14

AnimNet: An Animal Classification Network using Deep Learning

Kanak Manjari[1*], Kriti Singhal[2], Madhushi Verma[1] and Gaurav Singal[1]

[1]*Computer Science Engineering Bennett University, Gr. Noida, India*

[2]*Computer Science Engineering Galgotias University, Gr. Noida, India*

Abstract

Image classification is a combination of technologies: Image Processing (IP), Machine Learning (ML), and Computer Vision (CV). The classification of animals has been done in this work that are commonly found in the Indian scenario using two approaches: transfer learning and a custom-built classification network, i.e., AnimNet. For transfer learning, we have used VGG16, VGG19, and Xception network that are existing pre-trained networks and compared the results of the custom-built AnimNet network with these existing networks. The comparison was done on the basis of the accuracy and size of the models as the size of network is as important as the accuracy of network in this era of mobile computing. A lightweight network with good performance is the most optimal choice nowadays. The accuracy was observed to be highest for the Xception network whereas the AnimNet network is lightweight, i.e., 5X smaller than the Xception model with second-highest accuracy.

Keywords: Image classification, animal detection, VGG16, VGG19, computer vision, deep learning

14.1 Introduction

Classification is a methodical categorization of images based on its characteristics in different classes. Some classifiers are binary, leading to a conclusion that is yes/no. Others are multi-class, capable of classifying an object into one of many classifications. Image classification emerged to reduce the lap between computer and human vision by training the computer with data. The conventional methods used for the image classification are segments of the Artificial Intelligence (AI) field, called Machine Learning (ML). ML includes two modules: feature module and classification module. Feature module is responsible for extracting viable features such as textures and edges. Classification module classifies based on the extricated characteristics. Classification is a very common case of ML; classification algorithms are used to solve problems such as filtering email spam, categorization of documents, recognition of voice, recognition of images, and recognition of handwriting. The main drawback of ML is the need of large amount of training data which should be unbiased and of good quality. This is rectified by Deep Learning (DL).

DL is a sub-segment of ML, able to learn via its own computing method. DL uses complex, multi-algorithm framework represented as an artificial neural network (ANN). The ANN's architecture is replicated using the human brain's biological neural network. This makes DL more able than the standard models of ML. In DL, we consider the neural networks which recognize the image on the basis of its characteristics [1]. This is achieved by the construction of a complete characteristic extraction model which can solve the difficulties faced by conventional methods. In DL, to perform classification duties, a computer model learns directly from images, text, or sound. DL models, often exceeding human-level efficiency, may achieve state-of-the-art accuracy. Models are trained by the use of a large set of labeled data and architectures of neural networks that involve multiple layers.

14.1.1 Feature Extraction

The extraction of features is a reduction of dimensions where an initial set of unprocessed data is reduced to more convenient groups for processing. A feature of such big data sets is the great number of variables that require heavy computing resources for processing. Extraction of features is the term coined for methods that select and merge variables into characteristics, whilst decreasing the quantity of data to be processed and still accurately and completely describing the original unprocessed data.

14.1.2 Artificial Neural Network

In DL, a model masters at first hand from pictures, text images and audio to carry out classification task [2]. DL models can achieve pioneering precision, often exceeding output at the human level. DL models are trained using huge amount of labeled data and architectures of neural networks, which contain a lot of layers. ANN [3] is a value system of information processing, inspired by our own nervous system, such as information about the brain process as shown in Figure 14.1. ANNs are built like the human brain, with neuron nodes interconnected like a web.

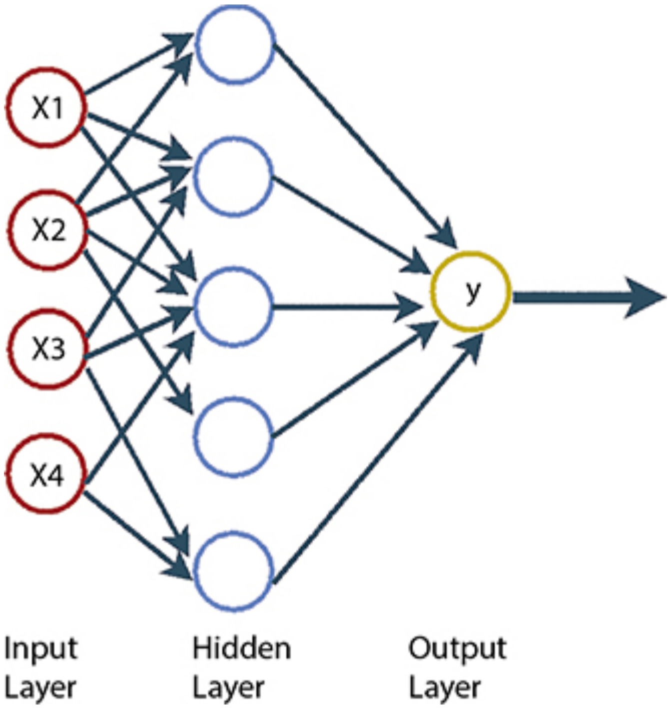

Figure 14.1 Artificial neural network.

An ANN has hundreds or thousands of artificial neurons that are interconnected by nodes, called processing units. Input and output units are made up of these processing units. Based on an internal weighting scheme, the input units obtain different types and structures of information and the neural network attempts to learn about the information provided to generate one output report. ANNs often use a series of learning rules called backpropagation to perfect their performance outcomes, just as humans need rules and instructions to come up with a result or performance. The ANN is taught what to look for and what its performance should be during the training and supervisory process, using yes/no query types of binary numbers. We train the network's data by providing an input image and conveying its output to the network.

