ANIDETNET USING EFFICIENTNET-B3 MODEL WITH TRANSFER LEARNING

S. Karthigai Selvi

Abstract
Image classification is now used to bridge the gap between computer vision and human vision, allowing machines to recognise images in the same way that humans do. It deals with assigning the appropriate class to the given image. We therefore propose a system and web application named AniDetNet using Efficientnet- B3 with transfer learning that classifies the given images as cows, cats, dogs, elephants, and pandas. This paper presents a modified network model and investigates the relationship between learning rate and accuracy before providing a computer application. As a result, the user gains knowledge about the given animal image while avoiding the risk of classification using CNN. The proposed methodology accounts 98.5% accuracy and recognition rate.

Corresponding Author: - S. Karthigai Selvi
Department of Computer Science and Applications, The Gandhigram Rural Institute – Deemed to be University, Gandhigram, India.
E-mail:- karthigachandru@gmail.com

DOI: 10.48047/ecb/2023.12.si5a.0361

1. Introduction

A human being lives with several creatures in the world and is willing to interact with them. The animals, such as cats, elephants, dogs, etc., provide happiness and stress relief to the pet keeper. However, some animals attack humans and cause injuries. Yudin et al. [1] reported that two million animals attack humans each year. The accidents might happen when the animals roam seeking food or water. Sometimes animals also get injured because of vehicles. Animal protection is also essential to securing our biodiversity and preserving it. Our young generations do not have much knowledge about animals since they are growing in a building. Subsequently, animal tracking is essential for forestry services and remote villages. In the artificial intelligence era, computer vision techniques scan multiple objects and are involved in object recognition [2]. To resolve these issues, we propose an image-based animal detection system called AniDetNet. The system is based on Artificial Intelligence (AI), which necessitates the automation of processes. AI includes Machine Learning (ML), which provides promising results in automation. ML mimics the human learning system and branches Neural Network (NN) in the same way. NN is made up of several layers, artificial neurons, connections between neurons, and some mathematical framework. It is employed in complex problems such as recognition, classification, and prediction tasks in classification [3].

Deep learning is a subset of neural networks and differs by the number of layers and their arrangement. Convolutional neural network (CNN) is one of the models under deep learning. The CNN is frequently used in the field of image recognition and classification. Other ML models accept features as input data, but CNN takes images as inputs and extracts features from them. The updated version of deep learning is called transfer learning (TL), which reuses the data of pre-trained models. Thus, it minimises the computational complexity and time consumption [4].

Since the manual analysis of animals from photographs is a tedious process and consumes more time [5]. The detection of animals using camera images is very encouraging for animal researchers, pet keepers, and wildlife photographers. Image processing techniques and AI take part in the animal recognition task [6].

In this model (AniDetNet), we present a deep learning method for animal image classification and recognition based on the new Efficient Net B3 with Transfer Learning model. Specially, our proposed system utilizes EfficientNet-B3, where a strategic fixation of learning rate improves the system. The following are the main contributions of the paper: 1) This paper proposes a method to classify the image by adjusting the learning rate in Efficient Net B3 with transfer learning, 2) erasing the background and putting a bounding box around the animal in the image using CNN (the entire system is called AniDetNet), and 3) testing the proposed model using Kaggle datasets.

The remaining part of this paper contains three sections and 15 paragraphs organised as methodology, results, discussion, and conclusion. The methodology section described the architecture of the ANNetDet system, the results and discussion shared the experiences obtained during the experiment, and the conclusion section summarised the results.

2. Proposed Method

The proposed system AniDetNet operates in two stages: the first recognises images using the Efficient Net B3 model, and the second separates the animal image from its background. The system allows users to register in the network and learn about animals by uploading images. The system accepts the images as input from the user and feeds them into Efficient Net B3, which is pre-trained by Image Net. Subsequently, the image is extracted using CNN from its background. Figure 1 depicts the detailed architecture of the proposed system. The data set is collected from the kaggle.com website, which contains 7500 images in five various classes (cat, elephant, cow, dog, and panda), with each class holding 1500 images.

