Proceedings of the 2023 International Conference on Machine Learning and Automation
DOI: 10.54254/2755-2721/39/20230595

Africa wildlife prediction based on custom convolutional neural network

Kezhu Li

School of Electronic and Information Engineering, Hebei University of Technology, Tianjin, 300401, China

likezhu@hbgydx29.wecom.work

Abstract. In domains like psychology, computer science, and artificial intelligence, facial expression recognition has major ramifications. This paper suggests classifying input animal photos using convolutional neural network (CNN). The CNN layer's kernel size has been modified to (2,2), (3,3), (4,4). The primary objective is to examine how the model responds to changes in the CNN layer's kernel size and operation. African Wildlife, a data set of four African species, was utilized for the studies. There are 1508 different themes in this collection, divided into 4 groups with 377 photographs each. Different numbers of test photos and training images were used to determine overall performances. The custom CNN model with a kernel size of (3,3) achieved an accuracy of 57.57% on the dataset. According to the experimental findings, having a kernel that is either too large or too tiny may negatively impact the model and result in undesirable poor accuracy. This study could provide suggestions for predicting animals based on the development of convolutional neural networks.

Keywords: Animal Recognition, Convolutional Neural Network (CNN), Kernel Size.

1. Introduction

Although one can easily identify and distinguish between various images, this task remains difficult for recognition systems in computers [1]. An endless number of alternative pictures that are produced by changes in location, scale, perspective, backdrop, or illumination may be altered by each item of interest. Real-world issues like wild animal categorization from automatic trap cameras, where the majority of collected photos are of poor quality, present more significant challenges. As a result, it is crucial to create models for picture classification that can maintain their sensitivity to inter-class objects while being invariant to specific input changes [2].

CNN was first proposed in 1998 by Lecun et al [3], of NYU. It is an artificial neural network structure designed inspired by the visual neural structure of cats and then simulating its form. Recently, CNN has achieved great breakthroughs in computing power, such as parallel computing on graphics processing units and Tensorflow which even outperformed humans in the image recognition task [4]. Nowadays, CNN shows utility in natural language processing, image classification [5], and speech recognition [6].

In the ImageNet Large-Scale Visual Recognition Challenge (ILSVRC) [7], the AlexNet [8] has an ingenious structure allows it to perform the best in the challenge. A variation of the AlexNet model outperformed the second-best entry by over 10% in the top-5 test error rate [8]. GoogLeNet [9], the ILSVRC-2014 champion, created an Inception Module that significantly lowers the amount of parameters. Additionally, Google's network removed many parameters that had no bearing on network performance by replacing the fully linked top layers with average pooling.

In this project, CNN-based deep learning model is created for predicting images of 4 different species of African animals: rhino, zebra, elephant, and buffalo. Specifically, picturing such visuals to comprehend how to go forward. To continue the experiment, a data frame is created in the second step. Third, a name array of animal species is created to remove animal names from the test data following the prediction. Additionally, the Conv layer's kernel sizes are altered to see how the model responds and how the CNN layer functions. Confusion matrix creation, however, is done to understand how the characteristics are related to one another biologically or graphically. The experimental findings show that CNN can accurately predict pictures and that having a large or small kernel would negatively impact the model's performance and result in unfavorable accuracy. Furthermore, the kernel size of (3,3) is a perfect fit for this project. Based on the advancement of convolutional neural networks, this research may offer recommendations for wildlife prediction.

2. Methodology

2.1. Dataset description and preprocessing

The experiments used a data collection called African Wildlife which is resourced from Kaggle [10], and included YOLO labels for four different African animals. This collection has 1508 distinct topics, broken down into 4 categories with 377 images each. This data collection includes four animal types that are often seen in South African nature reserves: buffalo, elephant, rhino, and zebra. This data collection was first gathered to teach an embedded system how to recognize animals in South African nature reserves in real-time. Some images are shown in Figure 1, which is from the African Wildlife dataset.

To calculate the total results, several test photo and training image numbers were employed. The file paths are gathered, and pictures are presented to help with the next step. Then create distinct training programs and evaluations for each kind of animal. Second, link the train and test dataset components in each folder, then reshuffle them. The data frame is also created. To identify the names of the animals from the test data following the prediction, an array of animal names has been created. The target value format is changed to int64 since the target values should be integers. To determine if the data frame is well-built or not, visualize the photographs from the dataset once. As it functions flawlessly, the project may move forward.