14.1.3 Transfer Learning

Transfer learning [4] is a ML expression in which a model designed for a specific problem statement is reused for a second problem statement as the base model. In DL, it is a popular approach where pre-trained models are used because of their vast computational and time resources needed to construct neural network models on these problems and the enormous skill leaps they provide on related issues [5]. Transfer learning is an optimization technique, time saving shortcut or performance optimizer. The types of pre-trained model mainly used for effective image classification are vgg-16, vgg-19, and Xception.

Here, a network to classify different types of animals commonly found in the Indian scenario has been presented. The aim is to build a network having good accuracy and make it light-weight so that it becomes suitable for the real-time usage. A customized network, i.e., AnimNet has been developed using the transfer learning approach that is both light-weighted as well as provides a good accuracy. Section 14.2 describes the related work done recently. The remaining part of the chapter is organized as follows: The proposed methodology is illustrated in Section 14.3 which includes details of dataset preparation and model development. The results obtained from the proposed solution are described in Section 14.4 followed by conclusion in Section 14.5.

14.2 Related Work

A variety of work has been done to perform animal classification and detection. A few of them has been discussed here in brief. Authors in [6] have tried to solve the CAPTCHA challenge that is based on the issue of distinguishing dog and cat images. They have used Dense-SIFT features, combining dense SIFT and color features, and features learned from convolution neural network (CNN). They have achieved a good accuracy of 94% using support vector machine (SVM) classifier.

Authors in [7] have proposed a novel method of fully connected dual deep convolution neural network (DCNN), which extracts and analyzes the image features on a large scale. This method has gained the capability of analyzing a large amount of dataset as well as extracting more features than before including batch normalization layer and exponential linear unit (ELU) layer. The proposed DCNN outruns its counterparts as it has fewer computing costs and was able to achieve an accuracy of 92%. In [8], authors have tried to study household animals' demeanor and body language using DL technique. Their aim was to find out whether the animals are sick or not and provide necessary help in time if required. They have applied transfer learning on vgg16, and the accuracy increased from 80% to over 95%.

Authors in [9] have opted for a robust learning method for animal classification using images captured in an extremely cluttered natural environment and annotated with noisy marks. To divide the training samples based on different characteristics, they used k-means clustering, which was then used to train various networks. Two publicly accessible camera-trap image datasets were evaluated for the performance of the proposed method: Snapshot Serengeti [10] and the Panama-

Netherlands datasets. The results suggested that the approach selected by the authors outperformed the literature's state-of-the-art methods and enhanced the precision of the classification of animal species from camera-trap images which contain noise.

In paper [11], a classification system for classifying images of real animals has been developed by the authors. Using the toy photos of animals to account for factors other than just the physical appearance of animals, the model was educated. The segmentation was performed using the k-means clustering technique after pre-processing the image. Following segmentation, the extraction of hog features from the segmented image was performed. The extracted features were used in the final step to classify the image into a suitable class using the supervised multi-SVM classifier. In paper [12], the authors have shed light on different methods available for feature extraction and features these methods extract to perform an efficient identification and classification. They presented the results obtained from a dataset containing 111,467 photos in the training of a CNN to identify 20 African wildlife species with an overall accuracy of 87.5%. In order to generate a visual similarity dendrogram of known organisms, hierarchical clustering of feature vectors associated with each image has also been used.

In classifying camera trap data, how to process datasets with only a few classified images that are generally difficult to model and applying a trained model to a live online citizen science project, authors in [13] found the accuracy of DL. In order to distinguish between images of various animal species, human or vehicle images, and empty images, CNNs were used. Accuracies ranged between 91.2% and 98.0% for identifying empty images through programs, while accuracies ranged between 88.7% and 92.7% for identifying individual organisms.

Authors in [14] have attempted to resolve the challenge faced during CNN-based fine-grain recognition. Generally, the need for large sets of training data and the learned approaches to feature presentations are high-dimensional, leading to less efficiency. The authors suggested an approach where online dictionary learning is incorporated into CNN to resolve these issues. A significant amount of weakly labeled information on the Internet can be learned from the dictionary by an incremental process.

During fine-grained image classification, the authors discussed the problems of elevated inter-class similarity and broad intra-class variations in paper [15]. They also suggested a system of fine-grained image classification that performs identification of bird species. An online dictionary learning algorithm has been proposed where the concept of sparsity is incorporated into the use of bilinear convolutional neural network (BCNN) classification. This method performs classification based on sparsity, where a lower number of dictionary atoms can reflect training data.

Although there are existing networks, but they are not lightweight that can be deployed on edge devices. AnimNet network, that is accurate and lightweight than the existing networks has been developed and is the key contribution of this work. AnimNet network is five times lighter than Xception model and can accurately classify different types of animals commonly found in the Indian scenario.

14.3 Proposed Methodology

The proposed approach consists of data collection measures, preparation, and performance assessment of models. Each procedure is therefore outlined briefly as follows.

14.3.1 Dataset Preparation

The dataset has been self-created which has six categories of animals commonly found in India. The six categories of animals include cat, cow, dog, horse, goat, and monkey. The images were captured in very common circumstances. The samples for each class are taken from different scenarios and has high-resolution as well as low-resolution images. These images were captured from raspberry pi and Mi Note 8 Pro mobile phone to achieve images with a combination of high and low resolutions. The dataset is not only focused on the images of the animals in their natural habitat but also contains pictures of animals in common surroundings. For example, the image of horses is from the barn, roads, and side of roads to have more natural and common surrounding areas. The dataset has been divided into two sets, i.e., training images and testing images containing 2,519 and 629 images, respectively.