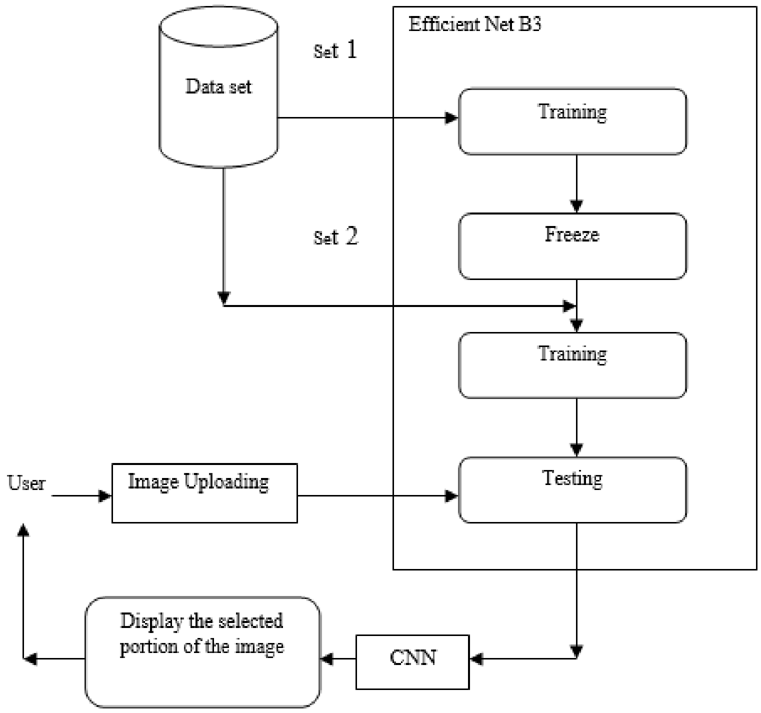
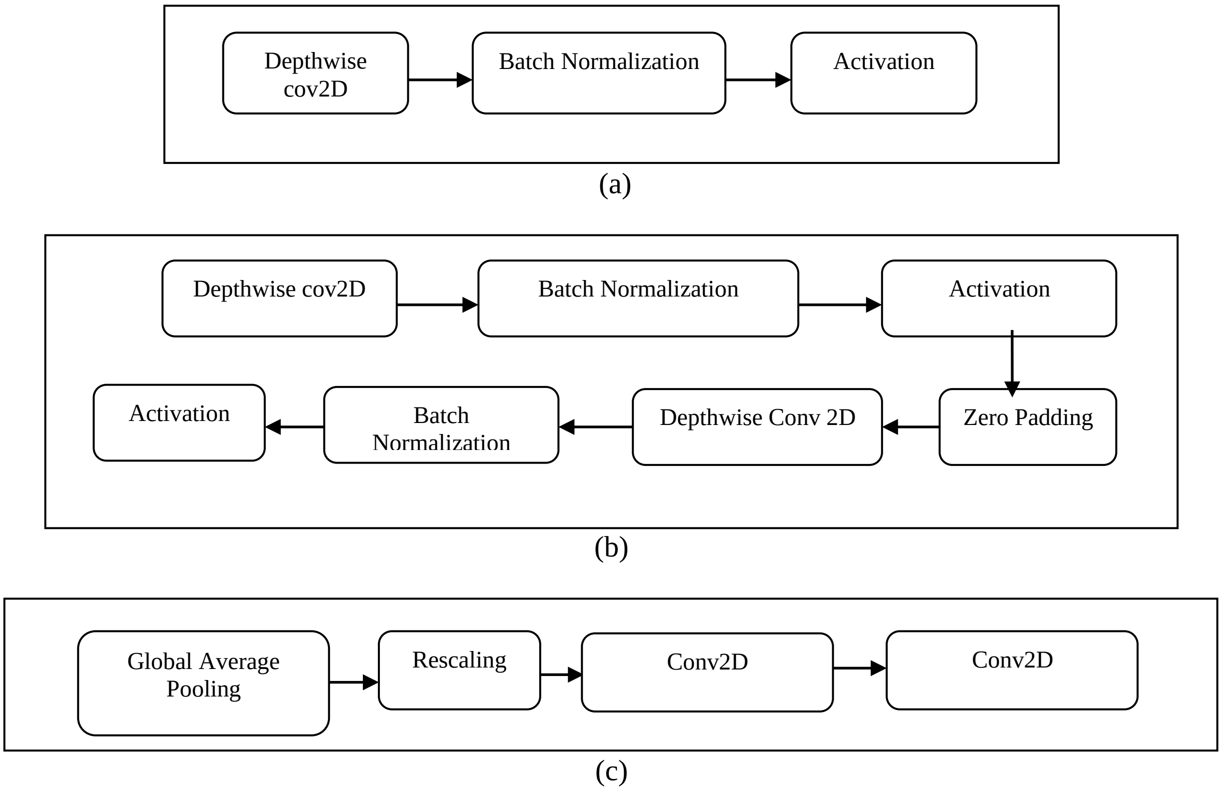
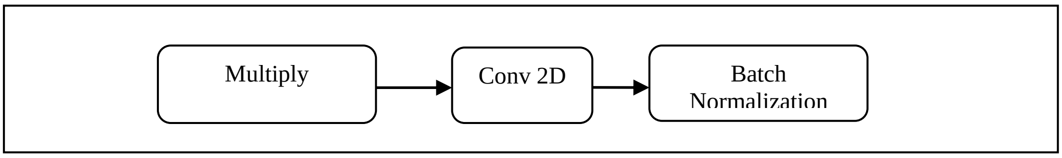


