International Journal of Science and Research (IJSR)
ISSN: 2319-7064
SJIF (2022): 7.942

Novel Approach of Animal Recognition Using Deep Learning Algorithm

Kavita Dhurve[1], Margi Patel[2]

[1]MTech Scholar, Indore Institute of Science and Technology, Indore, Madhya Pradesh, India
dhurvekavita24[at]gmail.com

[2]Assistant Professor, Indore Institute of Science and Technology, Indore, Madhya Pradesh, India
margi.patel[at]indoreinstitute.com

Abstract: ***Efficient and reliable monitoring of wild animals in their natural habitat is essential. This project develops an algorithm to detect the animals in wild life. Since there are large number of different animals manually identifying them can be a difficult task. This algorithm classifies animals based on their images so we can monitor them more efficiently. Animal detection and classification can help to prevent animal-vehicle accidents, trace animals and prevent theft. This can be achieved by applying effective deep learning algorithms.***

Keywords: Animal Detection and Classification, Deep Learning Algorithms

1. Introduction

Machine learning is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed. Machine learning focuses on the development of computer programs that can access data and use it learn for themselves. The process of learning begins with observations or data, such as examples, direct experience, or instruction, in order to look for patterns in data and make better decisions in the future based on the examples that we provide. The primary aim is to allow the computers learn automatically without human intervention or assistance and adjust actions accordingly.

Deep learning is a subset of machine learning. Deep artificial neural networks are a set of algorithms that have set new records in accuracy for many important problems, such as image recognition, sound recognition, etc., In deep learning, a convolutional neural network (CNN) is a class of deep neural networks, most commonly applied to analyzing visual imagery. CNNs use relatively little pre-processing compared to other image classification algorithms. This means that the network learns the filters that in traditional algorithms were hand-engineered. This independence from prior knowledge and human effort in feature design is a major advantage. They have applications in image and video recognition, recommender systems, image classification, medical image analysis, and natural language processing.

One of the applications of the deep learning technique called Convolutional Neural Network is animal detection. Observing wild animals in their natural environments is a central task in ecology. The fast growth of human population and the endless pursuit of economic development are making over-exploitation of natural resources, causing rapid, novel and substantial changes to Earth's ecosystems.

An increasing area of land surface has been transformed by human action, altering wildlife population, habitat and behavior. More seriously, many wild species on Earth have been driven to extinction, and many species are introduced into new areas where they can disrupt both natural and human systems. Monitoring wild animals, therefore, is essential as it provides researchers evidences to inform conservation and management decisions to maintain diverse, balanced and sustainable ecosystems in the face of those changes.

2. Literature Survey

The purpose of animal detection is to prevent or reduce the number of animal-vehicle collisions. These systems are specifically aimed at the wild animals that can cause human death, injury and property damage. This system detects the wild animals before they enter the road.

Historically animal-vehicle collisions have been addressed by putting up signs that warn peoples of potential animal crossings. In other cases, wildlife warning reflectors or wildlife fences have been installed to keep animals away from the road. In some selected areas wildlife fencing has been combined with a series of wildlife crossing structures. In most cases however, such crossing structures are limited in number and width, mostly because of their relatively high costs.

a) Animal Detection Using Template Matching Algorithm

Animal detection is useful in prevention of animal- vehicle accidents and will increase human and wildlife safety, it will detect large animals before they enter the road and warn the driver through audio and visual signals. This also helps in saving crops in farm from animals. In this project there is survey of different object detection techniques and for object identification as animal techniques such as object matching, edge based matching, skeleton extraction. After survey the most appropriate method is selected for animal detection and efficiency is measured. Proposed system has low false positive rate and false negative rate.

Template Matching

Template matching is a technique in digital image processing for finding small parts of an image which match a template image. To perform template matching the concept of normalized cross co relation can be used. In signal processing, cross-correlation is a measure of similarity of two waveforms as a function of a time-lag applied to one of them.

This is also known as a sliding dot product or sliding inner-product. It is commonly used for searching a long- duration signal for shorter, known feature. For image- processing applications in which the brightness of the image and template can vary due to lighting and exposure conditions, the images can be first normalized. This is typically done at every step by subtracting the mean and dividing by the standard deviation. Here we have used feature-based template matching mechanism using NCC

b) *Automatically Identifying, Counting, and Describing Wild Animals in Camera-Trap Images with Deep Learning*

Having accurate, detailed, and up-to-date information about the location and behavior of animals in the wild would revolutionize our ability to study and conserve ecosystems. This paper investigates the ability to automatically, accurately, and inexpensively collect such data, which could transform many fields of biology, ecology, and zoology into “big data” sciences. Motion sensor “camera traps” enable collecting wildlife pictures inexpensively, unobtrusively, and frequently. However, extracting information from these pictures remains an expensive, time-consuming, manual task. We demonstrate that such information can be automatically extracted by deep learning, a cutting-edge type of artificial intelligence.

