MULTICLASS WILDLIFE IMAGE CLASSIFICATION USING DEEP LEARNING TECHNIQUES

Data Mining and Machine Learning II

Swashchith Sagar Prashanth
MSc Data Analytics
(School of Computing)
(X18141412)
National College of Ireland
Dublin, Ireland

Srijan Kumar Pandey
MSc Data Analytics
(School of Computing)
(X18127312)
National College of Ireland
Dublin, Ireland

Bhagyashree Tatiya
MSc Data Analytics
(School of Computing)
(X18188788)
National College of Ireland
Dublin, Ireland

Sumit Bhatia
MSc Data Analytics
(School of Computing)
(X18191860)
National College of Ireland
Dublin, Ireland

***Abstract*— Image classification is a process in which we try to classify an image using the visual content and pattern of the images. In this paper we discuss multiclass image classification on a wild life dataset. The grading of the image is accomplished by distinguishing the image into the specified category based on content of the image. The images of various wildlife are used to train the model, so that it is used for classifying the images into different wild creatures. The deep learning technique like Convolution Neural Network is used to train the model and the model is used classify the input images. This research also used transfer learning technique where the data was trained with a pre-trained model called DenseNet. Library like Keras is used to apply deep learning techniques and the layers such as dense is used which connects input nodes to the output node and dropout layer in which the activation is set to zero. Finally, the model is evaluated based on accuracy and loss function. Applying optimizer tuning an accuracy of 92.40% was obtained on CNN trained model and 95.03% was obtained on pre-trained DenseNet model.**

Keywords— Image Classification, Transfer Learning, DenseNet, Convolution Neural Network, Deep learning techniques, dense, dropout.

I. Introduction

Image Classification is a hierarchical arrangement based on its features, in groups and categories. Classification of images came about to reduce the difference between machine vision and human vision by training the machine with the data. The grading of the image is accomplished by distinguishing the image into the specified category based on content of the image. This technique would be very useful in other department like medical science to identify any specific disease based on the characteristics of the image, microbiology, research and educational purpose.

In this paper, inspired by “Deep Learning Toolbox Model for AlexNet Network”, [21] we discuss the study of classification of images using deep learning. The traditional methods used for the classification of images are part in the artificial intelligence (AI) area which generally called as machine learning.

[22] Deep learning (DL) is a sub-field of machine learning which can learn from its own computational system. A profound model of learning is implemented to consistently break down knowledge into a homogeneous structure in a way how a human being take decisions. In order to achieve this , deep learning takes advantage of a layers of multiple parameters which is represented as an artificial neural network.

A. Wildlife Image Data

[24] The processing of animal imagery data with movement responsive cameras is a actually one of the most invasive approach to collecting numerical abundances and predicting population of wild lives patterns over time. It helps analysts to test their studies virtually by recording animals out of the images collected.

The images of wild lives are obtained from Kaggle. The dataset consists twenty different types of wild specious which will be used to train the model in a way that the model can be used to classify the input image.

B. Convoltional Neural Network

The main building blocks employed in convolutional neural networks are the standard layers. [23] A convolution is nothing but applying of a function to the input which results as an activation. Frequently applying the same function to the same input will generate activation map called a feature map, which represents the position and intensity of a recognized feature in an input like an image

The breakthrough of convolutional neural networks is the capacity to learn efficiently, with the limitations of a particular problem of predictive model like as image classification, a huge number of layers in parallel similar to a training data set. The effect is very particular features that can be found in the image data.

The convolutional neural network or CNN is a modified type of neural network meant to perform really well with two-dimensional image data efficiently even though it can be applied for one-dimensional and three-dimensional data.

C. Data Augumentation Technique

Data augmentation techniques is one of the most used operation in visual recognition, as quick and simplistic image processing makes it easy to produce new images. Though, it is tough to find effective transformation strategies for increasing the text data that also maintain the qualitative and punctuation structure of word documents.

[25] Within this paper we discuss several techniques of image data increase within. We are studying the impact of different augmented datasets on the productivity of several deep learning models for classification of relationships between images.