Figure 1. AniDetNet architecture

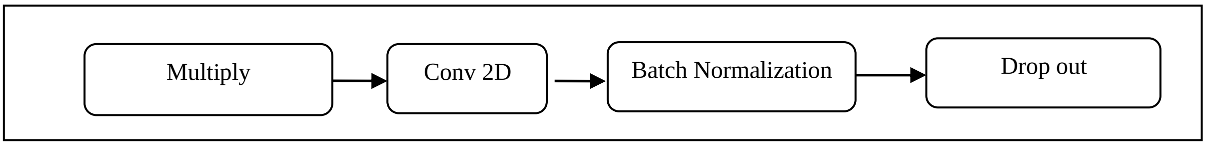
2.1.EfficientNet B3

Tan et al. [7] observed the CNN model and investigated its breadth and depth, then proposed eight models called EfficientNet B0 to EfficientNet B7 by increasing the layers and nodes in a layer. Thus, it provides promising results in all applications compared to ResNet and ImageNet, and it holds 5,330,564 parameters. The EfficientNet uses the CNN model by increasing numerous layers and neurons. The detailed description of CNN is given in Section 2.2. The number of layers in each EfficientNet model varies, for example, the B0 model has 237 layers and the B7 model has 813 layers. From module 1 to module 5, each layer contains five modules. Figure 2 depicts the structure of the modules.