14.3.2 Training the Model

We have used two approaches in building a classification model for animals in Indian scenario: 1) using transfer learning and 2) creating a custom model (AnimNet).

1) *Using Transfer Learning:* We have trained our dataset using available pre-trained models, i.e., VGG19, VGG16, and Xception. In this section, we will look into the structure of each of these models and perform training on the custombuilt dataset. Using transfer learning, we first train a base network on a base dataset, and then transfer the learned features to a second

target network to be trained on a target dataset. If the features are general, this mechanism would appear to perform, meaning that they are appropriate for both tasks, rather than unique to the base task. Here, we use pre-trained weights and retrain the network using the custom dataset which increases the accuracy. The analysis of the obtained results is provided in the later section.

VGG16 [16] is a model developed by K. Simonyan and A. Zisserman. ImageNet [17] is a dataset of over 14 million images in 1,000 classes, with a test accuracy of 92.7 %. The input to the cov1 layer is an image of fixed size 224 × 224. The image is executed through a stack of convolutional layers and the filter was used with less receptive fields: 3 × 3 which is the smallest size that captures the up/down, center, left/right. The padding layers are one pixel long. There are five layers of max pooling layer following the convolution layer. Max-pooling is achieved using a 2 × 2-pixel window with a stride of 2. The design of the fully connected layer is similar to other networks. Non-linearity of the rectification (ReLU) is fitted on all the hidden layers. None of the networks provide Local Response Standardization (LRN), except for one, such standardization does not boost the ILSVRC dataset efficiency, but results in increased memory usage and computation time.

In Figure 14.2, the architecture of VGG16 and VGG19 has been presented. The difference between both the architecture is that VGG16 has 16 layers of 3 × 3 convolution and VGG19 has 19 layers of 3 × 3 convolution. It means that VGG16 contains 16 layers that has some weights, whereas VGG19 contains 19 layers that has some weights. VGG16 is a large network which has a total of about 138 million parameters. In terms of the number of parameters to be trained, it is really large. VGG19 network, which is bigger than VGG16, but because VGG16 does almost as well as the VGG19, a lot of people will use VGG16. But, in order to achieve more accuracy, VGG19 is preferred in some cases.

VGG19 [18] is a convolution neural network with 19 layers. ImageNet database has a pre-trained version of the network on a million images. The pre-trained network has learned accurate attribute representations for a large image dataset. Its frame size is 224 × 224. A fixed-size RGB image of size 224 × 224 has been inputted into this network. This suggests the matrix would have been (224,224,3). The mean RGB value was subtracted from each pixel. Kernels of 3 × 3 size with a step size of 1 pixel were used which made covering the whole notion of image possible. Spatial information of the image was retained with the help of spatial padding. A 2 × 2-pixel window with stride 2 was used to perform max pooling. Rectified linear unit (ReLU) was used after this to add non-linearity. It is better than models using tanh or sigmoid functions as non-linearity increases the computational time. Three fully connected layers were implemented: the first two layers have 4,096 channels each, the third layer performs ILSVRC 1,000-way classification and thus has 1 channel per 1,000 channel class. Softmax is applied in the last layer for classification.

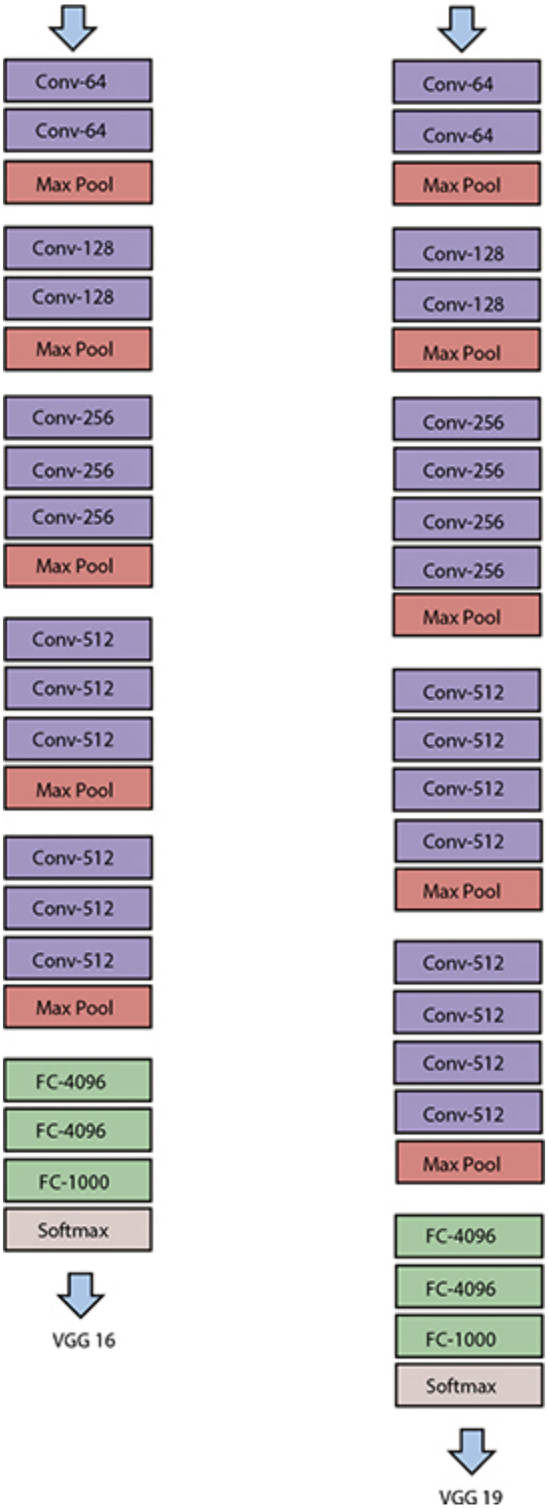

Figure 14.2 Layers of VGG16 and VGG19 network.

