Materials Today: Proceedings xxx (xxxx) xxx

Contents lists available at ScienceDirect

Materials Today: Proceedings

journal homepage: www.elsevier.com/locate/matpr

Animal species image classification

Likhith Prudhivi *, Narayana M., Subrahmanyam Ch, Gopi Krishna M.

Vardhaman College of Engineering, Hyderabad, Telangana 501218, India

ARTICLE INFO

Article history:
Received 4 February 2021
Accepted 26 February 2021
Available online xxxx

Keywords:
Computer vision
Convolutional neural network
Bottleneck features
Deep learning architectures

ABSTRACT

Animal species image classification is used in forests to classifiy animals in real time. In past, many computer vision techniques were introduced but they couldn't fulfill the requirements as the accuracy got depreciated since the technology advanced. But as per the requirement many techniques were introduced where accuracy got drastically improved where we could perform the image classification, image recognition and segmentation. This project aims to introduces efficient technique for animal species image classification with the goal of achieving good amount of accuracy. Convolutional neural network is been engineered for the image classification process. Bottleneck features are trained and synched to the pretrained architecture to achieve high accuracy. Numerous deep learning architectures are compared with the dataset.

Selection and peer-review under responsibility of the scientific committee of the Emerging Trends in Materials Science, Technology and Engineering.

1. Introduction

Wildlife monitoring is crucial for tracking animal habitat utilization, population demographics, poaching incidents, and movement patterns. Various technologies has been introduced for monitoring wild animals [1 2]. Efficient and reliable monitoring of wild animals in their natural habitats is essential to inform conservation and management decisions regarding wildlife species [3]. Image classification is the process of categorizing and labeling groups of pixels or vectors within an image based on specific rules [4].CNN is a deep-learning (DL) algorithm which take an image as input, process the image by performing operations like convolution, pooling, normalization and obtain the output. CNN's are used for image classification and recognition because of its high accuracy [3]. This paper concentrates on using deep learning pre trained architecture as a model and transfer learning is performed to sync our dataset with the architecture.

The pre trained architectures such as VGG16, Resnet50, InceptionV3, mobilenet are used. Initially, the dataset is divided into train, test and validation. The model is trained using the train and validation datasets. After training, we test the dataset using test dataset. Based on the results during training, the accuracy graphs and loss graphs are plotted and analysed. Finally, the confusion matrix is plotted from the test dataset results. The accuracy can determined from the confusion matrix. The process is same for all architectures. In results section, the performance of each model is compared.

The outline of this paper is as follows: In section 2, some existing related work will be discussed. In section 3, methodology will be discussed in brief. In section 4, results along with the discussion related to them also comparative analysis. In section 5, conclusion of the research.

2. Related work

Currently, deep-learning is used in various fields and research work. The leading common Convolutional neural nets (ConvNets) for the popularity and image-classification area unit the subsequent. AlexNet [3] is a big breakthrough in computer-vision. Network-in-Network (NIN) proposed in [5] was one in all the initial and important models, during which 1x1 Conv's were enforced, so as to produce additional functionality to the options of the Conv-layers. Most well liked deep learning architectures are GoogleNet, AlexNet, VGG, ResNet, NiN, and inception-1, 2, three that are utilized within the classification of images [6].VGG16 has 16 convolutional layers, bunch of max-pooling layers, and 3 final FC (fully-connected) layers. GoogLeNet model was introduced to be economical in computation, it offers high accuracy [7]. Residual learning architecture of ResNet in [8] has obtained good results by associating output of Conv-layers and their corresponding original input. Dense Conv-Network (DenseNet) [9] has raised

* Corresponding author.
E-mail address: likhith.prudhivi@gmail.com (L. Prudhivi).

https://doi.org/10.1016/j.matpr.2021.02.771
2214-7853/
Selection and peer-review under responsibility of the scientific committee of the Emerging Trends in Materials Science, Technology and Engineering.

the performance of classification-task by connecting each and every layer to different layer in a feed-forward manner. In [10] various deep learning architectures like AlexNet, NiN, VGG, ResNet-18, 34, 152 were to differentiate animals. Among the 6 architectures, VGG has provided good accuracy of 96.8% [10]. In our paper, deep learning architectures like VGG16, Resnet50, InceptionV3 and MobileNet are used classify animal species. Among all the four architectures InceptionV3 provided highest accuracy of 95% then later comes mobilenet with 93%, then comes VGG16 with 87% finally Resnet50 with 53%.This paper concentrates on the VGG16 design and therefore the same procedure is employed for different architectures.