We train deep convolutional neural networks to identify, count, and describe the behaviors of 48 species in the 3.2-million-image Snapshot Serengeti dataset. Our deep neural networks automatically identify animals with over 93.8% accuracy, and we expect that number to improve rapidly in years to come. More importantly, if our system classifies only images, it is confident about, our system can automate animal identification for 99.3% of the data while still performing at the same 96.6% accuracy as that of crowd sourced teams of human volunteers, saving more than 8.4 years (at 40 hours per week) of human labelling effort (i.e. over 17,000 hours) on this 3.2-million-image dataset.

Those efficiency gains immediately highlight the importance of using deep neural networks to automate data extraction from camera-trap images. Our results suggest that this technology could enable the inexpensive, unobtrusive, high-volume, and even real- time collection of a wealth of information about vast numbers of animals in the wild.

3. Block Diagram

a) *Convolutional Neural Network*

A convolutional neural network (CNN) is a specific type of artificial neural network that uses perceptron’s, a machine learning unit algorithm, for supervised learning, to analyze data. CNNs apply to image processing, natural language processing and other kinds of cognitive tasks. A convolutional neural network has an input layer, an output layer and various hidden layers. Some of these layers are convolutional, using a mathematical model to pass on results to successive layers.

- Input will hold the raw pixel values of the image and with three colour channels R, G, B.
- CONV layer will compute the output of neurons that are connected to local regions in the input, each computing a dot product between their weights and a small region they are connected to in the input volume.
- RELU layer will apply an element wise activation function. This leaves the size of the volume unchanged.
- POOL layer will perform a down sampling operation along the spatial dimensions (width, height), resulting in volume such as [16x16x12].

FC (i.e. fully-connected) layer will compute the class scores, resulting in volume of size. As with ordinary Neural Networks and as the name implies, each neuron in this layer will be connected to all the numbers in the previous volume.

b) *Convolutional Layer*

Fig. 1 shows the convolution which is the first layer to extract features from an input image. Convolution preserves the relationship between pixels by learning image features using small squares of input data. It is a mathematical operation that takes two inputs such as image matrix and a filter or kernel. Convolution of an image with different filters can perform operations such as edge detection, blur and sharpen by applying filters.

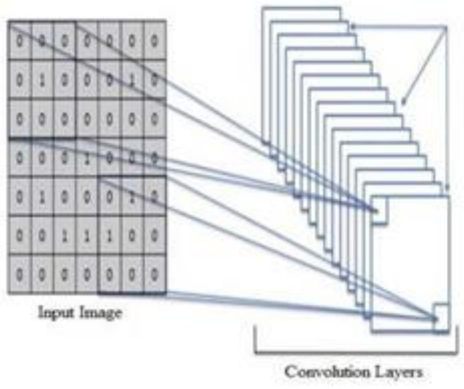


Figure 1: Convolutional Layer

c) *Pooling*

Pooling layers section would reduce the number of parameters when the images are too large. Max pooling take the largest element from the rectified feature map. The objective is to down-sample an input representation (image, hidden-layer output matrix, etc.), reducing its dimensionality. This is shown in Fig. 2.

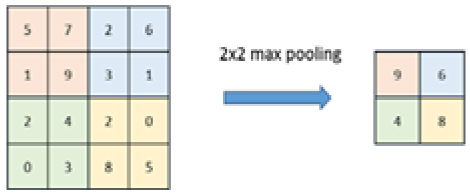


Figure 2: Pooling Layer

Flattening

Flattening is the process of converting all the resultant 2 dimensional arrays into a single long continuous linear vector. It gets the output of the convolutional layers, flattens all its structure to create a single long feature vector to be used by the dense layer for the final classification. This is shown in Fig. 3

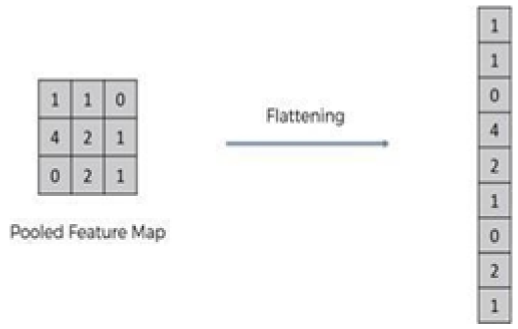


Figure 3: Flattening Layer

d) *Fully connected*

Fig.4 shows the hidden layers inside a Convolutional Neural Network w are called Fully Connected Layers. These are a specific type of hidden layer which must be used within the CNN. This is used to combine the features into more attributes that predict the outputs more accurately.