[26] In several Computer Vision projects, deep convolutional neural networks worked surprisingly well. Nevertheless, these networks depend on big data to prevent overfitting. Overfitting is nothing but the phenomenon where a network is learning a task with quite large variance like those of modelling the training data perfectly. Sadly, most technology areas do not provide connections to broad data like medical image analysis.

D. Transfer Learning

Transfer learning (TL) is one of the topics related to research in machine learning (ML) that centers on storing the acquired information while tackling one issue and applying it to a specific still related topic. For example, while trying to identify trucks, knowledge acquired by learning to identify cars could be applied

E. DenseNet

The Dense Convolutional Network (DenseNet) is a feed-forward network which connects each layer to every other layer in a feed-forward manner Whereas conventional L-layer of convolution networks have L connections, one layer in between each layer and its corresponding layer and our network has L(L+1)/2 direct connections.[26]

II. Research Question

How effectively can pre-trained models perform over Deep learning techniques for multiclass image classification?

III. Related Work

Classification is a process were objects, ideas are been recognized, understood and placed into different categories depending on their characteristics. In the same way image classification also takes place. Classifying images into different categories depending upon their types can be done by using various algorithm. In [1], the author has performed classification of Vehicle images using various Machine Learning Algorithm such as the Random Forest, Multilayer Perceptron, Simple Logistics, Sequential Minimal Optimization (SMO) etc. Among all this SMO achieved highest accuracy of 82.37%. This basically deals with Supervised Learning Algorithm so further various other Unsupervised Learning Algorithm were used. Here [2] Supervised and Unsupervised Algorithm were used for classifying images of animals. Accuracy comparison was done by using Linear Discriminant Analysis (LDA) and Symbolic Classifier as Supervised Learning Algorithm and Principal Component Analysis (PCA) and K-means as Unsupervised Learning Algorithm. The accuracy obtained by using LDA and Symbolic Classifier is more than that of PCA and k-Means.

Various others Machine Learning algorithm can be used for image processing as proposed in [6]. Here [5] the author has used Support Vector Machine and Convolutional Neural Network (CNN) for Classification of Food images. The results obtained from the CNN are way better than that of SVM as the images passes through various layers of the model. The model proposed in [6] can be used to extract the text content from the images using the python packages. This is done by using Long Short-Term Memory (LSTM). In [11] various machine learning algorithm were used in CNN in order to image classification. The comparison result of all these algorithms is given where SVM has the highest accuracy among all.

The images captured by the sensors are stacked into the server for classification were different classification algorithm can be used. Here [3] the use CNN of is done where a single filter is used for small IoT devices wherein the images captured by these devices would pass through the network which would eventually classify them. The size of the dataset matters a lot when it comes to gaining more accuracy from the model [4]. The use of 32 layers Residual Network (ResNet) is done where the layers of CNN are stacked. In wider sense, deeper the network better is the accuracy. Also, the accuracy of the network depends upon the number of filters being added in it. The author proposed a Bilateral Convolutional Neural Network which would be feasible with mobile devices [7]. The network is divided into 2 halves which has the specified set of responsibility. The accuracy obtained from this is 85.6%. Along with this the use of ResNet is done which has given relatively low results. Deep Convolutional Neural Network (DCNN) was proposed for recognizing the species of animal and classifying them into different categories. Images of animals of different species were used along with the bag visual words model as the basis of image recognition [8].

A classification method of criminal investigation using a combination of Spatial Convolutional Neural Network (SCNN) and Extreme Learning Machine (ELM) is used. This proposed method has given more accuracy than CNN [9]. Another implementation of CNN using Keras and Tensorflow that would run on Raspberry Pi was done [10]. This was executed for the classification of snow leopards with the accuracy ranging from 74% to 97% respectively. Combination of two Neural Network Architecture is done for the classification of images namely the CNN and Shallow CNN [12]. The fusion of these methods has relatively improved the performance of the classifier by 5% to 11%.

