Data Augmentation with 3DCG Models for Nuisance Wildlife Detection using a Convolutional Neural Network

Naoya Ryo^a, Hironori Kitakaze^a, Ryo Matsumura^{a,*}

^aNational Institute of Technology, Oshima College, 1091-1 Oaza-Komatsu, Suo-Oshima-cho, Oshima-gun, Yamaguchi, 742-2193, Japan

*Corresponding Author: matumura@oshima-k.ac.jp

Abstract

In this paper, we propose a data augmentation method using 3DCG models for nuisance wildlife detection. Nuisance wildlife damage to crops has become a major problem for farmers, leading to a decline in their motivation. Therefore, there is an urgent need for countermeasures against wildlife damage. To that end, we are developing a nuisance wildlife repellent system using a convolutional neural network (CNN). Therefore, it is necessary to collect training images of nuisance wildlife. This is a very difficult task, but the method we propose can solve it easily. We obtain experimental results that show that a CNN can be trained using the images generated by our method, and our trained model has an accuracy level of 92%.

Keywords: Nuisance wildlife detection, data augmentation, convolutional neural network, VGG-16.

1. Introduction

Nuisance wildlife causing damage to crops has become a major problem for farmers. In Japan, the damage done to crops by wildlife in 2019 was valued at 1.58 billion yen⁽¹⁾. Also, the nuisance wildlife population is estimated to be increasing. For example, boars have increased by about 600,000 in the 29 years between 1989 and 2017⁽¹⁾. Damage caused by nuisance wildlife has reduced the motivation of farmers and has led to an increase in abandoned land.

One example of nuisance wildlife damage control is the use of electric fences. However, they are not safe enough, with reports of accidental electric shocks which are sometimes fatal. In addition, electric fences can only prevent one type of wildlife; in order to support multiple types of wildlife,

further capital investment is required, which increases operating costs. As a solution to this problem, we are developing a system which uses machine learning to detect nuisance wildlife and repels them with sound and light effects. The system consists of computers, cameras, audio equipment, and lighting equipment, and is safe because it does not use anything that can harm humans. In contrast to electric fences, our system can be used against multiple types of wildlife simply by increasing the number of detection target classes, so there is no need to update equipment, thus reducing operating costs. In our system, a single computer is sufficient for the detection process and the control of sound and lighting, and the number of cameras is minimal because they can monitor a wider area. The number of pieces of sound and lighting equipment can be reduced by installing them in appropriate locations, so the system is inexpensive to install.

We use a convolutional neural network (CNN) for nuisance wildlife detection. A large number of training images is required to perform highly accurate object detection using machine learning technologies such as CNNs. For domesticated animals such as cats and dogs, data sets for machine learning are publicly available, but for nuisance wildlife they do not exist. In this paper, we propose a data augmentation method using 3DCG models for nuisance wildlife detection. We show that it is possible to train a CNN using the proposed method and detect real nuisance wildlife with high accuracy.

2. Related work

Various data augmentation methods have been proposed for deep learning in previous work. Data augmentation methods can be broadly classified into two categories⁽²⁾ basic image manipulations and deep learning approaches. Tradi-

Fig. 1. VGG-16 network architecture.

tional data augmentation methods such as Geometric Transformations and Color Space Transformations, as well as other approaches such as Kernel Filters, Mixing Images, and Random Erasing, are all basic image manipulations. Sharpened and blurring are examples of Kernel Filters that are also used in image processing, and Patchshuffle⁽³⁾ has been proposed for others. Patchshuffle is a method for randomly shifting pixels in an $n \times n$ window in an image using probability parameters. Mixing Images⁽⁴⁻⁶⁾ are a method for generating a new image by combining numerous images, with Sample-Pairing⁽⁴⁾ as an example. SamplePairing randomly selects the image of label A and the image of label B from the dataset and generates a new image of label A by taking their average. That combines the two images linearly, however there is also a nonlinear mixing method⁽⁵⁾. Random Image Cropping And Patching⁽⁶⁾ is a technique for generating a new image by randomly cropping four images and integrating them into one. Random Erasing⁽⁷⁾ randomly erases some areas in an image, and a similar method is Cutout⁽⁸⁾.