3. Methodology

Data Augmentation: Later the train, validation, test datasets are sent to data augmentation process and transforms into generators. Generally, the original dataset may not be sufficient for training process to achieve good amount of accuracy hence, we perform data augmentation process to our dataset. The operations such as flipping, rotation, translation, rotation, cropping, geometric transformations and color space transformations are performed and many images are derived from a single image and hence, dataset gets extended Fig. 1.

Model Architecture: There are many pre-trained architectures in deep learning. Among them, VGG16 design is employed during this paper at first sequential model is formed. Then the VGG16 design is synced to the model. The VGG16 is a (CNN) design that won Imagenet competition in 2014. It's one of the best computer vision model design until date. VGG16 is that rather than having an outsized variety of hyper-parameter they targeted on having 3x3 kernel with a stride one convolution layers and forever used same padding and maxpool layer of a pair of 2x2 kernel of stride two. The convolutional and polling layers are systematically arranged throughout the entire design. Finally, the pair of fully connected layers followed by a soft-max function for the output. VGG16 architecture is demonstrated in Fig. 2.

In Convolutional 2D layer, a kernel is a Small matrix. It is used to perform the operations such as blurring, sharpening, edge detection, and more. This is achieved by performing a convolution between a kernel and the query image. The output consists of features extracted from the previous layer. For each convolution layer the channels are increased. The no of output features in each convolution operation is represented as:

$$n_{out} = \left[\frac{n_{in} + 2p - k}{s}\right] + 1 \quad (1)$$

n_{in} : number of input features, n_{out} : number of output features, k : convolution kernel size, p : convolution padding size, s : convolution stride size

Maximum pooling, or max pooling, computes the maximum value in each feature map. The results are down scaled pooled feature maps which represents maximum features of the previous layer. For every max pooling layer the scale of the image is attenuated.

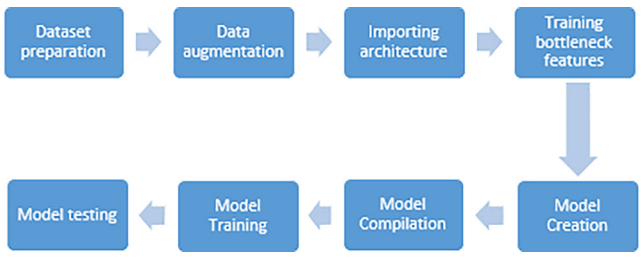

Fig. 1. The block diagram of the methodology.

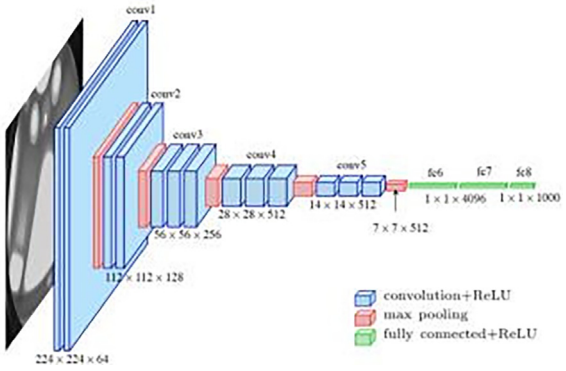

Fig. 2. 3D representation of VGG16 Network.

Table 1
Dataset samples division.

Species name	Total count	Train count	Validation count	Test count
Butterfly	1000	800	100	100
Cat	1000	800	100	100
Cow	1000	800	100	100
Dog	1000	800	100	100
Panda	1000	800	100	100
Sheep	1000	800	100	100
Squirrel	1000	800	100	100

Table 2
Deep learning architectures training results.