Figure 1. Images from the African Wildlife dataset.

2.2. Proposed Approach

In this project, CNN-based deep learning model is created for predicting images of 4 different species of African animals: rhino, zebra, elephant, and buffalo. As shown in Figure 2, the first step in creating a deep learning model for predicting photos is to set up the environment, download the data, and unzip it. Secondly, the necessary libraries should be imported. Thirdly, the file paths for the photographs are identified by understanding the stacking of folders. After that, the route is gotten and basic visualizations

DOI: 10.54254/2755-2721/39/20230595

are done. What's more, the photos are combined with data frames. Moreover, the data is ready to feed into the model and it comes to a compilation for the model after it has been built. Finally, the model gets fitted and assessed.

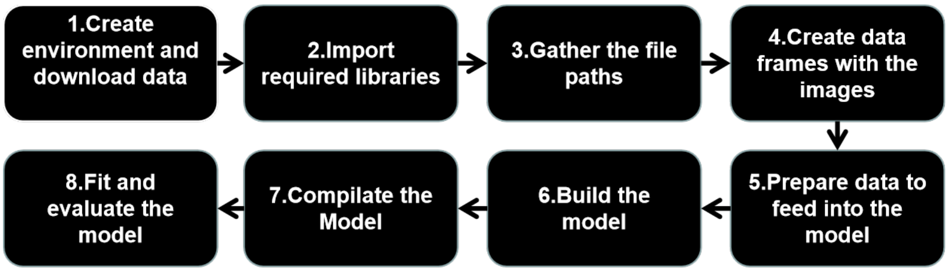

Figure 2. The pipeline of the model.

2.2.1. Convolutional Neural Network. The CNN is a neural network-based learning model for analyzing pictures's spatial structure which is a three-dimensional volume with a width, height, and depth. The Rectifier Linear Unit (ReLU) [10] combined with a max pooling layer, before getting completed with some interconnected layers. Neurons in a traditional neural network are completely linked to other neurons in the layer below them and are independent of others. Parameter sharing is one crucial characteristic of CNNs which can dramatically lower the number of variables and lower processing complexity.

2.2.2. Loss Function. To successfully train deep learning models, it is essential to select the appropriate loss. The Categorical Cross entropy loss function, which excels at multi-class classification issues, is the best option for this emotion classification challenge. By measuring the discrepancy between the model's predictions and the real labels, categorical cross-entropy encourages the model to give the right class a greater probability during the training phase. The model's estimated percentages for each emotion class are evaluated compared to the one-hot encoded real label, and the loss is determined for each image.

2.3. Implementation Details

The runtime type of the project is GPU. For the library, this project is proceeding with TensorFlow. Some supporting libraries will help to wrangle the data and produce the data frames and others. Last but not least some libraries are imported to make a confusion matrix as this is a problem of multiclass classification.

Specifically, picturing such visuals to comprehend how to go forward. To continue the experiment, a data frame is created in the second step. Third, a name array of animal species is created to remove animal names from the test data following the prediction. To identify the names of the animals from the test data following the prediction, create a name array of the various animal species. The target value format should be modified to int64 since the target values should be integers. Additionally, display a single image from the dataset to determine the quality of the data frame's construction. It certainly performs well.

Declare a batch size of 20 for the project's most crucial step, model fitting, since this will be helpful for the model's correctness and timeliness. 10 epochs are installed for training. To see how the model performed and determine whether it is overfitting, underfitting, or has little to no bias, store this fitting into a variable hist.

3. Result and Discussion

Two conclusions have been explored in this section. First off, having a kernel that is either too large or too little would negatively impact the model, resulting in unfavorable accuracy problems. What's more,

DOI: 10.54254/2755-2721/39/20230595

the kernel size of (3,3) works extremely well for this project. Second, a confusion matrix is created for the dataset to test the model's prediction, and it reveals that the model predicts buffalo and zebra the best and elephants decently. In certain instances, the zebra has been anticipated incorrectly. Additionally, there may be some biologically generated errors.

3.1. Kernel Sizes

The Conv-layer's kernel sizes are modified to (2,2), (3,3), and (4,4) to observe how the model responds. The model is overfitted after the fifth epoch with a validation accuracy of 62.83% when the kernel size is (3,3), as can be seen in Figures 3, 4, and 5 below. Additionally, when the kernel size is (4,4), the model is overfitted after the fourth iteration, and the validation accuracy is 58.55%. When the kernel size is (2,2), the model is overfitted after the third epoch, and the validation accuracy is 49.34%, which is not significantly better than previous scenarios.