Xception [19] is a convolutional, 71-layer, deep neural network. It is a CNN architecture that relies completely on depth-wise separable convolution layers. A pre-trained version of the trained network can be loaded from the ImageNet database which has over a million images. As a result, the network has learned rich representations of features for a wide array of images. The network size is 299 × 299 input images. It is a convolutional architecture of the neural network, formed entirely on depth-wise, separable layers of convolution. Later, the following hypothesis was proposed that in the feature maps of CNN the mapping of cross-channels correlations and spatial correlations can be completely decoupled. As this hypothesis is a firmer version of the Inception architecture hypothesis, this architecture was coined the term "Xception" which stands for "Extreme Inception". The architecture consists of 36 convolutional layers which form the network's base for extracting features. These 36 layers are arranged into fourteen modules, all except for the first and last modules, which have residual linear ties around them.

2) *Creating a Custom Model (AnimNet):* The main aim to develop a custom model is to have a good performance model that is lightweight to make it suitable for mobile devices. AnimNet required no pre-trained weights for training. This model has been trained from scratch by adjusting the weights, adding customized layers and tried to make the proposed model as light as possible with enough neurons for efficient feature extraction along with a good accuracy. The convolution is followed by max pooling and then dropout of 0.5 has been added. The input shape is (128 × 128 × 3), kernel is of (3 × 3), and l2 regularizer of 0.01 has been used. After this, flattening is done followed by adding dense layers and dropout. Rmsprop has been used as optimizer, ReLU as activation function, and categorical cross-entropy has been used for the training. The typical classification and localization network architecture has been shown in Figure 14.3 with an additional regression head on the top right with CNN classification network.

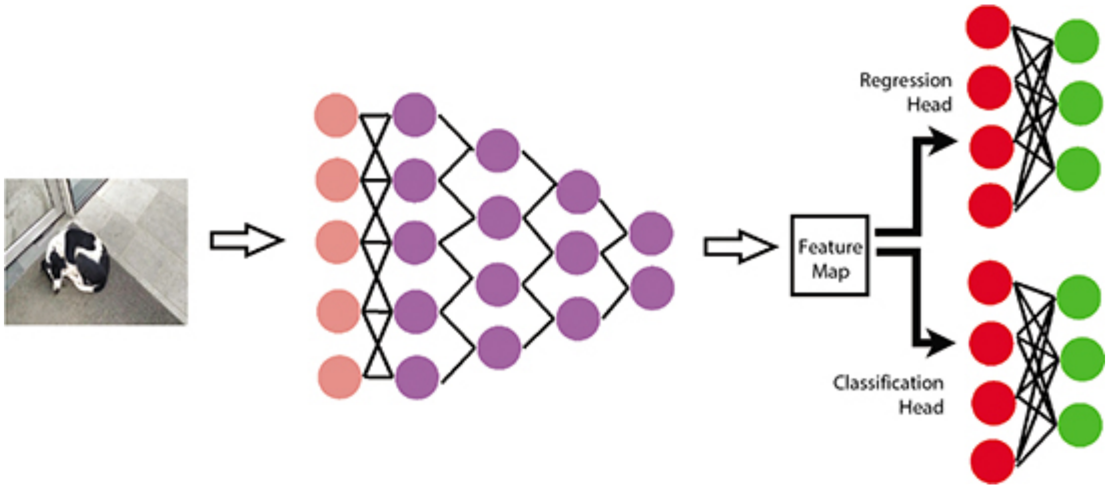

Figure 14.3 The architecture of convolution neural network.

14.4 Results

The result achieved by applying transfer learning and by AnimNet has been presented in this section. The training, validation, and testing performance of the models has been shown in graphs and tables. The training loss, training accuracy, validation loss, validation accuracy, and size of the network are the performance evaluation parameters which we have used in this chapter to discuss the performance of networks. The loss and accuracy achieved during training and validation process for each of the networks has been compared to analyze their performance. Along with the loss and accuracy, size of each networks has been compared to understand their compatibility with mobile devices. These evaluation parameters are discussed briefly below:

- Training Loss: The loss encountered at each epoch during the training process which tends to be decreasing for a good model is called as training loss.
- Training Accuracy: The accuracy achieved at each epoch during the training process which tends to be increasing for a good model is called as training accuracy.
- Validation Loss: The loss encountered during the validation process which is done after the training process and before the testing process is called as validation loss.
- Validation Accuracy: The accuracy achieved during the validation process which is done after the training process and before the testing process is called as validation accuracy.
- Size: The overall size of the network after the training and validation process is complete. A network with lower size is the preferred choice.