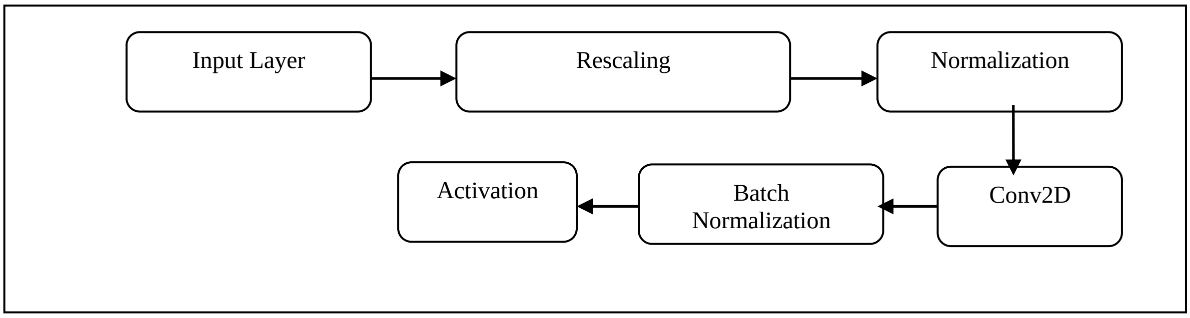




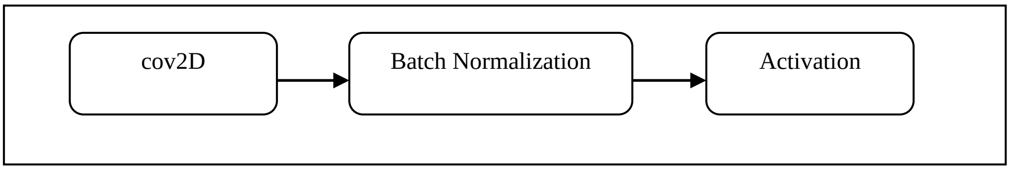
(d)



(e)



(f)



(g)

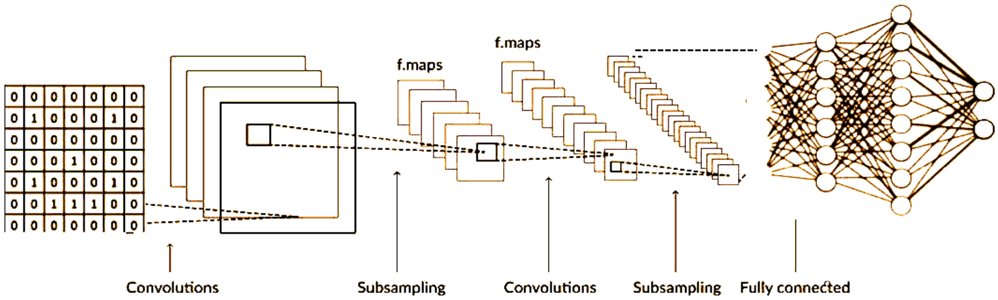
Figure 2. Modules of EfficientNet. (a) Module 1, (b) Module 2, (c) Module 3, (d) Module 4, (e) Module 5, (f) Stem Layer and (g) Final Layer (https://towardsdatascience.com/complete-architectural-details-of-all-efficientnet-models-5fd5b736142)

The modules are EfficientNet's components; they create sub-blocks and handle the analysis process. Module 1 involves the processes of convolution and normalisation and evaluates the results with an activation function. Module 2 performs depth-wise convolution, normalisation and padding of the batch of data, convolution again, and provides results for subsequent processes. Module 3 performs the pooling and rescaling operations. Module 4 rescales and normalises the image. The stem layer is the first layer that receives the data, rescales the data, and sends it to the modules. The final layer generates the result and serves as the output layer. The sub-block 1 consists of modules 1, 3, and 4, the sub-block 2 of modules 2, 3, and 4, and the sub-block 3 of modules 2, 3, and 5.Module 3 and module 4 act as the skip connections in the sub-block. Module 4 combines the skip connection, and module 5 connects each sub-block with the next sub-block. The EfficientNet B3 holds sub-block 2 as the starting point for all layers.

In most cases, the training process in ML entails assigning appropriate weights to each data set based on the input and output data to the network. The testing process uses weight to analyse the unknown data.