Model	Accuracy	Loss
VGG16	0.86	0.41
RESNET50	0.53	1.26
INCEPTIONV3	0.95	0.18
MOBILENET	0.93	0.24

Final layers of the model are dense layer with leaky-relu activation function and dense layer with softmax activation function. The leaky-relu activation function is represented as:

$$f(x) = \begin{cases} x & if \quad x > 0 \\ 0.01x & otherwise \end{cases} \quad (2)$$

ReLU is the most widely used activation function But, in our model both activation functions (ReLU and Leaky-ReLU) gave the same output. We can consider either of the activation functions. In this paper Leaky-ReLU is used to train the model.

The softmax function is represented as:

$$f_j(x) = \frac{\exp(x_i)}{\sum_j \exp(x_j)} \quad (3)$$

Since there are 7 output layers we are using soft-max function. If there are 2 output layers, then the sigmoid function is used.

Bottleneck Features: To perform transfer learning, the FC-layer is detached from the model and therefore the desired layers are plugged. These are called bottleneck features. The reduced model output features that will fill the model.

Model compilation and Training: During compilation, RMSPROP optimizer is employed. Instead The RMSPROP optimizer is analogous to the gradient descent algorithm with momentum. Categorical cross entropy is employed as loss function. Model is

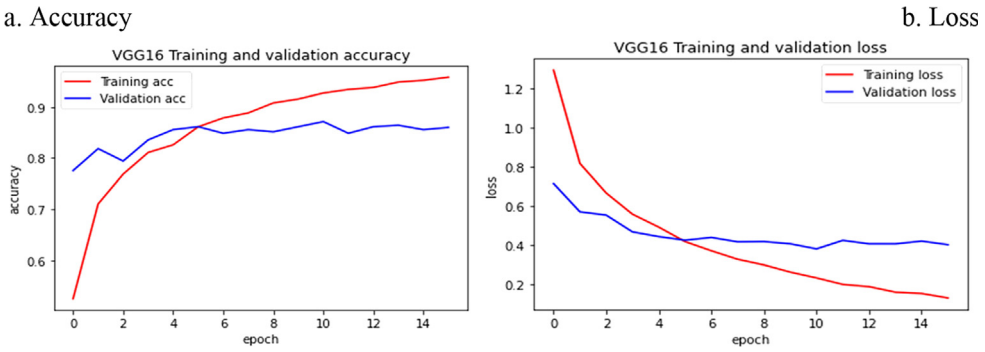

Fig 3.1. (a) Training accuracy and (b) loss of VGG16 model.

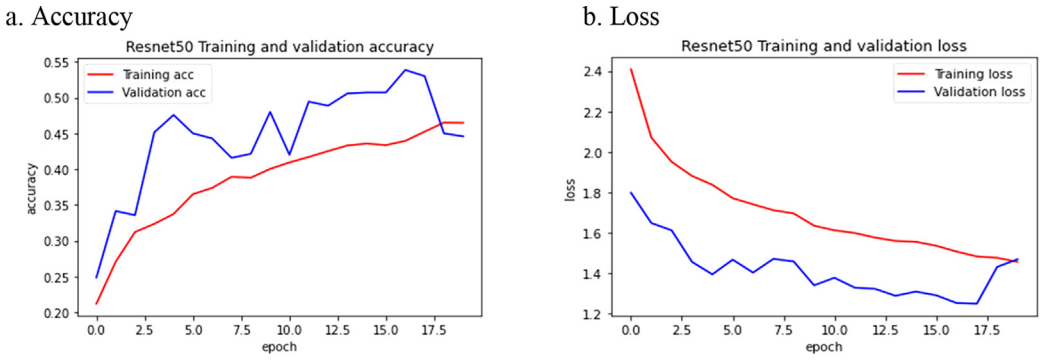

Fig 3.2. (a) Training accuracy and (b) loss of resnet50 model.