14.4.1 Using Pre-Trained Networks

As can be seen in Table 14.1, the model with highest accuracy (training - 89% and validation - 92%) and minimum loss (training - 0.354 and validation - 0.0862) among all is Xception model. There is no problem of overfitting or under-fitting has been observed; hence, it is the best choice to go for when size of the model is not an issue. It has lowest validation loss as well as training loss. The highest loss was encountered when we have trained using VGG19 network for training.

The loss encountered during training and validation has been shown by a plot in Figure 14.4a and the accuracy achieved during training and validation has been shown by a plot in Figure 14.4b. It can be clearly observed that the validation loss is lower than the training loss and the validation accuracy is higher than the training accuracy.

14.4.2 Using AnimNet

A self-created dataset, customized layers, and adjusted neurons have been used for custom network development. The training process took a longer time than the time taken by training using pre-trained model. AnimNet was trained using conv2D and dense layers, with less neurons as compared to pre-trained model to make it lightweight. Although the accuracy achieved using Xception model is higher than AnimNet network, but this network is light-weight of 16mb which is five times lighter than Xception model as shown in Table 14.2. One of the major reason of this is the dataset which we have used had limited number of images (≤3,200).

Table 14.1 Results obtained using pre-trained networks.

Pre-trained network	Training loss	Training accuracy	Validation loss	Validation accuracy	Size (mb)
VGG16	0.6125	0.7345	0.5326	0.8231	89
VGG19	1.0318	0.6748	0.9873	0.8154	93
Xception	0.354	0.8849	0.0862	0.9201	87

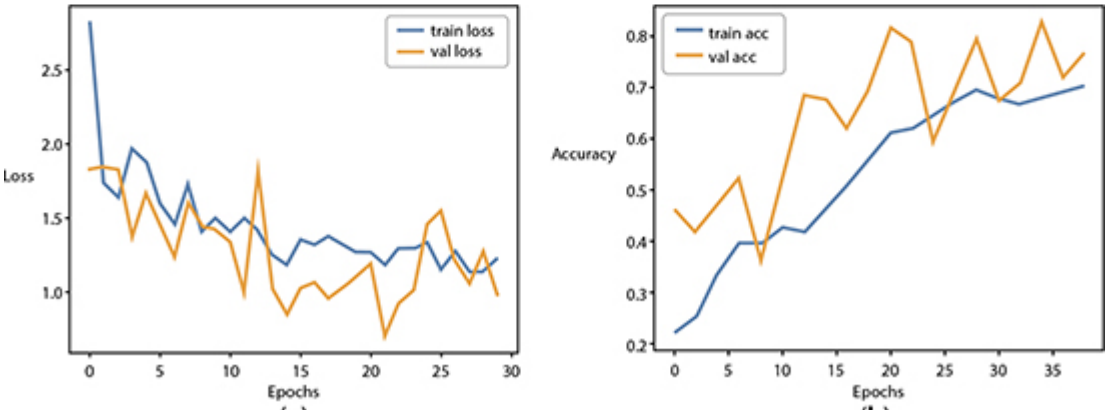

Figure 14.4 Plots of Xception network. (a) Training loss of Xception network (b) accuracy of Xception network.

Table 14.2 Results obtained using AnimNet network.

Training loss	Training accuracy	Validation loss	Validation accuracy	Size (mb)
1.2617	0.7948	0.6797	0.8467	16

The loss and accuracy of the AnimNet network during training and validation has been shown in form of a graph in Figures 14.5a and b. It has been observed that the validation loss reached below 1, whereas the training loss is more than that. Also, the validation accuracy is higher than the training accuracy which is clearly visible in the graph.

14.4.3 Test Analysis

In this section, the qualitative performance analysis of Xception (pretrained) and AnimNet (custom-built) network has been shown on the test data. The difference in testing accuracy obtained by using pre-trained network, i.e., Xception and AnimNet network, has been shown. The classification result is shown class-wise, i.e., for six classes. The time taken for classification of each image by both the networks are almost same, i.e., 9–11 ms/step. By looking over the validation accuracy obtained by these two models, it can be assumed that the Xception model would provide a better accuracy while predicting images from our test dataset. Although Xception network achieved the highest accuracy, the accuracy of AnimNet is also acceptable. However, if the need is to have a lightweight as well as accurate model, then the proposed network (AnimNet) may be preferred.

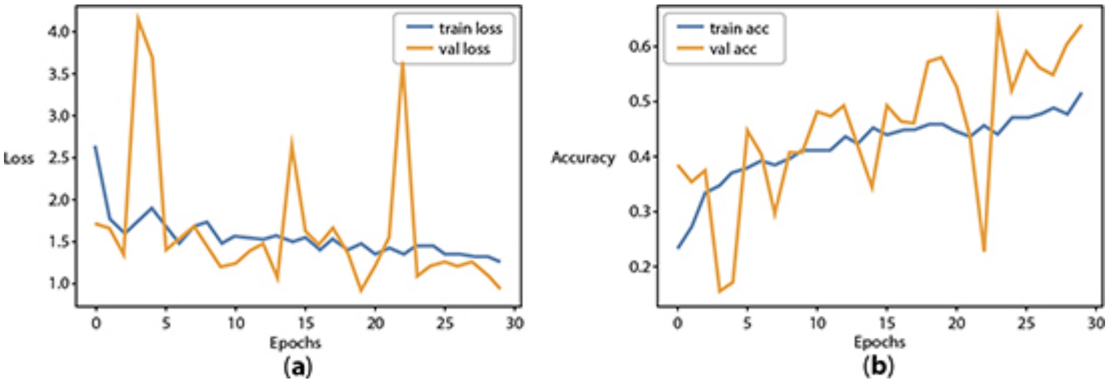

Figure 14.5 Plots of AnimNet network. (a) Training loss of AnimNet network (b) accuracy of AnimNet networks.