2.2. Convolution Neural Network (CNN)

Convolution and sub-sampling layers are connected to a fully connected layer that uses the softmax function in this network. Moving from input to output layers, a series of several convolution layers extracts features with increasing fineness at each layer. The convolution layers are followed by fully linked layers that carry out classification. Every convolution layer is frequently followed by a sub-sampling or pooling layer. Each layer is made up of clusters of 2D neurons known as kernels or filters. Unlike other neural networks, a 2D, n by n pixel image is given as the input for CNN one time. The neurons assembled in the network's feature extraction layer are not connected to the neurons in the nearby levels. Rather, they are only connected to the spatially mapped neurons, which are of fixed size; the preceding layer contains partially overlapped neurons. The area of input is referred to as the Local Receptive Field (LRF). The fewer connections mean less training time and a smaller likelihood of over-fitting. All neurons in a filter must have the same order of weights and biases and must be coupled to an equal number of neurons in the preceding input layer (or feature map). These elements accelerate learning and lower the network's memory needs. As a result, the neuron in a particular filter searches for the same pattern in several locations within the given image. The network's size is decreased by the sub-sampling layers. The sub-sampling layer reduces neuron size by using max or mean pooling or averaging filters to achieve sub-sampling. The actual classification work is done by the fully connected layers. The neurons in the layers are connected with other layers as given in Fig. 3. The layer strength of CNN highly supports its ability to provide excellent results in the job of object detection.



Figrue 3. Convolution Neural Network architecture

2.3. User Environment

The final and most important phase in the system life cycle is the implementation of the new system. Proper implementation of a research work is essential to provide a reliable system to meet user requirements. The user environment is crucial in the implementation process. This part of the work integrates into the above-mentioned user environment network model and properly connects with a graphic user environment. The AniDetNet work is developed using the Python language. This AniDetNet contains a user interface section that holds two parts: the registration part and the image search part. User information is collected and stored in a database through the registration process. Through the search part, the users can upload images to learn more about the animals in detail. The user request is sent to the testing phase of proposed EfficientNet B3, the resulted animal name is accepted, and the description of the animal is fetched from the animal database. Further, detailed information about the animal is displayed to the user. Some screen shorts are given in the Appendix part of this paper. The system is tested properly by providing various real-time animal images and other images.

3. Result and Discussion

The data set was taken from Kaggle.com which includes the dataset as given in Table 1.

Table 1. Classes, Training, Testing and validation.

Table 1. Training and testing data set details

Classes	Training	Testing	Validation
Cat	1500	381	500
Cow	1500	193	500
Dog	1500	89	500
Elephant	1500	306	500
Panda	1500	235	500

In the coding experiments, the training data frame contains 1500 images in each class from the kaggle.com dataset. As a result, the training data frame was 6000 images long, the test data frame was 1200 images long, and validation required 500 images from each class. During the training process, the average height and width of the image are selected to fix the image size. Because large images necessitate more layers and channels, 80% of the total image data frames were used to train the model, with the remainder used to test the model. While running the AniDetNet, each class was reduced to approximately 450 images, and the entire set was reduced to 2250 data frames with 5 classes. Additionally, augmented images are also generated by the processes of flipping, rotation, and scaling.

Usually, tensor flow learning involves using the features learned on one problem and applying them to new problems. Initially, the model is trained using one set of data, which holds half of the training images; based on the validation loss, the layers are frozen. Subsequently, we added some layers on top of the frozen ones. Hence, training was done with two batches of images. Finally, unfreeze the entire network and retrain with a new set of images. The AniDetNet assumes that the initial weight assignment is the only thing that consumes time. Generally, the weight obtained from each epoch is taken as the best weight for the next epoch. This model elapses approximately 17 seconds per step and for 10 epochs during training.

This model retains validation loss; at the first epoch, the validation loss is saved as the lowest loss, and it compares the validation loss to the subsequent epochs. The validation loss is shown in Fig. It shows that the loss decreases in the following epochs. Simultaneously, the accuracy is increasing from 0.6 to 0.9978. As given in Table 1.