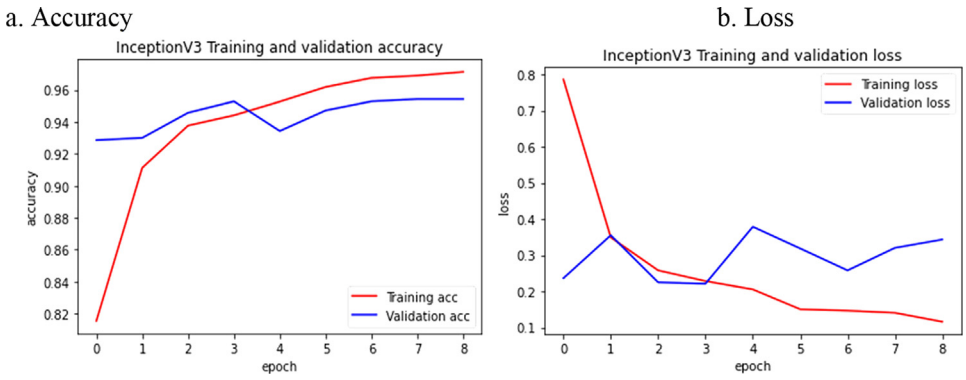

Fig 3.3. (a) Training accuracy and (b) loss of InceptionV3 model.

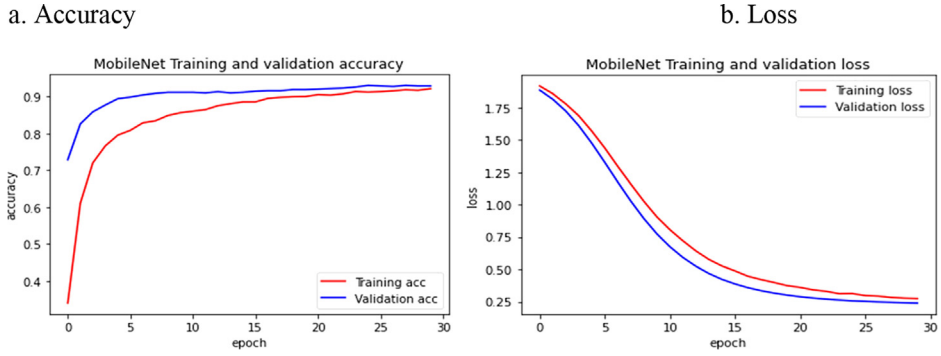

Fig. 3.4. (a) Training accuracy and (b) loss of Mobilenet model.

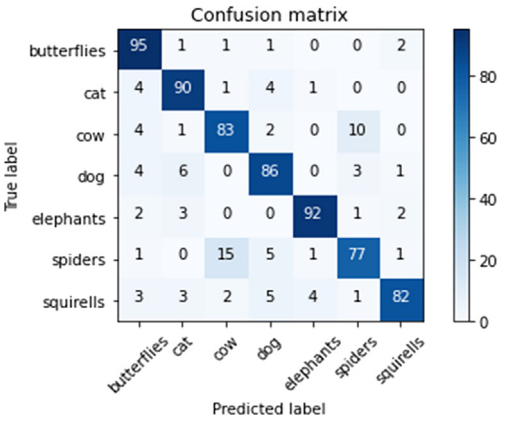

Fig 4. VGG16 Confusion Matrix.

trained for twenty epochs with train generator and validation generator. The weight and bias are Adam and RMSPROP converge quicker than GD or SGD, they have higher native minima. This paper concentrates on RMSPROP as there's no huge distinction within the output whether or not the optimizer is Adam or RMSPROP.

$$vdw = \beta * dw + (1 - \beta) * dw^2 \tag{4}$$

$$vdb = \beta * dw + (1 - \beta) * db^2 \tag{5}$$

$$W = W - \alpha * \frac{dw}{\sqrt{vdw} + \varepsilon} \tag{6}$$

$$b = b - \alpha * \frac{db}{\sqrt{vdb} + \varepsilon} \tag{7}$$

Adam and RMSPROP converge faster than GD or SGD, they have better local minima. This paper concentrates on RMSPROP as there is no big difference in the output whether the optimizer is Adam or RMSPROP.