The test results of the model on the test images along with the accuracy obtained for each class has also been shown. The test results obtained on test images for Xception and AnimNet (custom-built) network has been shown in Figures 14.6, 14.7, 14.8, 14.9, 14.10, and 14.11, respectively. These results are after training all the models for the self-created dataset. It has been observed from the test results that the accuracy was higher when Xception model was used as it is pre-trained on large dataset. Although AnimNet network achieved less accuracy than Xception, it can also be used where size of the model is a constraint. The Xception model is quite heavy, i.e., 87 mb which makes it five times heavier than AnimNet which is of 16 mb size. As per the requirement and need, one can choose either Xception or AnimNet network.

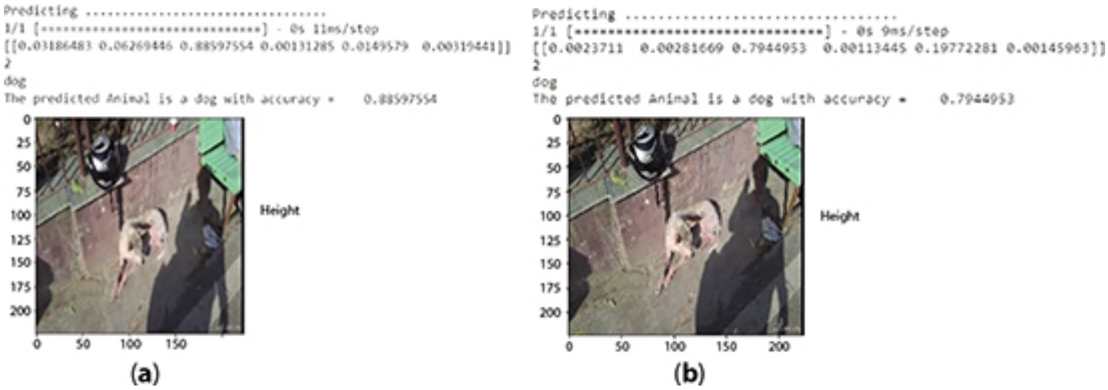

Figure 14.6 Accuracy of class "dog" (size of image = 200*200 pixels). (a) Using Xception network (b) using AnimNet network.

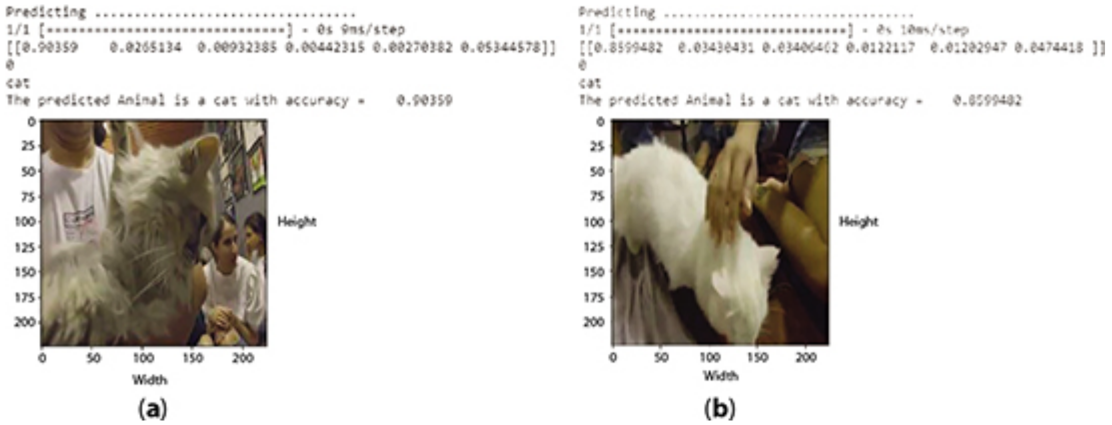

Figure 14.7 Accuracy of class "cat" (size of image = 200*200 pixels). (a) Using Xception network (b) using AnimNet network.

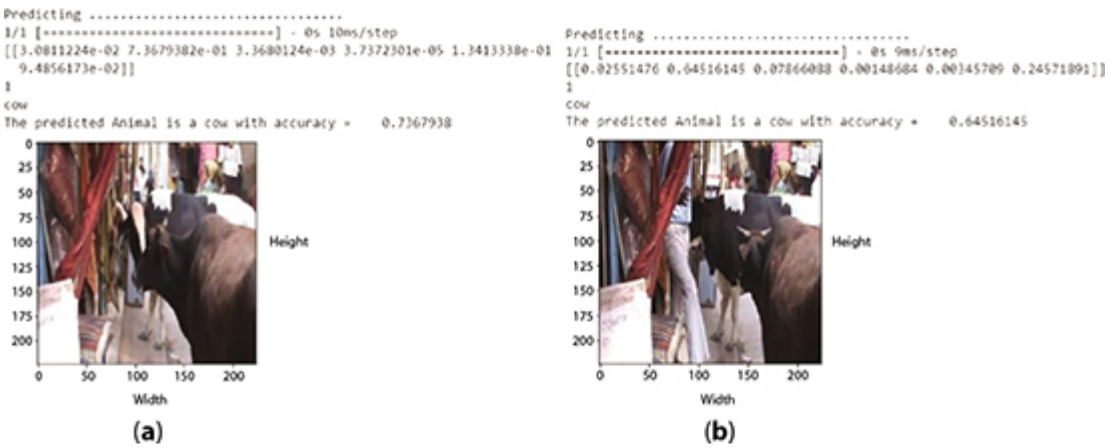

Figure 14.8 Accuracy of class "cow" (size of image = 200*200 pixels). (a) Using Xception network (b) using AnimNet network.