The testing of the AniDetNet is carried out over 381 cat, 193 cow, 39 dog, 306 elephant, and 235 panda images. Fig4 shows the validation loss; Accuracy and f1 scores in Fig. 4 (a) illustrate that during training and validation, validation loss is very low compared to the training stage, and it attains low validation loss at the 10th epoch. The accuracy during the training phase for readers up to epoch 1 was 0.7%; the validation accuracy was less than 0.2% compared to the training accuracy and high accuracy readers at epoch 9. Cow's score increased significantly as well, with 0.996 at training and 0.98 at validation. Table 2 columns 3 and 4 show the recall value of the earlier and modified models. In this model, the proposed model provides a 0.1% higher result than the existing model. The F1 score value is shown in the last two columns of Table2. In a 150-dog detection task, the F1 score is as low as 0.95, but in Panda prediction, the score is raised to 0.99.

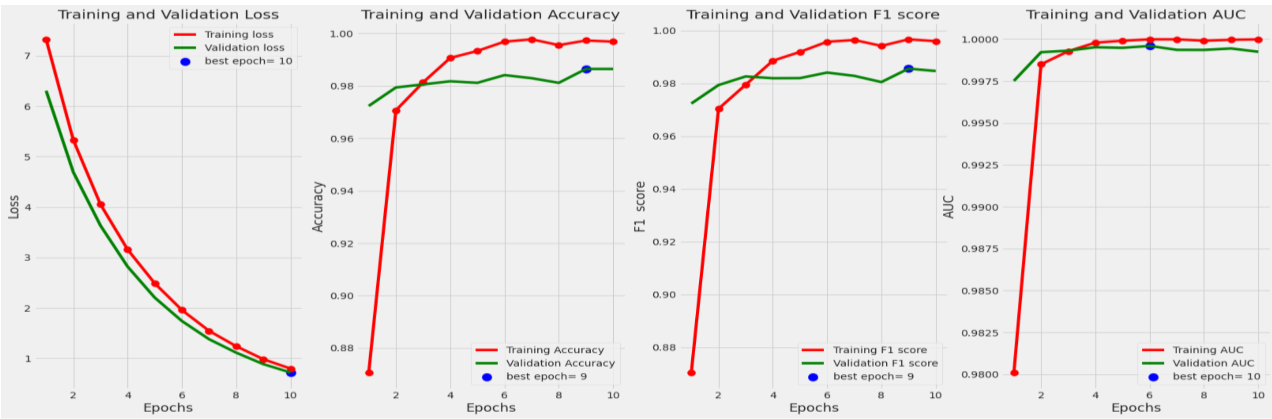
The results of AniDetNet's entire work are consolidated by confusion matrix and are shown in Fig.5.During the testing phase, 381 images were given, among which one image was identified as a dog and our image as a panda. Among the 191 test cow images, 177 images are identified as cows, and 0 items are recognised as cats (5), elephants (5), pandas (3), and dogs (1). Over 39 dog images, 83 were correctly recognised as dogs, and the other 6 were recognised as cats. In the elephant detection, among 306 images, 301 were correctly identified as elephants, while the other images were recognised as cats (2) and dogs (3). In the panda detection task, 233 images were correctly identified as pandas, while the remaining two were incorrectly identified as cats.

The overall accuracy of the system is 98%; some simple training and testing images are provided in the Appendix. The results are hallmarked using precision, recall, and F1-score. The results are illustrated in Table 2.

In the Table 2, two results were illustrated for each class of animals. The first precision, recall, and f1-score are the result of a lower learning rate of 0.001. In this system, learning rate is automatically updated when learning rate = learning rate x factor. The factor value is set at 1/3 of its learning rate. The results before and after adjusting the learning rate are given in the table. When the learning rate was reduced to 1/3 of the normal rate, the second sets of results were obtained.