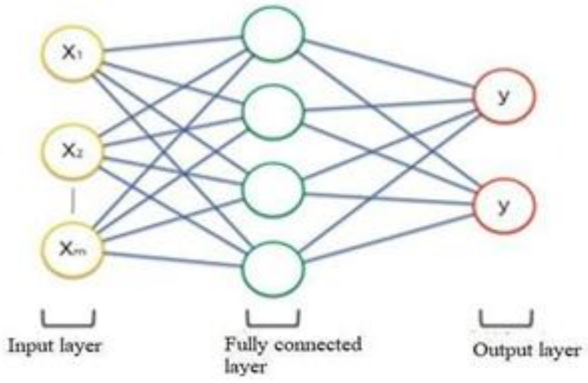


Figure 4: Fully Connected Layer

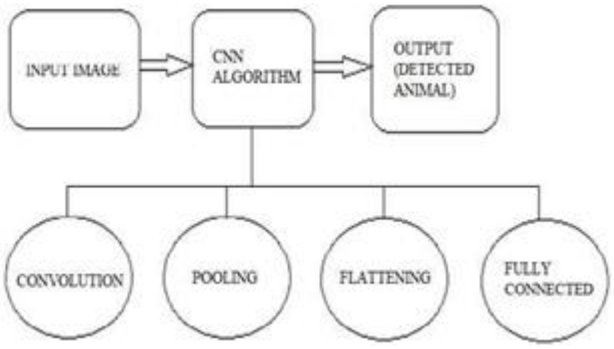


Figure 5: Block Diagram

Fig. 5 shows the block diagram of the animal detection using deep learning algorithm.

Flow Diagram

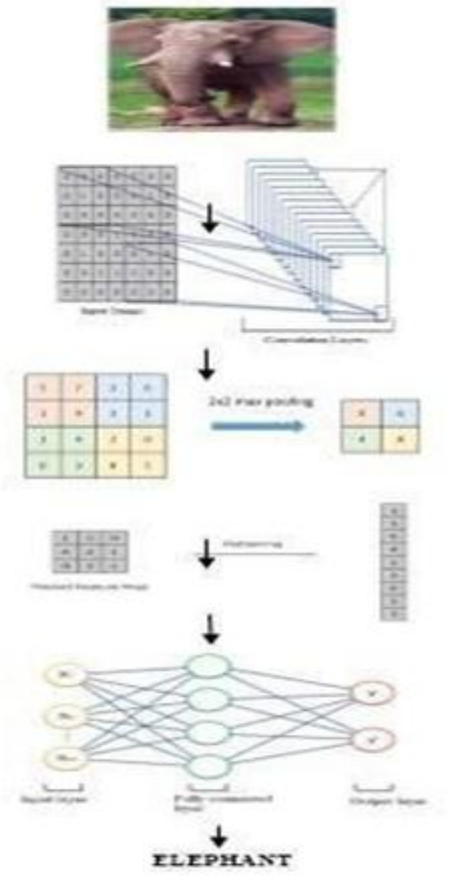


Figure 6: Flow Diagram

Fig. 6 shows the flow diagram of animal detection.

1) Dataset

The dataset used here is a collection of image data that contains various images of animals. The Dataset is splitted into train and test in the ratio of 75:25 respectively.