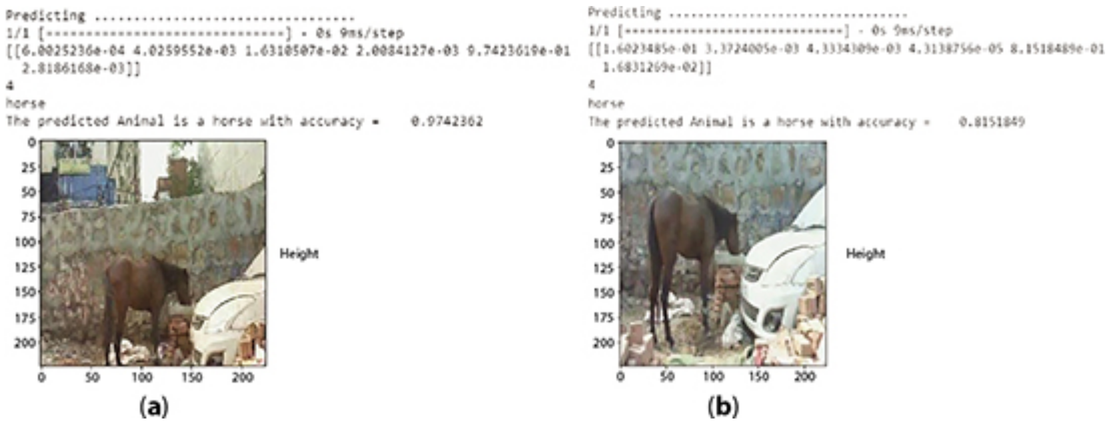

Figure 14.9 Accuracy of class "horse" (size of image = 200*200 pixels). (a) Using Xception network (b) using AnimNet network.

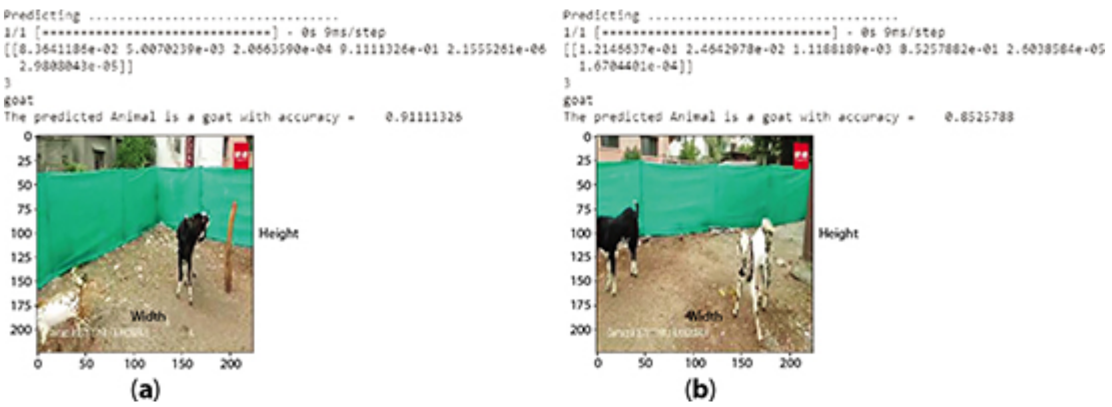

Figure 14.10 Accuracy of class "goat" (size of image = 200*200 pixels). (a) Using Xception network (b) using AnimNet network.

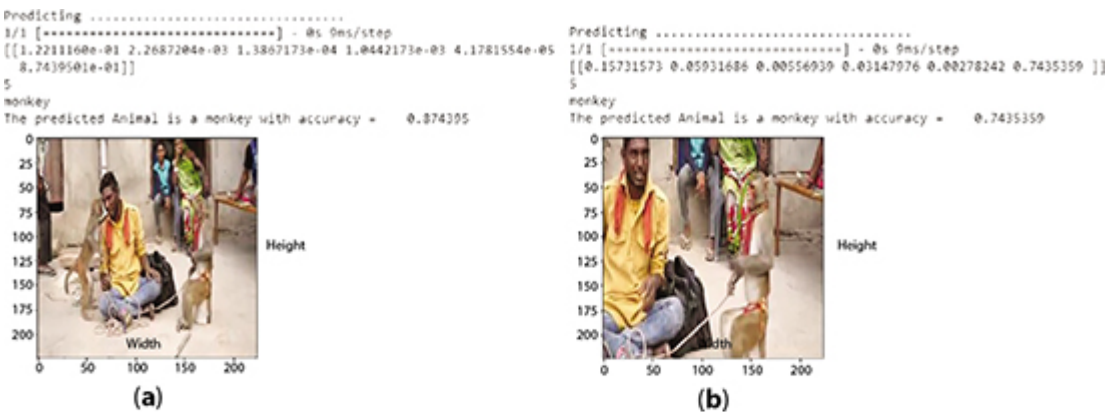

Figure 14.11 Accuracy of class "monkey" (size of image = 200*200 pixels). (a) Using Xception network (b) using AnimNet network.