Classification of fruit image using the MobileNetV2 with Transfer Learning is proposed [13]. Compared with Inception V3 and DenseNet, MobileNetV2 is suitable for mobile phones. This pre-trained network was used to extract the features and Softmax classifier was used to classify the features by performing at the accuracy of 85.12%. MobileNet architecture is used for solving the overfitting problem. Apart from this Data Augmentation technique was used so that a new training image would be generated which will relate to the same image on which this technique is applied [14]. After applying this technique, the data is then passed through the MobileNet which performs well compared to other techniques.

Transfer Learning based Deep Learning approach is implemented for classifying the objects [15]. Various models along with the CNN are used where the performance of CNN is much higher as compared to other models. Along with this Vehicle classification is also done by using the videos of the roadway by Haar-like features. Use of pre-trained CNN along with Transfer Learning is done [16]. The proposed model gives the performance accuracy of 98% which is way higher than that of a model without Transfer Learning. Classification of Sports videos is important for broadcasting companies. Here [17] a model is proposed using CNN, RNN and Transfer Learning. The features extracted from CNN are combined with the sequential information from RNN where further application of Transfer Learning takes place. The accuracy obtained by the model is 94%. The use of Deep Learning is also done in Medical field for various purposes. Here the use of DCNN is done with Transfer Learning [18]. Different validation technique was used such as Sensitivity, Specificity, Accuracy which showed that the performance of the model in contrast with other models.

Image captured from the cameras can be used for analyzing and obtaining a specified result from it. In the same way images captured from the CCTV can be used to analyze the disaster-prone areas in the country. The existing CNN achieves good accuracy, but it is unable to mark the affected areas [19]. So, the use of Transfer Learning is done to reduce the training time and computation resources required during execution. The proposed method achieved a specificity of 98.5% in real-world images. There are numerous species of dog which needs to be classified into different categories in order to provide appropriate training. The author presents two approaches in [20] where one approach is by using the CNN with Transfer Learning. This approach outperforms the other approach with the accuracy level of 96.75%. A dual channel CNN is used for image classification of Polarimetric SAR images in order to collect various spatial features.[27] The collection of features is done by two CNN structures are placed parallelly. The performance of this model is 91.01% compared to the other methods. Author has proposed a CNN model of classification and detection of plant leaf disease [28]. For extracting the image from every leaf use of RGB images are converted into white and then converted into grey image. Then application of morphological function on the image took place. The performance of the proposed CNN model is 99.53% which shows that the plant diseases are identified correctly.

Dual Shot Face Detector using DenseNet as the backbone is proposed in [29]. Apart from this various other function for classification and regression loss function is done. However, the accuracy obtained by using ResNet is low as compared to that of DenseNet which is 85.22%.

IV. Methodlogy

The methodology used here to describe the framework is Knowledge Discovery in Database (KDD). This methodology has 5 phases. This framework helps in solving the problem systematically by giving the guidelines for the steps that needs to be followed. Here Convolutional Neural Network is used to solve the classification problem. The phases of KDD are modified according to the need of the problem. The below shown figure represent various phases of KDD.

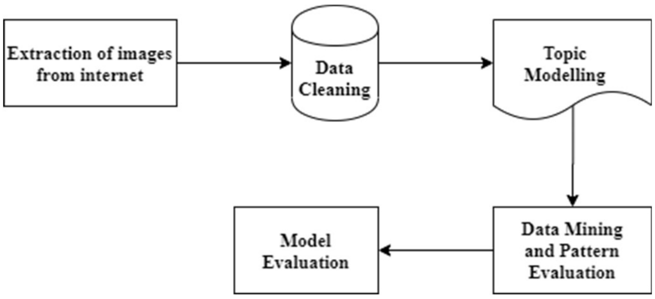

Extraction of images from internet
Data Cleaning
Topic Modelling
Data Mining and Pattern Evaluation
Model Evaluation

Fig 1: KDD

A. Extraction of Images from Internet

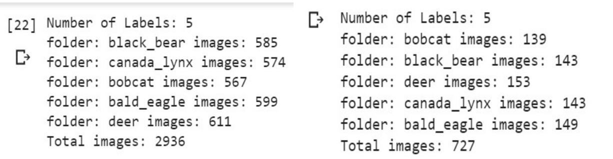
In this phase all the data required for model construction and evaluation is collected and stored in database for further processing.