Feature Space Augmentation, Adversarial Training, GAN Data Augmentation, and Neural Style Transfer are all deep learning approaches. Feature Space Augmentation^(9,10) is a technique for improving the quality of an image's feature vectors. In this method, a new feature vector is obtained by adding Gaussian noise, interpolation, and extrapolation to the context vector obtained from LSTM-AutoEncoder⁽⁹⁾. The icing on the cake⁽¹⁰⁾ extracts the feature vector representation of the data from the trained model and uses it to re-train that model. Adversarial Training⁽¹¹⁾ improves the robustness of CNNs by training them with Adversarial Examples⁽¹²⁻¹⁴⁾. However, it is not known whether robustness improvement is correlated with the accuracy improvement. GAN Data

Augmentation is a GAN-based image generation technique for training image augmentation, that has been used in facial expressions emotion recognition⁽¹⁵⁾ and medical image recognition⁽¹⁶⁾. The Neural Style Transfer method involves data augmentation through style transformation⁽¹⁷⁾. Furthermore, a data augmentation method using style transformation for simulated images has been proposed⁽¹⁸⁾.

These previous works can be categorized into two types, processing and generation, depending on the different operations on the image. Our proposed method is similar to the Mixing Images and GAN Data Augmentation because it is classified as a generation. Furthermore, it is also similar to⁽¹⁸⁾ because our method uses simulated data such as 3DCG models. However, Mixing Images is less explainable, and GAN Data Augmentation is computationally expensive for generation. Reference⁽¹⁸⁾ is also costly in selecting effective style images for data augmentation and computation. Our method requires only a small number of 3DCG models, as described below, so the cost of model creation is low, the cost of overall data augmentation is low, and in addition, we can generate images with high explainability.

3. VGG-16

In this paper, we use the VGG-16⁽¹⁹⁾ CNN model. VGG-16 consists of 13 convolutional layers and three fully-connected layers, for a total of 16 layers. Fig.1 shows the VGG-16 network architecture. The convolutional layers are denoted by "conv layer" and the fully-connected layers are denoted by "FC layer." The convolution size is set to 3×3 for all convolutional layers. The output size and depth are shown in the bottom part of the figure.

Fig. 2. Overview of the proposed method.

Fig. 3. Example of the training and test images.

In our nuisance wildlife repellent system, boar and deer are the targets of detection. Therefore, the output of the final layer, FC layer3, will have three classes: “boar,” “deer,” and “others”. We trained our VGG-16 model using transfer learning from a pre-trained model by ImageNet. Since the input image size of the pre-trained model is 224×224 , we resized the images to the above size and used them as input for our model.

4. Data Augmentation using 3DCG model

As described above, applying machine learning methods such as CNNs to object detection requires a large number of training images. However, it is difficult to manually collect a set of training images of nuisance wildlife such as boar and deer that includes appearance variations such as differences in scale and view.

In this paper, we propose a data augmentation method using 3DCG models. The proposed method generates training images by creating a composite image from 3DCG models of nuisance wildlife and real images. Fig.2 shows the

overview of the proposed method. We create 3DCG models of nuisance wildlife using Smoothie-3D⁽²⁰⁾, which is a web application that can easily create a 3DCG model from an image. This allows us to generate a large number of training images from a small number of nuisance wildlife images: we create one 3DCG model from one real nuisance wildlife image, which we can synthesize with a large number of real images to generate training images. Also, by changing the size of the 3DCG model and rotating it, we can reproduce various variations in appearance.