Table 2. Quantitative validation

	Precision		**Recall**		**F1 Score**	
	Existing	**Proposed**	**Existing**	**Proposed**	**Existing**	**Proposed**
Cat	0.9948	0.9974	0.9619	0.9746	0.9859	0.9859
Cow	0.9326	0.9667	1	1	0.962	0.9748
Dog	0.9326	0.9348	0.9432	0.9773	0.9379	0.9556
Elephant	0.9837	0.9838	0.9837	0.9935	0.9837	0.9886
Panda	0.9915	1	0.9831	0.9958	0.9873	0.9979



(a) (b) (c) (d)

Figure 4. Validation accuracy and loss

A comparison with an earlier model is essential to ensure the significance of the modified version. The results of earlier and modified models are listed in precision value of EfficientNet 133 and the proposed model. The proposed model provides higher results for all classes than the existing one.

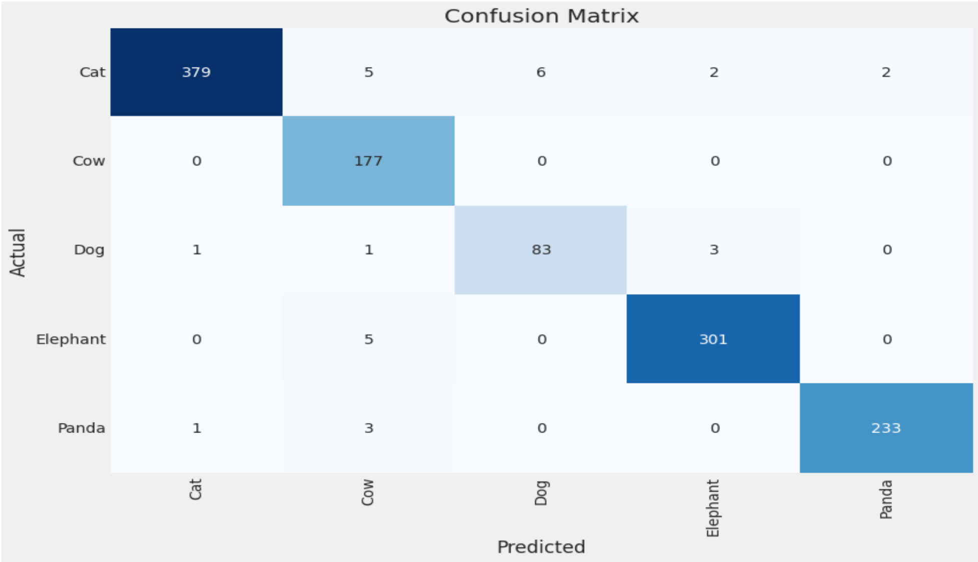


Figure 5. Confusion Matrix

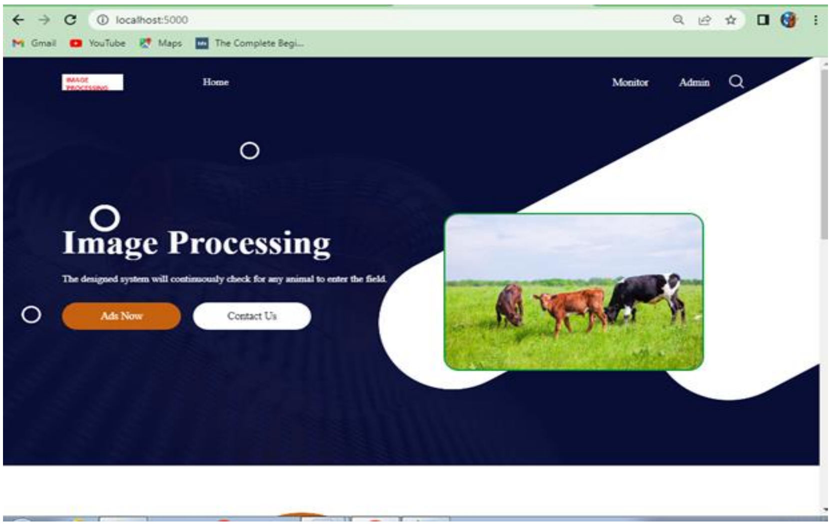
4. Conclusion

This paper presented a software model called AniDetNet is recognized and segmented 150 animals from image. The task of animal detection from an image is difficult and challenging one. However, it is essential for tourism and forestry department. 5 and common people who wants to get knowledge about animal. This model recognizes the animals' images using model EfficientNet B2. The experiments carried out using kaggle dataset, in different oriented and poor illuminated images. In order to improve the functionality to EfficentNet B3 the training mechanism was modified. The proposed system efficiently increased the recognition rate over the earlier. Experiments were conducted on our own dataset and the results showed that our classification and segmentain system has achieved an excellent accuracy rate of 99%. The proposed system is validated and the results show that the method is good and efficient.