Model Evaluation: The training was performed for seventeen epochs after, it terminated due to the presence of early stopping callback. The training accuracy kept on rising but the validation

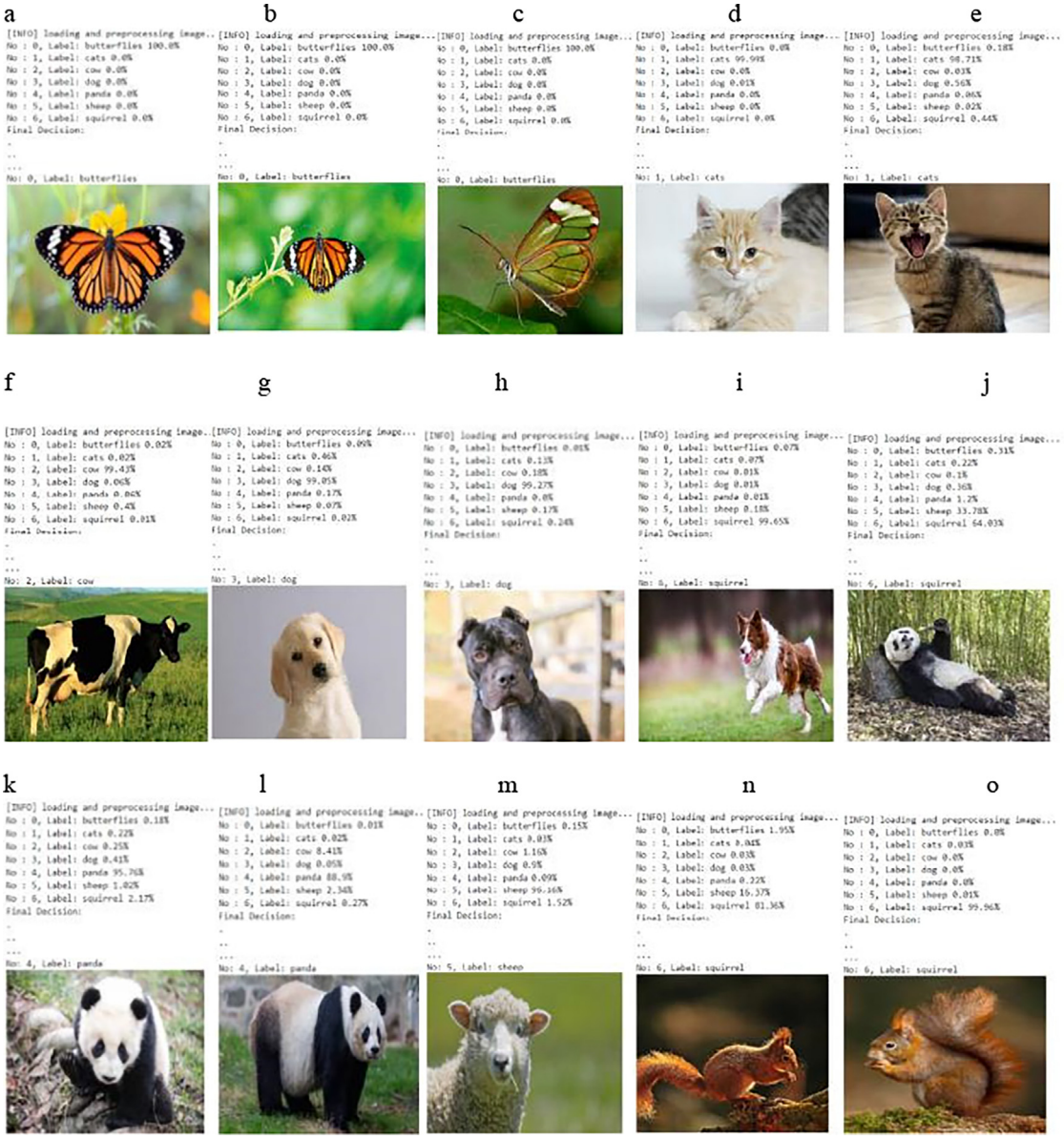

Fig 5. (a – o) Prediction of Sample images.