a) *Train Dataset Elephant Train Dataset*

Figure 7: Elephant Train Dataset

b) *Cheetah Train Dataset*

Figure 8: Cheetah Train Dataset

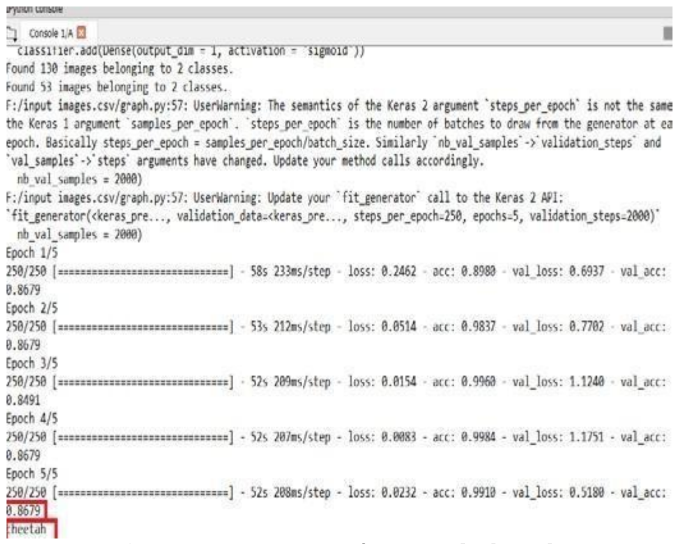
c) Test dataset Elephant Test Dataset

Figure 9: Elephant Test Dataset

d) Cheetah Test Dataset

Figure 12: Detected Elephant

e) Detection of Cheetah



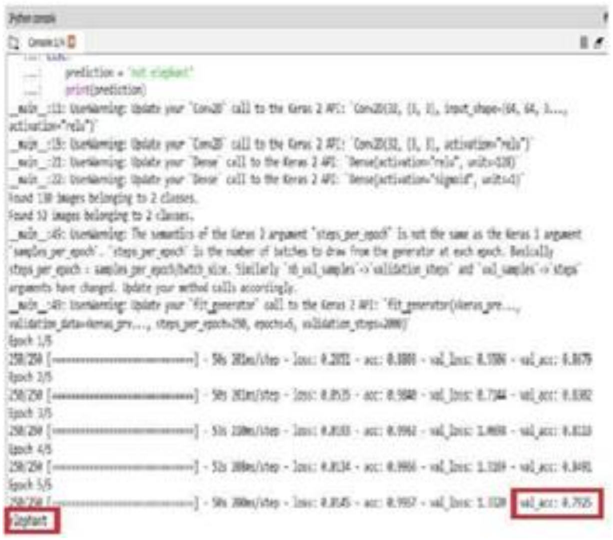
```
Python console
Console 1/A
 classifier.add(Dense(output_dim = 1, activation = 'sigmoid'))
Found 130 images belonging to 2 classes.
Found 53 images belonging to 2 classes.
F:/input images.csv/graph.py:57: UserWarning: The semantics of the Keras 2 argument `steps_per_epoch` is not the same
the Keras 1 argument `samples_per_epoch`. `steps_per_epoch` is the number of batches to draw from the generator at ea
epoch. Basically steps_per_epoch = samples_per_epoch/batch_size. Similarly `nb_val_samples`->`validation_steps` and
`val_samples`->`steps` arguments have changed. Update your method calls accordingly.
  nb_val_samples = 2000)
F:/input images.csv/graph.py:57: UserWarning: Update your `fit_generator` call to the Keras 2 API:
`fit_generator(<keras_pre..., validation_data=<keras_pre..., steps_per_epoch=250, epochs=5, validation_steps=2000)`
  nb_val_samples = 2000)
Epoch 1/5
250/250 [==============================] - 58s 233ms/step - loss: 0.2462 - acc: 0.8980 - val_loss: 0.6937 - val_acc:
0.8679
Epoch 2/5
250/250 [==============================] - 53s 212ms/step - loss: 0.0514 - acc: 0.9837 - val_loss: 0.7702 - val_acc:
0.8679
Epoch 3/5
250/250 [==============================] - 52s 209ms/step - loss: 0.0154 - acc: 0.9960 - val_loss: 1.1240 - val_acc:
0.8491
Epoch 4/5
250/250 [==============================] - 52s 207ms/step - loss: 0.0083 - acc: 0.9984 - val_loss: 1.1751 - val_acc:
0.8679
Epoch 5/5
250/250 [==============================] - 52s 208ms/step - loss: 0.0232 - acc: 0.9910 - val_loss: 0.5180 - val_acc:
0.8679
cheetah
```