We create one model of a boar and three models of deer using Smoothie-3D. These models are rotated, scaled, and then synthesized with the real images. In this way, training images of boar and deer are efficiently generated. The training images for the “others” class is cropped from real images of farmsteads, fields, vegetable gardens, etc. collected by image search engines. We generated 70,000 training images for each class, for a total of 210,000 images, using the proposed method. Similarly to the training images for the others class, we use an image search engine to collect images which we then crop to generate the test images for each class: “boar,”

“deer,” and “others.” A total of 900 test images, 300 for each class, are generated.

For the experiments in this paper, we have two training image datasets for training VGG-16: one dataset with a uniform image resolution of 224×224 (single-scale images), and another dataset which is scaled down by 12px every 5000 images, from an image resolution of 224×224 to an image resolution of 68×68 (multi-scale images). Fig.3 shows an example of training and test images. The top row (a) shows the generated training images and the bottom row (b) shows the test images. Also, the rows (a) and (b) shows images of a boar in the first and second columns, deer in the third, fourth columns, and others in the fifth column.

5. Experiments

5.1 Experimental overview

In our experiment, we use the previously generated training images to train VGG-16 and evaluate its performance. As described in Section 3, we trained our model using transfer learning from a pre-trained model. In transfer learning, only the output size of the final layer is changed from the pre-trained model, and the weights of the pre-trained model are used as the initial values for the weights of the model to be trained. In the training for our model, SGD with momentum is used as the optimization algorithm. As the training parameter, we set the batch size to 16 and dropout is applied to the first and second FC layers, with a ratio of 0.5. As described in Section 3, we also prepare two training datasets, one for single-scale images and one for multi-scale images, and compare the performance of the models trained using each of them.

5.2 Experimental results and discussions

Table.1 shows confusion matrices for our experiment results. The matrix of the model trained using single-scale images and the matrix of the model trained using multi-scale images is shown in (a) and (b), respectively. In a confusion matrix, the rows show the true class and the columns show the class predicted by our trained model. The results show that the model trained using single-scale images has an average recall of 80%, an average specificity of 90%, and an accuracy level of 85%. Also, the model trained using multi-scale images has an average recall of 90%, an average specificity of 95%, and an accuracy level of 92%.

Fig.4 shows the Receiver Operating Characteristic (ROC) curves of our trained models. The blue dotted lines

show the curves of the model trained using single-scale images and the red lines show the curves of the model trained using multi-scale images. The curve for the “boar” class is shown in (a), the “deer” class is shown in (b), the “others” class is shown in (c), and the average of all of the classes is shown in (d). We calculated the Area Under the Curve (AUC) for each of the ROC curves, with the following results: the AUCs of the model trained using single-scale images are 0.93 for “boar” class, 0.83 for “deer” class, 0.82 for “others” class, and 0.89 for the average of the all classes; the AUCs of the model trained using multi-scale images are 0.94 for “boar” class, 0.97 for “deer” class, 0.99 for “others” class, and 0.97 for the average of the all classes. For all classes, the model trained using multi-scale images had higher AUC than the model trained using single-scale images. In particular, there is a difference of more than 0.15 in the respective AUCs for the “deer” class. This is due to a decrease in the number of deer misclassified as “others” or “boar.” In addition, the model trained using multi-scale images also reduced misclassification in the “boar” class. These results show that the model trained using the proposed method can detect real nuisance wildlife with high accuracy. We also found that training using multi-scale images is more accurate than training using single-scale images.

We compare the output of Grad-CAM⁽²¹⁾ for the model trained using single-scale images with its output for the model trained using multi-scale images. Grad-CAM is a method to visualize the areas of attention in the image that affected the output of the trained model as a heat map. Fig.5 shows the heat maps of our model’s Grad-CAM output for the test images. In Fig.5, (a) shows the input images, (b) shows the output of Grad-CAM for the model trained using single-scale images, and (c) shows the output of Grad-CAM for the model trained using multi-scale images. In (a), (b), and (c), respectively, the first and second columns show example images for the “boar” class, the third and fourth columns show example images for the “deer” class, and the fifth column shows an example image for the “others” class.