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 222, 222, 32)      896
_________________________________________________________________
batch_normalization (BatchNo (None, 222, 222, 32)      128
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 111, 111, 32)      0
_________________________________________________________________
dropout (Dropout)            (None, 111, 111, 32)      0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 109, 109, 64)      18496
_________________________________________________________________
batch_normalization_1 (Batch (None, 109, 109, 64)      256
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 54, 54, 64)        0
_________________________________________________________________
dropout_1 (Dropout)          (None, 54, 54, 64)        0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 52, 52, 128)       73856
_________________________________________________________________
batch_normalization_2 (Batch (None, 52, 52, 128)       512
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 26, 26, 128)       0
_________________________________________________________________
dropout_2 (Dropout)          (None, 26, 26, 128)       0
_________________________________________________________________
flatten (Flatten)            (None, 86528)             0
_________________________________________________________________
dense (Dense)                (None, 512)               44302848
_________________________________________________________________
batch_normalization_3 (Batch (None, 512)               2048
_________________________________________________________________
dropout_3 (Dropout)          (None, 512)               0
_________________________________________________________________
dense_1 (Dense)              (None, 7)                 3591
=================================================================
Total params: 44,402,631
Trainable params: 44,401,159
Non-trainable params: 1,472
```

Fig. 6. Summary of model without Pre trained Architectures.

accuracy has dropped with fluctuations due to overfitting. To prevent overfitting only 17 epochs were performed out of 20. In case of Adam optimizer, 20 epochs were performed instead of 17 but there is no considerable change in the result.

4. Results and discussions

4.1. Dataset description

The Dataset consists of the images of the animals obtained from the internet. The dataset comprises of the images of various animals like butterfly, cat, cow, dog, panda, sheep and squirrel. The dataset has total 7000 images which is divided into train, validation and test datasets. 1000 images per animal and hence 1000x7 images. Then in 1000x7 images 800x7 are categorized as trained images, 100x7 are validation images and 100x7 are test images is demonstrated in Table 1

After training the model with the dataset with various models the InceptionV3 achieved highest accuracy (from Table 2).

4.2. Accuracy graph and loss graphs

The training accuracy kept on rising and validation accuracy turned constant at a point where train and validation graphs intersect Fig. 3.1(a).

The training loss kept on decreasing and validation loss turned constant at a point where training and validation graphs intersect Fig. 3.1(b).

4.3. Other graphs

The resnet50 architecture accuracy and loss graphs are observed in Fig. 3.2(a) and (b)

The inceptionV3 architecture accuracy and loss graphs are observed in Fig. 3.3(a) and (b).

The Mobilenet architecture accuracy and loss graphs are observed in Fig. 3.4(a) and (b).

4.4. Confusion matrix

Confusion matrix shows the validation of the model with test images. It also helps to determine the accuracy by visualizing the true positives, negatives also false positives and negatives. Hence, the performance can also be determined. Fig. 4 is the confusion matrix obtained from the VGG-16 architecture.