B. Data Cleaning

In this phase all the data that is collected from the previous phase is cleaned and transformed so that it can easily pass through the model. Here the images are separated into different folders so that high accuracy could be obtained from the model. The dataset initially contained 14013 images which was distributed in 20 class of wild animals. For the purpose of this project 5 different classes of image has been taken considering the hardware capabilities because the size of the entire folder was over 5 GB. That's the reason why we have used a subset of this large dataset reducing the number of classes to 5.

C. Data Preprocessing

In this project in order to get neural networks function with a high accuracy the data needs to be processed. The image data has a structure where independent variables are the pixels distributed in 3D arrays. For the preprocessing of the data first it was divided in training sets and Test set. The division was made in ratio of 80:20 where 80% of the data went to our training set and then rest 20% to our test set. This was followed by doing some exploratory Data Analysis (EDA). The colored images are converted into 3D arrays and black and white images are converted into 2D arrays. Since all the images we have are colored so this project would use the images which are in 3D arrays. The images in the folder has size of (964, 750) over 3 channels of RGB. Before passing the images in our model it needs to be first converted into a fixed size so that all the images of same size are passed through models. In order to process the image, the size of it has been reduced to (150, 150) for our convolutional neural network model and (224, 224) for transfer learning on pre-trained DenseNet121 model respectively. Few other results of the EDA are shown below:



```
[22] Number of Labels: 5
     folder: black_bear images: 585
     folder: canada_lynx images: 574
     folder: bobcat images: 567
     folder: bald_eagle images: 599
     folder: deer images: 611
     Total images: 2936
```

```
Number of Labels: 5
folder: bobcat images: 139
folder: black_bear images: 143
folder: deer images: 153
folder: canada_lynx images: 143
folder: bald_eagle images: 149
Total images: 727
```

Fig 2: Training and Test data images.

Data Augmentation was also used a pre-processing technique because the number of Images in our training data is very less so multiple training data was generated by rescaling, zooming, flipping the image horizontally so that we have sufficient amount of data and our model can be trained properly.

The images were also shuffled using the library random so that the basic features can be extended for python and the images in the list can be shuffled as well. After performing the EDA and all the pre-processing steps the nest stage is to create our model.

D. Data Modelling

This research has been carried out using two techniques. These are:

- Multiclass classification using Convolutional Neural network
- Multiclass classification using transfer learning on pre-trained DenseNet.

The purpose of using two models were to learn the use of these models in image classification and also to compare the accuracy of our data over both the models. We will compare the result of both the models in the end. The experiments were carried out in Jupyter notebook for CNN and Google Colab for transfer learning on pre-trained DenseNet model. Python was used as programming language.

Classification using CNN

In the first test a Deep Learning model i.e. Convolutional Neural Network (CNN) is applied to the dataset. CNN is an artificial neural network on which we use convolution trick to add some convolutional layers. This is used to preserve spatial structures in images and then use these structures to classify images. CNN are used for image classification and recognition problems such as identification of objects, classification of images etc. It basically takes the input image, assign weights to various features of the image which would differentiate it from the other images and these weights further helps it for classification. Keras and Tensorflow are the libraries which are used for performing Deep Learning algorithm. Along with this Data Augmentation technique is used in which the brightness level of the image, contrast, shearing, flipping etc. are performed. Our goal would be to classify images and to tell of each image the class of the image. This project is using 5 classes of wildlife image that is black bear, Canada lynx, bobcat, bald eagle and deer. This project will use the help of python codes to extract the label name from 5 classes mentioned above from images of these labels to specify to the algorithm whether the image belongs to the same class or not. And somehow, we get our dependent variable vector because we can fill in the dependent variable vector with the label name which we are able to extract from the images of these corresponding labels. The libraries used for this classification is Sequential from keras.models. This would be used to initialize our neural network. There are 2 ways to initialize a neural network. Either through layers or through graphs. Since CNN is a sequence of layers, we will use Sequential to initialize it. Convolution2D, MaxPooling2D, Flatten, Dense, Dropout from keras.layers and optimizers from Keras library. Keras is a deep learning library it contains some tricks and tools to import images in efficient way. In order to import images from Keras we need to prepare a structure of our dataset which we would be discussing in the following steps. The images have already been labeled in our pre-processing step. Keras is quick and very easy to use. It is very user friendly and can be used on both CPU and GPU. Another advantage of Keras is that it can be used on both convolutional and recurrent neural networks. The image in CNN passes through various layers which are discussed below. These layers are:

1. Convolutional Layer

In this layer filters are applied to each part of the image. These small parts of image are collectively called as weights which are mostly of size 3X3 pixels. As the image is moved there is a check for pattern in that section of image. When the image is trained the weights get updated, so by the time of evaluation these weights return high values if it encounters a pattern that it has already seen before. The combination of these weights let the network predict the content of the image. Rectified Linear Unit (ReLU) is a Non-Linear activation function which is used in this layer. It identifies the positive values and zero for negative values.

2. Pooling Layer

In this layer partitioning of the image into set of rectangles such that each rectangle outputs a value. Basically, it is function that reduces the spatial size of the representation to reduce the number of parameters in the network. Also, further computation on the network is reduced. Pooling layer operated independently on each weight. As the number of parameters are reduced the chances of overfitting of data get reduced. There are other more two type of pooling i.e. the Max pooling which outputs the maximum value of the sub-region and Average Pooling which outputs the average value of the sub-region.

3. Flattening

In this layer all the output obtained from the Pooling Layer is in matrix format which is flattened here. This flattened output is in 1-dimenesional array format. Further the data needs to pass through the Fully connected layer where this array like structure is generated is connected to every block of last layer.

4. Fully Connected Layer

This is the last layer of the model where every block is interconnected. In this layer final classification of the image takes place. The accuracy obtained from this model says whether the data is correctly classified or not.

Steps of creating the Convolutional Neural Network

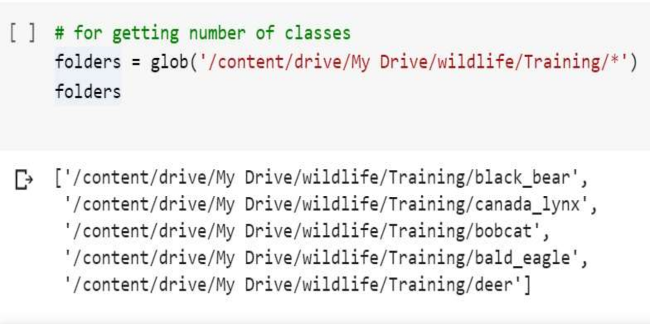
- Importing Keras libraries and packages.
- Initialising the CNN using Sequential class to create an object of this class and it was named as classifier.
- Creating our first convolutional layer. In this step it applies several feature detectors on our input image. Each feature detector slides all over the input image. For each feature detector we apply on the input image we get a feature map. Feature map contains some numbers and the highest numbers of the feature map is where feature detector detected specific feature in the input image. This is called convolution operation. So, this project was performed using 32 feature detectors of 3 rows and 3 columns which will go through the image to create 32 feature maps. The reason of choosing 32 filters was only because this project was performed on CPU so considering the hardware capacity the research started with 32 first.
- The next step is passing the input shape of our images which was (150, 150, 3) since this project is using Tensorflow backend.
- The activation function used is ReLU in order to avoid any negative pixel values and avoid any linearity.
- The next step involved using of max pooling to get our pooling layer. This is used to reduce the number of nodes that we get for next step. Pooling layer has reduced feature maps from convolution layer. Pool size was taken (2,2).
- All the pooled feature map was converted into a single vector and the technique used was flattening. We get a huge single vector of different cells from pooled layers and this single vector would be the input layer of the classic artificial neural net of fully connected layers which will be created after that.
- The final step of CNN architecture is to create a fully connected layer which can also be called as hidden layer which will be followed by creating an output layer. The output dimension for hidden layer was kept as 128 to make sure its not to low so the model couldn't get trained properly or too high to increase the computational complexity. The activation function used was ReLU.
- The final output layer was created with dimension of 5 because we have 5 classes in our data with activation function as softmax because we have an outcome which is of multiple category.
- All the layers were compiled together in the next step with Stochastic Gradient Descent (SGD) as optimizer, categorical cross entropy as loss function and accuracy as the metrics.
- Data augmentation was carried out to generate multiple images for effective training purposes. Batch size was used as 64 to decide how many images should be sent together and class mode was selected as categorical since the project is working on a categorical output of data.
- The final step was fitting the model to our training data and then calculating the accuracy on our test result. As this code was executed on CPU so this research went with only 10 epochs over the data and the results were evaluated based on 10 epochs. The results of which would be discussed in evaluation section.

