García, J., Villela, H.T., 2014. Fast automatic detection of wildlife in images from trap cameras. In: Bayro-Corrochano, E., Hancock, E. (Eds.), Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications. Springer, Cham, pp. 940–947.
Garrote, G., Perez de Ayala, R., Pereira, P., Robles, F., Guzman, N., García, F.J., Iglesias, M.C., Hervás, J., Fajardo, I., Simón, M., Barroso, J.L., 2011. Estimation of the Iberian lynx (Lynx pardinus) population in the Doñana area, SW Spain, using capture-recapture analysis of camera-trapping data. Eur. J. Wild. Res. 57 (2), 355–362.
Janečka, J.E., Munkhtsog, B., Jackson, R.M., Mallon, D.P., Naranbaatar, G., Gerelmaa, K., Murphy, W.J., 2011. Comparison of noninvasive genetic and camera-trapping techniques for surveying snow leopards. J. Mammal. 92 (4), 771–783.
Jumeau, J., Petrod, L., Handrich, Y., 2017. A comparison of camera trap and permanent recording video camera efficiency in wildlife underpasses. Ecol. Evol. 7 (18), 7399–7407.
Kaehler, A., Bradski, G., 2016. Learning OpenCV 3: Computer Vision in C++ with the OpenCV Library. O'Reilly Media, Inc., Sevastopol.
Li, S., McShea, W.J., Wang, D., Shao, L., Shi, X., 2010. The use of infrared-triggered cameras for surveying phasianids in Sichuan Province, China. Ibis 152 (2), 299–309.
Luo, G., Yang, C., Zhou, H., Seitz, M., Wu, Y., Ran, J., 2019. Habitat use and diel activity pattern of the Tibetan Snowcock (*Tetraogallus tibetanus*): a case study using camera traps for surveying high-elevation bird species. Avian Res. 10 (1), 4.
Niedballa, J., Sollmann, R., Courtiol, A., Wilting, A., 2016. camtrapR: an R package for efficient camera trap data management. Methods Ecol. Evol. 7 (12), 1457–1462.
Norouzzadeh, M.S., Nguyen, A., Kosmala, M., Swanson, A., Palmer, M.S., Packer, C., Clune, J., 2018. Automatically identifying, counting, and describing wild animals in camera-trap images with deep learning. PNAS. 115 (25), E5716–E5725.
O'Brien, T.G., Kinnaird, M.F., 2008. A picture is worth a thousand words: the application of camera trapping to the study of birds. Bird Conserv. Int. 18 (S1), S144–S162.
O'Brien, T.G., Kinnaird, M.F., Wibisono, H.T., 2003. Crouching tigers, hidden prey: Sumatran tiger and prey populations in a tropical forest landscape. Anim. Conserv. 6 (2), 131–139.
R Core Team, 2018. R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria. https://www.R-project.org/.
Singh, P., Macdonald, D.W., 2017. Populations and activity patterns of clouded leopards and marbled cats in Dampa Tiger Reserve, India. J. Mammal. 98 (5), 1453–1462.
Swanson, A., Kosmala, M., Lintott, C., Simpson, R., Smith, A., Packer, C., 2015. Snapshot Serengeti, high-frequency annotated camera trap images of 40 mammalian species in an African savanna. Scientific data. 2.
Tabak, M.A., Norouzzadeh, M.S., Wolfson, D.W., Sweeney, S.J., Vercauteren, K.C., Snow, N.P., Halseth, J.M., Di Salvo, P.A., Lewis, J.S., White, M.D., Teton, B., 2018. Machine learning to classify animal species in camera trap images: applications in ecology. Methods Ecol. Evol. 0, 1–6.
Tack, J.L.P., West, B.S., McGowan, C.P., Ditchkoff, S.S., Reeves, S.J., Keever, A.C., Grand, J.B., 2016. AnimalFinder: a semi-automated system for animal detection in time-lapse camera trap images. Ecol. Inform. 36, 145–151.
Tan, C.K.W., Rocha, D.G., Clements, G.R., Brenes-Mora, E., Hedges, L., Kawanishi, K.,

Mohamad, S.W., Mark Rayan, D., Bolongon, G., Moore, J., Wadey, J., Campos-Arceiz, A., Macdonald, D.W., 2017. Habitat use and predicted range for the mainland clouded leopard Neofelis nebulosa in Peninsular Malaysia. Biol. Conserv. 206, 65–74.
Willi, M., Pitman, R.T., Cardoso, A.W., Locke, C., Swanson, A., Boyer, A., Veldthuis, M., Fortson, L., 2019. Identifying animal species in camera trap images using deep learning and citizen science. Methods Ecol. Evol. 10 (1), 80–91.
Yu, X., Wang, J., Kays, R., Jansen, P.A., Wang, T., Huang, T., 2013. Automated identification of animal species in camera trap images. EURASIP. J. Image Vide. 2013 (1), 52.