The results of sample images are shown in Fig. 6. All the sample images were classified with more than 85% Accuracy. Fig. 5 are sample results obtained from the model.

Out of all samples (i) and (j) predicted incorrectly. (i) is a dog but the model predicted it as a squirrel and (j) is a panda but the model predicted it as a squirrel.

4.5. Comparative analysis

We have also implemented the project without performing transfer learning by convolution layers and pooling layers. The summary of the model is represented in Fig. 6.

After training this model with the same dataset the accuracy has been decreased compared to the model trained by performing transfer learning. The accuracy and loss graphs are represented in Fig. 7(a) and (b).

This model has obtained the accuracy of 30% which is very less compared to the Inceptionv3 which obtained highest accuracy of 95% among all pre trained networks.

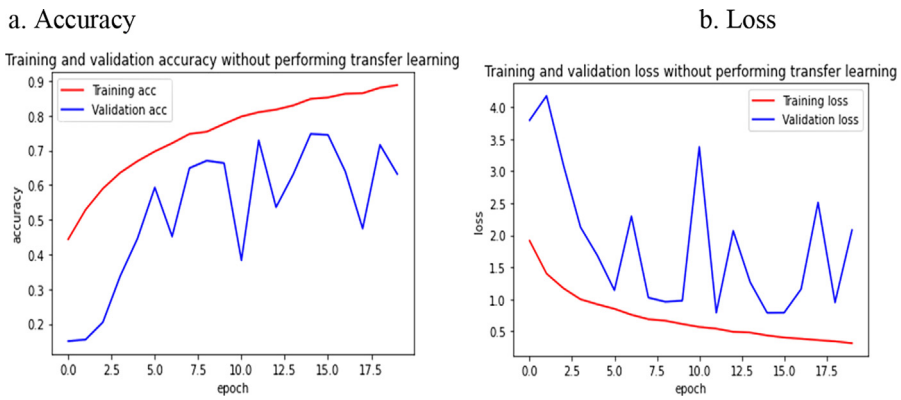

Fig 7. (a) Represents accuracy and (b) represents loss.

5. Conclusion

Using deep learning techniques the animal species are classified and labelled accurately. The accuracy improved by performing transfer learning and training bottleneck features. The four architectures are trained with the dataset and Out of all architectures, InceptionV3 gave high accuracy of 95% but learning curves didn't went well. Then comes mobile net with 93%, then VGG16 with 86% with good learning curves and then resnet50 with least accuracy of 53%. From comparative analysis the accuracy obtained from the pre trained network is more than the model trained without pre trained network.

CRediT authorship contribution statement

Likhith Prudhivi: Conceptualization, Methodology, Software. **M. Narayana:** Visualization, Writing - original draft. **Ch Subrahmanyam:** Data curation, Supervision, Validation. **M. Gopi Krishna:** Writing - review & editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] R. kays, S. Tilak, B. Kranstauber, P. A. Jansen, C. Carbone and M. J. Rowcliffe, 2010, Monitoring wildlife communities with arrays of motion sensitive camera traps, Ithaca.
[2] r. Thangarasu, v. kalippan, r. surendran, s. kandasamy and P. Jayasheelan, 2019, Recognition of animal species on camera trap images using machine learning and deep learning models, International Journal of Scientific & Technology Research, p. 0.
[3] Kavish Sanghvi, Adwait Aralkar, Saurabh Sanghvi and Ishani Saha, 2020, Fauna Image Classification using Convolutional Neural Network, International Journal of Future Generation Communication and Networking, vol. 13, no. Vol. 13 No. 1s (2020): Vol. 13 No.1s (2020) Special Issue, p. 16.
[4] M. Favorskaya and A. Pakhirka, 2019, Animal species recognition in the wildlife based on muzzle and shape features using joint CNN., in Procedia Computer Science, Krasnoyarsk.
[5] Min Lin, Qiang Chen and Shuicheng Yan, 2104, Network In Network, Singapore.
[6] Guobin Chen, Tony X. Han, Zhihai He, Roland Kays and Tavis Forrester, 2014, DEEP CONVOLUTIONAL NEURAL NETWORK BASED SPECIES RECOGNITION FOR WILD ANIMAL MONITORING, in 2014 IEEE International Conference on Image Processing (ICIP), Paris.
[7] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke and Andrew Rabinovich, 2014, Going deeper with convolutions, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA.
[8] K. He, X. Zhang, S. Ren and J. Sun, 2016, Deep Residual Learning for Image Recognition, in 2016 IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA.
[9] H. Gao, Z. Liu, L. v. d. Maaten and K. Q. Weinberger, 2018, Densely Connected Convolutional Networks, arXiv.org.
[10] N. M. Sadegh, A. Nguyen, M. Kosmala, A. Swanson, M. Palmer, C. Packer and J. Clune, 2017, Automatically identifying wild animals in camera trap images with deep learning, in Proceedings of the National Academy of Sciences.