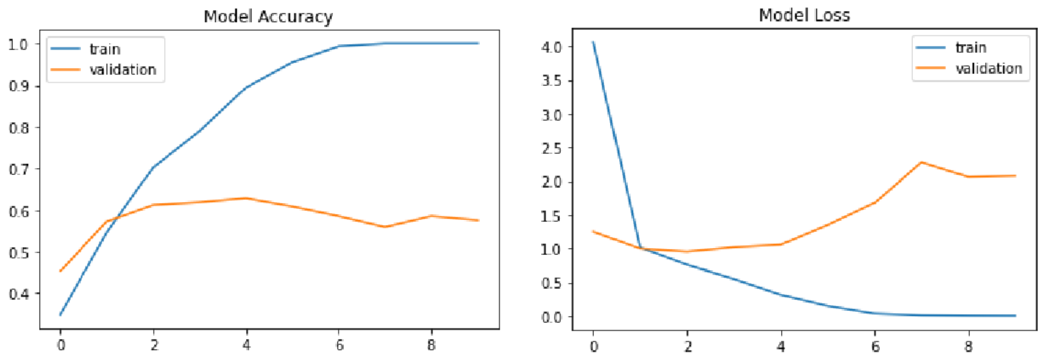

Figure 3. The accuracy and loss of the custom CNN model with the kernel size of (3,3).

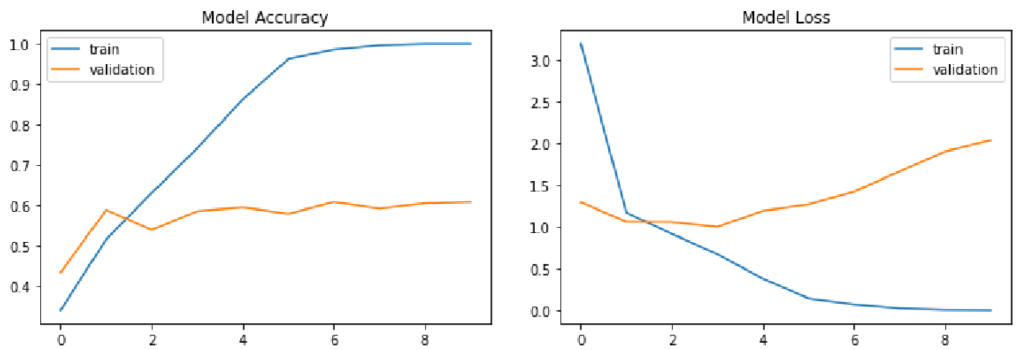

Figure 4. The accuracy and loss of the custom CNN model with the kernel size of (4,4).

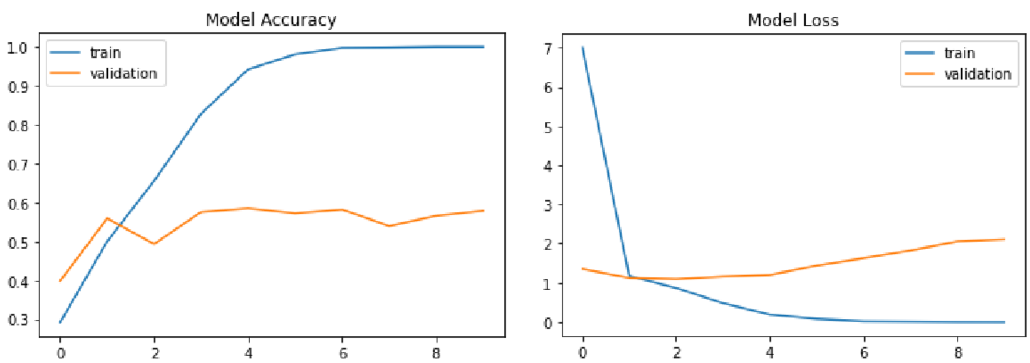

Figure 5. The accuracy and loss of the custom CNN model with the kernel size of (2,2).

It is concluded that having a kernel that is either too large or too tiny would hurt the model and result in unfavorable accuracy problems. what's more, the kernel size of (3,3) works extremely well for this project.

3.2. *Confusion Matrix Generation*

A confusion matrix is created, as illustrated in Figure 6, to show how the characteristics are related to one another in graphical or biological form.