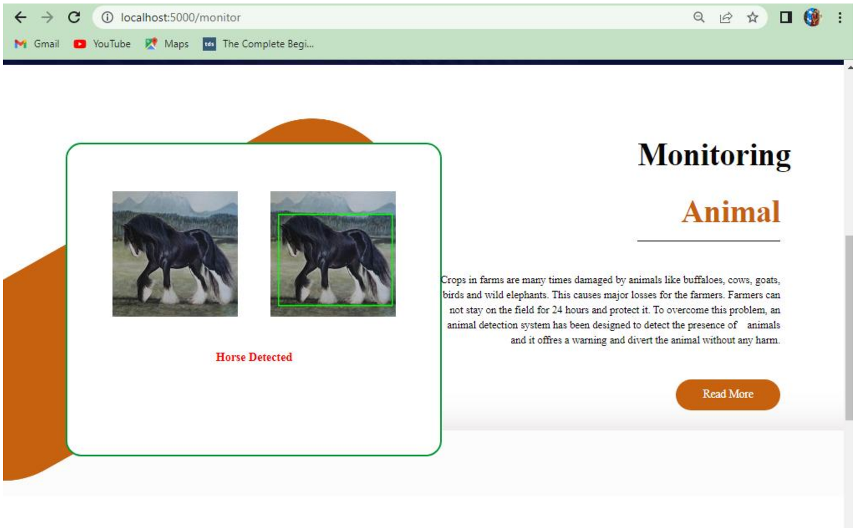
References

1. Yudin D,. Sotnikov A., Krishtopik A. Detection of Big Animals on Images with Road Scenes using Deep Learning. In Proceedings of the 2019 International Conference on Artificial Intelligence: Applications and Innovations (IC-AIAI), Vrdnik, Banja, Serbia, 30September–4 October 2019.
2. Matsuzaka Y, Yashiro R. AI-Based Computer Vision Techniques and Expert Systems. AI. 2023; 4(1) : 289-302. https://doi.org/10.3390/ai4010013
3. Murphy, Kevin P. Machine learning: a probabilistic perspective. MIT press, 2012.
4. Brownlee, Jason. Deep learning for computer vision: image classification, object detection, and face recognition in python. Machine Learning Mastery, 2019.
5. Norouzzadeh, Mohammad Sadegh, et al. A deep active learning system for species identification and counting in camera trap images. Methods in ecology and evolution, 2021, 12 (1) : 150-161.
6. Suryawanshi S., V. Jogdande, and A. Mane. Animal classification using deep learning. Int. J. Eng. Appl. Sci. Technol, 2020, 4 (11): 305-307.
7. M. Tan and Q. Le. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks, in International Conference on Machine Learning, May 2019, pp. 6105–6114, Accessed: Oct. 11, 2020. [Online]. Available: http://proceedings.mlr.press/v97/tan19a.html.

Apendix I
Home and other pages of the Application



Home Page Screen Shot



Animal Data