14.5 Conclusion

A quantitative as well as qualitative performance analysis of the pre-trained networks and AnimNet has been provided in this chapter for clear understanding. As can be observed from result analysis, the accuracy given by the AnimNet network is very well though the images have many things other than animals like human and vehicles. A demonstration of how a technology called deep neural networks is capable of extracting the valuable features that can classify a plethora of real-life objects including internal human body tumors as well. It was observed that Xception model gave us a better testing accuracy than the AnimNet for different classes of animals. A point can also be concluded that this AnimNet network was good at predicting test images from dataset, which is acceptable and good to go as our classification was focused on animals commonly found in Indian scenario. AnimNet can be used for classification where it is aimed to be used for edge devices that are resource-constrained. A pre-trained model would always give us better accuracy as they are trained on a huge dataset, when it comes to feature extraction, nevertheless were able to achieve a good accuracy with AnimNet network in limited time period. Also, the size of the AnimNet network is five times lighter than Xception model. So, the selection of network is clearly up to the choice of end user. As a future work, AnimNet network can be improvised to have higher accuracy than all the pre-trained networks but not at the cost of size.

References

1. Manjari, K., Verma, M., Singal, G., A Survey on Assistive Technology for Visually Impaired. *Internet Things*, Elsevier, 11, 100188, 2020.

2. Manjari, K., Verma, M., Singal, G., A Travel Aid for Visually Impaired: R-Cane, in: *International Conference on Smart City and Informatization*, 2019, November, Springer, Singapore, pp. 404–417.

3. Jain, A.K., Mao, J., Mohiuddin, K.M., Artificial neural networks: A tutorial. *Computer*, 29, 3, 31–44, 1996.

4. Torrey, L. and Shavlik, J., Transfer learning, in: *Handbook of research on machine learning applications and trends: algorithms, methods, and techniques*, pp. 242–264, IGI Global, New York, 2010.

5. Manjari, K., Verma, M., Singal, G., CREATION: Computational ConstRained Travel Aid for Object Detection in Outdoor eNvironment, in: *2019 15th IEEE International Conference on Signal-Image Technology & Internet-Based Systems (SITIS)*, 2019, November, pp. 247–254, 2020.

6. Suryawanshi, M.S., Jogdande, M.V., Mane, M.A., Animal classification using Deep learning. *Int. J. Eng. Appl. Sci. Technol*., 6, 305–307, 2020.

7. Khan, R.H., Kang, K.W., Lim, S.J., Youn, S.D., Kwon, O.J., Lee, S.H., Kwon, K.R., Animal Face Classification using Dual Deep Convolutional Neural Network. *J. Korea Multimed. Soc*., 23, 4, Korea multimedia society, 525–538, 2020.

8. Lin, L. and Link, Y., *Household Animals Classification Using Deep Learning*, CS230: Deep Learning, Winter 2020, Stanford University, CA, 2020.

9. Ahmed, A., Yousif, H., Kays, R., He, Z., Animal species classification using deep neural networks with noise labels. *Ecol. Inf*., 57, 101063, 2020.

10. Swanson, A., Kosmala, M., Lintott, C., Simpson, R., Smith, A., Packer, C., Snapshot Serengeti, high-frequency annotated camera trap images of 40 mammalian species in an African savanna. *Sci. Data*, 2, 1, 1–14, 2015.

11. Nanditha, D. and Manohar, N., Classification of Animals Using Toy Images, in: *2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS)*, IEEE, pp. 680–684, 2020, May.

12. Miao, Z., Gaynor, K.M., Wang, J., Liu, Z., Muellerklein, O., Norouzzadeh, M.S., Getz, W.M., Insights and approaches using deep learning to classify wildlife. *Sci. Rep*., 9, 1, 1–9, 2019.

13. Willi, M., Pitman, R.T., Cardoso, A.W., Locke, C., Swanson, A., Boyer, A., Fortson, L., Identifying animal species in camera trap images using deep learning and citizen science. *Methods Ecol. Evol*., Wiley, 10, 1, 80–91, 2019.

14. Srinivas, M., Lin, Y.Y., Liao, H.Y.M., Learning deep and sparse feature representation for fine-grained object recognition, in: *2017 IEEE International Conference on Multimedia and Expo (ICME)*, IEEE, pp. 1458–1463, 2017, July.

15. Srinivas, M., Lin, Y.Y., Liao, H.Y.M., Deep dictionary learning for finegrained image classification, in: *2017 IEEE International Conference on Image Processing (ICIP)*, IEEE, pp. 835–839, 2017, September.

16. Alippi, C., Disabato, S., Roveri, M., Moving convolutional neural networks to embedded systems: the alexnet and VGG-16 case, in: *2018 17th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, pp. 212–223, 2018, April.

17. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L., Imagenet: A largescale hierarchical image database, in: *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255, 2009, June.

18. Carvalho, T., De Rezende, E.R., Alves, M.T., Balieiro, F.K., Sovat, R.B., Exposing computer generated images by eye's region classification via transfer learning of VGG19 CNN, in: *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*, 2017, December, pp. 866–870.

19. Chollet, F., Xception: Deep learning with depthwise separable convolutions, in: *Proceedings of the IEEE*, 2017.

**Corresponding author*: KM5723@bennett.edu.in