The model trained using single-scale images misclassified the images in the first and fifth columns as “deer,” the images in the second and fourth columns as “others,” and the image in the third column as “boar.” By contrast, the model trained using multi-scale images is able to classify them correctly. For the “boar” class of images, the model trained using single-scale images pays attention to the background around the boar. However, the model trained using multi-

Table.1. Confusion Matrices

(a) Model trained using single-scale images				(b) Model trained using multi-scale images			
	Boar	Deer	Others		Boar	Deer	Others
Boar	252	29	19	Boar	270	20	10
Deer	30	181	87	Deer	29	258	13
Others	0	13	287	Others	0	14	286

Fig. 4. ROC curves.

scale images pays attention to the boar's body. These results indicate that the model recognizes a boar using its body shape and color as cues. For the "deer" class of images, the model trained using single-scale images pays attention to the deer's torso and the background around the deer. However, the model trained using multi-scale image pays attention to the deer's limbs. This result indicates that the model recognizes a deer using the shape of its limbs as a cue. For the "others" class of images, the model trained using single-scale images pays attention to the utility pole and the area around the guardrails, while the model trained using multi-scale images pays attention to various areas. Based on the above, we

will discuss the causes of the misclassifications made by the model trained using single-scale images. The images in the first and fifth rows were misclassified as "deer" because of background clutter. The image of the deer in the third row was misclassified as "boar" because the model paid attention to the torso of the deer. Similarly to the images in the first and fifth rows, the images in the second and fourth rows were misclassified as "others" because of background clutter. These results show that training using multi-scale images makes it easier to learn nuisance wildlife features and to correctly classify nuisance wildlife.

Fig. 5. Example of the heat maps of our model's Grad-CAM output.

6. Conclusions

In this paper, we proposed a data augmentation method using 3DCG models for nuisance wildlife detection. The proposed method generates training images by create a composite image from 3DCG models of nuisance wildlife and real images. This simplifies building a dataset of training images of nuisance wildlife for which images are difficult to collect manually. In the experiments, we evaluated the accuracy of the model trained using the proposed method by calculating the AUC for the ROC curves of our trained models. The results showed that the model trained using the proposed method can detect nuisance wildlife with high accuracy. In addition, we found that the model trained using multi-scale images is more accurate than the model trained using single-scale images. In this paper, accuracy is evaluated using “false positives per window”; in future work, we will evaluate the accuracy of our model using “false positives per image.” Therefore, we will extend our model to R-CNN⁽²²⁾-based approach that combines BING (Binarized Normed Gradients)⁽²³⁾ with our trained model. With the above extensions, we

aim to enable CPU processing for edge computing and implement the model in a nuisance wildlife repellent system. Also, the 3DCG models used in this paper are simple models created using Smoothie-3D. The use of software such as Blender to create a more realistic 3DCG model is expected to further improve the detection accuracy. In addition, it is possible to change the pose of the 3DCG model using animation, which can be used to generate many different training images to improve the detection accuracy.

References

- (1) Rural Development Bureau, Ministry of Agriculture, Forestry and Fisheries : “Choju higai no genjo to taisaku 鳥獣被害の現状と対策(令和 2 年 11 月) (current status and measures for wildlife damage (November, 2020)),” <https://www.maff.go.jp/j/seisan/tyozyu/higai/attach/pdf/index-374.pdf>, 2020. (Accessed June 30, 2021.)
- (2) Shorten, C. and Khoshgoftaar, T. M. : “A survey on image data augmentation for deep learning,” *Journal of Big Data*, 6(1), pp.1-48, 2019.
- (3) Kang, G., Dong, X., Zheng, L., and Yang, Y. : “P