Figure 13: Accuracy of Detected Cheetah

4. Result

Figure 10: Cheetah Test Dataset

Detection of Elephant



```
Python console
Console 1/A
         prediction = 'not elephant'
         print(prediction)
__main__:11: UserWarning: Update your `Conv2D` call to the Keras 2 API: `Conv2D(32, (3, 3), input_shape=(64, 64, 3...,
activation="relu")`
__main__:15: UserWarning: Update your `Conv2D` call to the Keras 2 API: `Conv2D(32, (3, 3), activation="relu")`
__main__:21: UserWarning: Update your `Dense` call to the Keras 2 API: `Dense(activation="relu", units=128)`
__main__:22: UserWarning: Update your `Dense` call to the Keras 2 API: `Dense(activation="sigmoid", units=1)`
Found 130 images belonging to 2 classes.
Found 53 images belonging to 2 classes.
__main__:45: UserWarning: The semantics of the Keras 2 argument "steps_per_epoch" is not the same as the Keras 1 argument
`samples_per_epoch`. `steps_per_epoch` is the number of batches to draw from the generator at each epoch. Basically
steps_per_epoch = samples_per_epoch/batch_size. Similarly `nb_val_samples`->`validation_steps` and `val_samples`->`steps`
arguments have changed. Update your method calls accordingly.
__main__:49: UserWarning: Update your `fit_generator` call to the Keras 2 API: `fit_generator(<keras_pre...,
validation_data=<keras_pre..., steps_per_epoch=250, epochs=5, validation_steps=2000)`
Epoch 1/5
250/250 [==============================] - 59s 283ms/step - loss: 0.2071 - acc: 0.8888 - val_loss: 0.5586 - val_acc: 0.8679
Epoch 2/5
250/250 [==============================] - 50s 201ms/step - loss: 0.0535 - acc: 0.9840 - val_loss: 0.7344 - val_acc: 0.8302
Epoch 3/5
250/250 [==============================] - 51s 220ms/step - loss: 0.0133 - acc: 0.9962 - val_loss: 1.0698 - val_acc: 0.8113
Epoch 4/5
250/250 [==============================] - 52s 208ms/step - loss: 0.0134 - acc: 0.9966 - val_loss: 1.1109 - val_acc: 0.8491
Epoch 5/5
250/250 [==============================] - 50s 200ms/step - loss: 0.0145 - acc: 0.9957 - val_loss: 1.1120 - val_acc: 0.7925
Elephant
```

Figure 11: Accuracy of Detected Elephant

5. Conclusion

Thus, this project uses Convolutional Neural Network (CNN) algorithm to detect wild animals. The algorithm classifies animals efficiently with a good number of accuracy and also the image of the detected animal is displayed for a better result so that it can be used for other purposes such as detecting wild animals entering into human habitat and to prevent wildlife poaching and even human animal conflict.

This work can be further extended by ending an alert in the form of a message when the animal is detected to the nearby forest office. Furthermore, it can be used to reduce human wildlife conflict and also animal accidents.

References

[1] Xie, Z., A. Singh, J. Uang, K.S. Narayan and P.Abbeel. Multimodal blending for high-accuracy Instance cognition. In: 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems. Tokyo: IEEE 2013, pp. 2214-2221. ISBN 978-1-4673-6356-3. DOI: 10.1109/IROS.2013.

[2] Tiber Trnovszky, Patrik Kamencay, Richard Orjesek, Miroslav Benco, Peter Sykora. Animal recognition system based on convolutional neural network.

[3] Ahonen, T., Hadid, A., Pietikainen, and M.: Face description with local binary patterns:Application to face recognition. IEEE TPAMI 28(12), 2037-2041 (2006).

[4] Burghardt, T., Calic, J.: Real-time face detection and tracking of animals. In:Neural Network Applications in Electrical Engineering. pp. 27{32. IEEE (2006).

[5] Felzenszwalb, P.F., Girshick, R.B., McAllester, D., Ramanan, D.: Object detection with discriminatively trained part- based models. IEEE TPAMI 32(9), 1627-1645(2010).

[6] He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR. pp. 770-778 (2016).

[7] Kamencay, P., T. Trnovszky, M. Benco, R. Hudec, P. Sykora and A. Satnik. Accurate wild animal recognition using PCA, LDAand LBPH, In: 2016 ELEKTRO. Strbske Pleso: IEEE, 2016, pp. 62–67. ISBN 978-1-4673-8698-2.DOI: 10.1109/ELEKTRO.2016.7512036.

[8] WU, J. L. and W. Y. MA. A Deep Learning Framework for Coreference Resolution Based on Convolutional Neural Network. In: 2017IEEE 11th International Conference on Semantic Computing (ICSC). San Diego: IEEE,2017, pp. 61– 64. ISBN 978-1-5090-4284-5.DOI: 10.1109/ICSC.2017.57.

[9] P. M. Vitousek, H. A. Mooney, J. Lubchenco, J. Melillo, "Human domination of Earth's ecosystems", Science, vol. 277, no. 5325, pp. 494-499, 1997.

[10] G. C. White, R. A. Garrott, Analysis of wildlife radio-tracking data, Elsevier, 2012.

[11] J. Godley, J. Blumenthal, A. Broderick, M. Coyne, M. Godfrey, L. Hawkes, M. Witt, "Satellite tracking of sea turtles: Where have we been and where do we go next?", Endangered Species Research, vol. 4, no. 1–2, pp. 3-22, 2008.

[12] Gomez, A. Salazar, F. Vargas, towards automatic wild animal monitoring: Identification of animal species in camera-trap images using very deep convolutional neural networks, 2016.