Optimizer: The optimizer used in this experiment was SGD which is called as Stochastic Gradient Descent. The function of an optimizer is to reduce the cost function and help things speed up and get to minimum faster. This requires cost function to be convex. This optimizer basically functions where the rows or the data is passed through our neural network one by one and then it adjusts the weight after each iteration. This helps us to avoid the situation to find local minimum rather than going for the overall global minimum and it is able to do that because it has much higher fluctuations. It is faster as well because it doesn't have to load all the data into memory and can run it one by one.

Classification using Transfer Learning on pre-trained DenseNet121

Another approach used in order to classify multiclass images were to use transfer learning on a pre-trained DenseNet121 model. Before this project explains the methodology of this technique it would be important to learn about transfer learning. Transfer learning is a technique in which we can use the knowledge which we have gained from computing one problem to solve other problems which are similar in nature. Transfer learning here is some of the convolutional neural network designed which gives us a state of art algorithm to identify different types of images. This project would use DenseNet which was derived from ImageNet database which is a platform which produces a best state of art algorithms based on convolutional neural network where the researchers try to improve the model which can work better on similar kind of problems. Every year a new algorithm is introduced by ImageNet. The DenseNet model which I am using is very well compatible with Tensorflow. These models are trained on weights and this research will use it to train our training dataset and then use it to test our testing data. This will involve changing the output layer of DenseNet to solve this project problem statement. Image net is basically created to categorize on 1000 different images. DenseNet121 is also a model based on sequential layer. So, in order to use it this project will remove the last layer of DenseNet and then replace it with 5 outputs because our dataset has 5 outputs and then we will append this to the last layer of DenseNet. This is the entire concept of transfer learning. Some of the packages which were imported was DenseNet121, preprocess input and image from Keras. This test was performed on google colab which is a cloud service that supports GPU as well and it can be used to program deep learning algorithms without using hardware space. The steps performed while following this technique is:

- Uploading the dataset in google drive.
- Connecting google drive to google colab by downloading, importing and installing some drivers and codes.
- Mounting the dataset with google colab.
- Importing all the necessary libraries.
- Resizing all the images of size (224, 224) because DenseNet was created with same image size.
- Assigning training path and validation path of the dataset to a variable called train_path and valid_path.
- The next step involved removing the last layer of DenseNet algorithm because we have only 5 classes and it is trained on 1000 classes.
- As input shape the size of the image was passed along with 3 channels as the images, we have on the dataset is coloured image.
- Weight would be provided by ImageNet weights which is already present in Keras.
- The next step involves avoiding the training of DenseNet layers because it is already trained. If we don't do this the whole model will start training again and again.
- The next operation used was to check the number of categories we have in our dataset for which glob function was used.



```
[ ] # for getting number of classes
    folders = glob('/content/drive/My Drive/wildlife/Training/*')
    folders
```

```
['/content/drive/My Drive/wildlife/Training/black_bear',
 '/content/drive/My Drive/wildlife/Training/canada_lynx',
 '/content/drive/My Drive/wildlife/Training/bobcat',
 '/content/drive/My Drive/wildlife/Training/bald_eagle',
 '/content/drive/My Drive/wildlife/Training/deer']
```

Fig 3: number of classes in folder

- The next thing which was done was flattening the last layer of DenseNet121 which would be done using flatten function.
- A variable prediction was created using the classification folders as our dense layer and softmax as activation function because we have multiclass outputs.
- Model was compiled with Categorical cross entropy as loss function and optimizer as Adam with metrics as Accuracy.
- The next step involves data augmentation like we did for the last test in order to generate more training data.
- After performing data augmentation step the model was fitted with the training data with 10 epochs to calculate accuracy and loss.