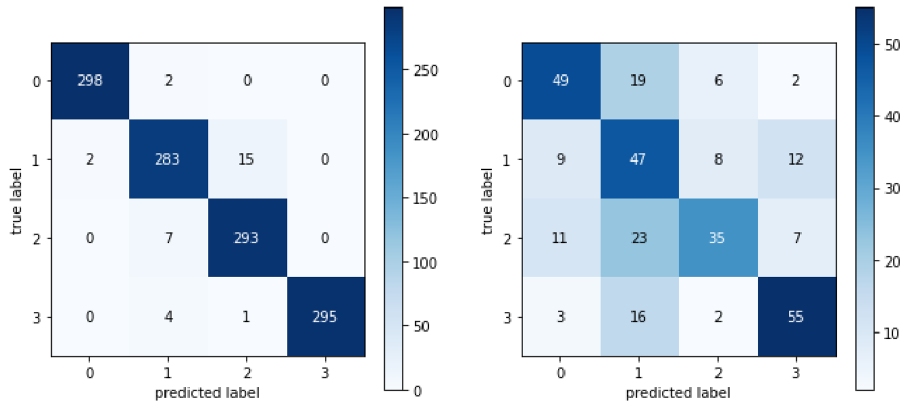

Figure 6. The confusion matrix of the test data ("0" represents buffalo, "1" represents elephant, "2" represents rhino, "3" represents zebra).

The test data's confusion matrix shows that the most accurate predictions are for buffalo and zebra. Additionally, while most predictions for elephants are accurate, some anticipate zebras, which is strange. It can be seen that the rhino was mistakenly forecasted as an elephant, which is a poor prediction. Moreover, in rare instances, zebras have been mistaken for rhinos.

Since it anticipated zebras instead of elephants, the model is not tailored to fit an elephant. A model mistake caused it. Given how closely the colors of the rhinoceros and elephant match and how symmetrical they are, this may be a biological mistake.

4. Conclusion

In this project, CNN-based deep learning model is created for predicting images of 4 different species of African animals: rhino, zebra, elephant, and buffalo. Create an environment first, then download and unzip the data, and import the necessary libraries. Second, comprehend the collections of image folders to compile their file paths and principal visuals. Thirdly, prepare the data for the model by creating data frames using the photos. Model construction and compilation are thus possible. The model fitting and assessment are the final and most crucial steps. A confusion matrix is built for the dataset to test the model's prediction, and the kernel sizes of the Conv layer are altered to examine how the model responds.

On the dataset, the customized CNN model had an accuracy of 57.57%. According to the experimental findings, the model accurately predicts elephants, zebra, and buffalo. The zebra has occasionally been predicted erroneously. There could also be some inaccuracies caused by biology. Additionally, a kernel that is either too big or too little would have a detrimental effect on the model, leading to undesirable accuracy issues. The study goal for the following stage will be to investigate further strategies for expanding the dataset and using deeper CNN models to improve the performance of the model. Three CNN designs with varying depths—Lite AlexNet, VGG-16, and ResNet-50—will be the subject of the study.

References

[1] N. Pinto, D. D. Cox, and J. J. DiCarlo 2008 Why is real-world visual object recognition hard? PLOS Computational Biology vol 4 no 1 p e27

[2] C. M. Bishop 2006 Pattern recognition Machine Learning vol 128 p 1–58
[3] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel 1989 Backpropagation applied to handwritten zip code recognition Neural Computation vol 1 no 4 p 541–551
[4] D. Ciregan, U. Meier, and J. Schmidhuber 2012 Multi-column deep neural networks for image classification Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) p 3642–3649
[5] O. Russakovsky, J. Deng, H. Su et al. 2015 ImageNet large scale visual recognition challenge International Journal of Computer Vision vol 115 no 3 p 211–252
[6] R. Collobert and J. Weston 2008 A unified architecture for natural language processing: Deep neural networks with multitask learning Proceedings of the 25th International Conference on Machine Learning (ICML) p 160–167
[7] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei 2009 ImageNet: A large-scale hierarchical image database Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) p 248–255
[8] A. Krizhevsky, I. Sutskever, and G. E. Hinton 2012 ImageNet classification with deep convolutional neural networks Advances in Neural Information Processing Systems p 1097–1105
[9] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich 2015 Going deeper with convolutions Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) p 1–9
[10] https://www.kaggle.com/datasets/biancaferreira/african-wildlife