Animal Classification

Problem Definition:

The project aim is to identify the animals mainly in farm and in wildlife monitoring. It is also the problem of animals in human inhabited areas by recognition of animals through AI. Since it is essential to know whether animals are present in human-populated regions for security and traffic safety, this is the main objective. This is significant because there may be a risk to people. It highlights the shortcomings of existing models when utilized in real-world contexts and has consequences for transfer learning, a frequently used technique in machine learning and deep learning. The performance and prediction have been improved by the authors' usage of several models.

Project Objectives:

- The primary goal of this project is to identify animals in the wild forest, which will be highly valuable in identifying new kinds of animals.
- Ensure that the model recognizes all the species of the animals with high resolution and
- Aim for high accuracy when classifying animal photographs. This entails training and fine-tuning the model in order to improve its ability to differentiate between different animal classes and reduce misclassifications.
- The model should generalize well to unseen animal samples and be able to reliably classify animals from a variety of sources, habitats, or viewpoints. Generalization ensures that the model can be used in contexts other than the training data.
- The goals of animal classification can vary based on the application, such as wildlife monitoring, illness detection, behavior analysis, or habitat assessment. These objectives are matched to the relevant application domain's requirements and challenges.

Analysis:

The assumption here is to classify animals from the images and predict which category does it belong to. I have analyzed the model. I have used different model architecture and changed to different number of layers. I have first used normal convolution layers with 32,64,128,512 filters and kernel size as (3,3) and padding as same but didn’t achieve the accuracy I was expecting. So had to try other architectures so tried vgg 16 and loaded the weights of the data, now I have changed to different number of channels, filters, kernel sizes and also added regularization so that it would give better accuracy. I have used two convolution layers and input image size as (224,224). Similarly, I had to try different optimizers such as Adam, SGD and keep changing the learning rate of it to check the accuracy of the model. In the end I found a model fitting better with good accuracy of 85% which uses 512,1024 channels and learning rate as 0.001. I will try to improve my model more by adding or reducing the number of layers. I have used Regularization methods (such dropout and weight decay) and the availability of a suitably diverse and balanced training dataset must all be taken into account in order to prevent overfitting. Accurate categorization depends on the generalization of the model's learned features to unknown animal images.

Model Architecture:

Input shape: Color images of the shape (224, 224, 3) are expected as input to the model.

The variable newmodel indicates a vgg model that uses the input as its input tensor. This model's architecture and parameters are image shape, weights from ‘image net’.

Flatten layer: The Flatten layer flattens the output of the pre-trained model. The 3D feature maps are converted into a 1D feature vector.

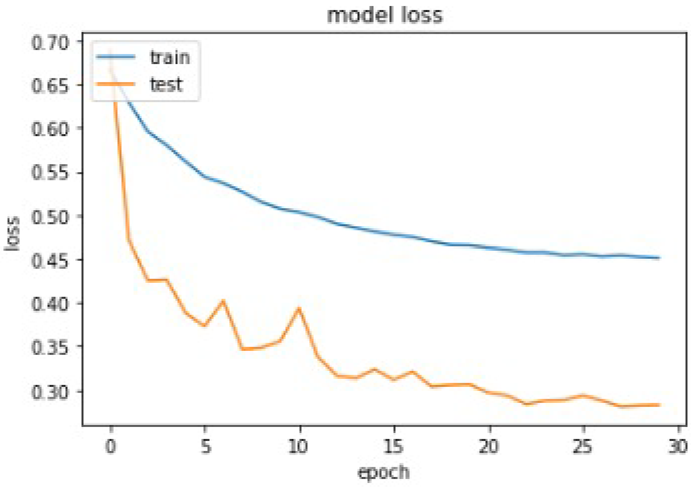
Dense layers: The flattened feature vector is then passed through two dense layers, which are fully connected. The first Dense layer has 1024 units and introduces non-linearity by using the ReLU activation function. The second Dense layer contains 512 units and employs the ReLU activation function.

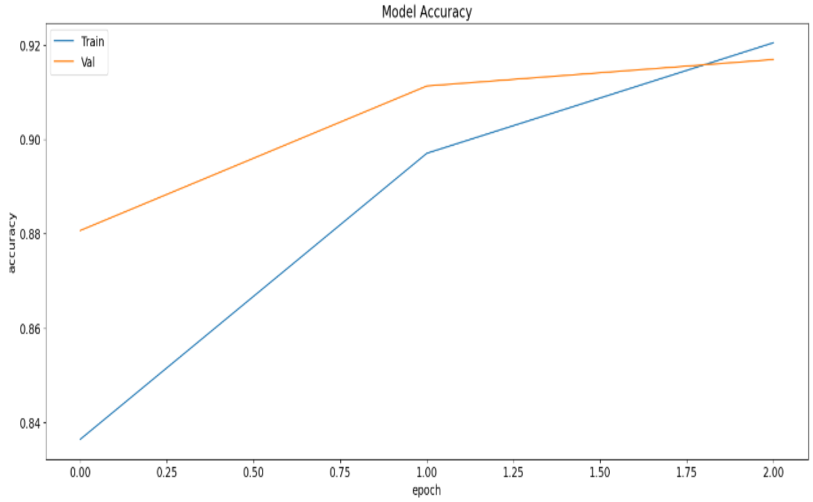
Finally, as the output layer, a Dense layer with 10 units and softmax activation is utilized. The softmax activation function generates a probability distribution expressing the likelihood of each of the ten potential classes.

Model definition: The tf.keras.Model class is used to specify the input and output tensors of the overall model.

Results:

The training approach with the fit function yields the following results: After 0 epochs, the training accuracy is around 82%, with a loss of 0.22. Gradually, the validation loss lowers. Accuracy gradually increases from 30% in the first epoch to 82% in the last epoch.





Discussions:

Here the topic can be the correct selection of model architectures. It was quite long process to work on all the model architectures and find the best one out of it. I have used convolution layers first but didn't achieve a better accuracy which is why I have choose vgg16 which gave me best accuracy of about 85%. As this is one factor, in a similar way there are factors like data collection, to classify animals, one can use pre-trained models, such as those developed using massive image datasets like ImageNet. Even with little training data, transfer learning enables the transfer of knowledge from previously trained models to the task of classifying animals.

In order to handle class imbalance and make sure that the model does not favor the majority classes, strategies like oversampling, under sampling, or class weighting might be used. In order to evaluate the effectiveness of the animal categorization model, it is crucial to select acceptable assessment measures. Commonly used measures include accuracy, precision, recall, F1-score, and top-k accuracy Animal categorization can be complex by an unbalanced class distribution, in which certain animal species have much fewer samples than others. Also, there are factors to consider are the pretrained model selection and the fine-tuning approach.

Evaluation and Reflection

Data Preparation: training split to 80% and validation split as 20%..

Model Training: Trained your animal classification model on the training data using appropriate techniques like convolutional neural networks (CNNs) and Up sampling layers.

Prediction: Utilize the vgg 16 model with trained weights to predict the animal classes for the images in the testing set.

After executing the epochs about 30 of them. I have observed increase in the rate of accuracy while decrease in the loss. Validation accuracy stays same.

There is accuracy metrics with which we can say model is performing better it is not the only factor that decides the performance. There are others such as precision, recall, f1 score.

For this vgg 16 model I have observed an accuracy of 84% and precision as 68% and recall as 0.65, f1 score as 72%.

I have also observed the model's predicted classes are tabulated against the actual classes in a confusion matrix. It makes it possible to analyze the true positives, true negatives, false positives, and false negatives in great detail, giving information on particular classification mistakes.