- atchshuffle regularization,” arXiv preprint arXiv:1707.07103, 2017.
- (4) Inoue, H. : “Data augmentation by pairing samples for images classification,” arXiv preprint arXiv:1801.02929, 2018.
 - (5) Summers, C. and Dinneen, M. J. : “Improved mixed-example data augmentation,” 2019 IEEE Winter Conference on Applications of Computer Vision (WACV), pp.1262-1270, 2019.
 - (6) Takahashi, R., Matsubara, T., and Uehara, K. : “Data augmentation using random image cropping and patching for deep CNNs,” IEEE Transactions on Circuits and Systems for Video Technology, 30(9), pp.2917-2931, 2019.
 - (7) Zhong, Z., Zheng, L., Kang, G., Li, S., and Yang, Y. : “Random Erasing Data Augmentation,” arXiv preprint arXiv:1708.04896, 2017.
 - (8) DeVries, T. and Taylor, G. W. : “Improved regularization of convolutional neural networks with cutout,” arXiv preprint arXiv:1708.04552, 2017.
 - (9) DeVries, T. and Taylor, G. W. : “Dataset augmentation in feature space,” arXiv preprint arXiv:1702.05538, 2017.
 - (10) Konno, T. and Iwazume, M. : “Icing on the cake: An easy and quick post-learnig method you can try after deep learning,” arXiv preprint arXiv:1807.06540, 2018.
 - (11) Li, S., Chen, Y., Peng, Y., and Bai, L. : “Learning more robust features with adversarial training,” arXiv preprint arXiv:1804.07757, 2018.
 - (12) Goodfellow, I. J., Shlens, J., and Szegedy, C. : “Explaining and harnessing adversarial examples,” arXiv preprint arXiv:1412.6572, 2014.
 - (13) Moosavi-Dezfooli, S. M., Fawzi, A., and Frossard, P. : “Deepfool: a simple and accurate method to fool deep neural networks,” Proceedings of the IEEE conference on computer vision and pattern recognition, pp.2574-2582, 2016.
 - (14) Su, J., Vargas, D. V., and Sakurai, K. : “One pixel attack for fooling deep neural networks,” IEEE Transactions on Evolutionary Computation, 23(5), pp.828-841, 2019.
 - (15) Zhu, X., Liu, Y., Qin, Z., and Li, J., “Data augmentation in emotion classification using generative adversarial networks,” arXiv preprint arXiv:1711.00648, 2017.
 - (16) Frid-Adar, M., Diamant, I., Klang, E., Amitai, M., Goldberger, J., and Greenspan, H. : “GAN-based synthetic medical image augmentation for increased CNN performance in liver lesion classification,” Neurocomputing, 321, pp.321-331, 2018.
 - (17) Jackson, P. T., Abarghouei, A. A., Bonner, S., Breckon, T. P., and Obara, B., “Style augmentation: data augmentation via style randomization,” CVPR Workshops, pp. 83-92, 2019.
 - (18) Tobin, J., Fong, R., Ray, A., Schneider, J., Zaremba, W., and Abbeel, P. : “Domain randomization for transferring deep neural networks from simulation to the real world,” 2017 IEEE/RSJ international conference on intelligent robots and systems (IROS) , pp.23-30, IEEE, 2017.
 - (19) Symonyan, K. and Zisserman, A. : “Very deep convolutional networks for large scale image recognition,” arXiv preprint arXiv:1409.1556, 2014.
 - (20) Smoothie-3D : https://www.smoothie-3d.com/site/page_index.php (Accessed June 30, 2021.)
 - (21) Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., and Batra, D. : “Grad-cam: Visual explanations from deep networks via gradient-based localization,” International Journal of Computer Vision, 128, pp.336-359, 2020.
 - (22) Girshick, R., Donahue, J., Darrell, T., and Malik, J. : “Rich feature hierarchies for accurate object detection and semantic segmentation,” Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 580-587, 2014.
 - (23) Cheng, M. M., Zhang, Z., Lin, W. Y., and Torr, P. : “BING: Binarized normed gradients for objectness estimation at 300fps,” Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp.3286–3293, 2014.