Adam Optimizer: Adam also known as Adaptive Moment Estimation is an optimization algorithm which is made by combination of AdaGrad and RMSProp. This algorithm is very useful when we have a noisy situation and we have to handle sparse gradients. The advantage of this optimizer is that it doesn't require any parameter like learning rate or decay to assign to it because it updates itself with these parameters. It is very fast and helps in reducing cost function while training of a neural network. The learning rate of Adam optimizer is called lambda which is always used as an upper limit.

V. Evaluation

Evaluation is very important to check how well a machine learning model can be implemented in a real time problem. The evaluation can be considered using multiple metrics however this project is considering accuracy and loss function. Accuracy can be defined using confusion metrics mentioned as:

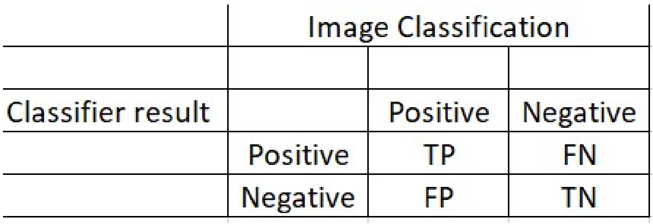


	Image Classification		
Classifier result		Positive	Negative
	Positive	TP	FN
	Negative	FP	TN

Accuracy = (TP + TN)/ (TP + TN + FP + FN)

In this section we will discuss the evaluation on both the tests one by one.

Evaluation of model trained on CNN

In this section we will evaluate the output produced by our CNN model. The CNN model used in this model was used with a single convolution layer and single dense layer however the initial results showed overfitting of the data. The model was later tuned in order to get a good accuracy.

Use of Dropout to avoid overfitting

In order to avoid overfitting of our training data this project used dropout technique. This technique basically drops out some input at the update of each training set in order to reduce the over dependency of layers on few of its input.[30] Dropout was kept as 0.2 and was added after flattening the parameters.

Tuning of SGD

SGD was tuned by adding learning rate, decay, momentum and nestrov. Learning rate is a hyper parameter which is used to adjust the weight of the network with loss gradient. The value of learning rate was chosen to be 0.01 as it can't be too low because that will take time to train model and it can't be too high as well because then it will cause irregular behavior in our loss function. the value of decay was chosen to be 1e-6 (10 to the power minus 6) which means how much each update will affect the current weight and momentum as 0.9.

The graph obtained for Loss of model and accuracy after completing 10 epochs are given below.

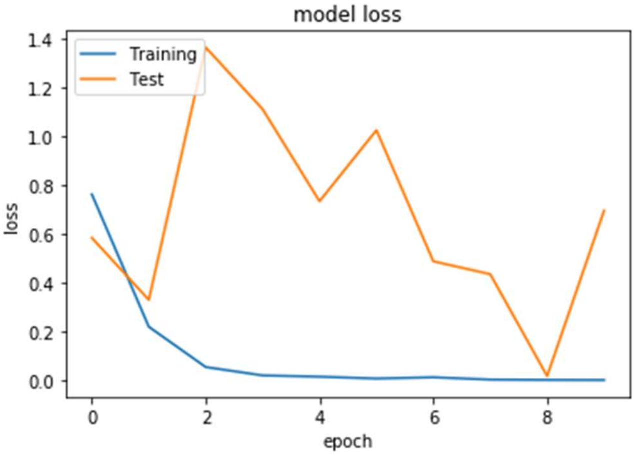


Fig 4: Model loss for training and test set over epochs

As we can see from the graph the learning rate is showing a zig zag movement but it is coming down with increase in epoch except for the last epoch where it increased from previous one. We can say that there is still some overfitting because training loss is less than that of validation loss. Our desired result would have been if both of them were almost equal. The next we will discuss the result with the help of graph of accuracy over epochs for training and test set.

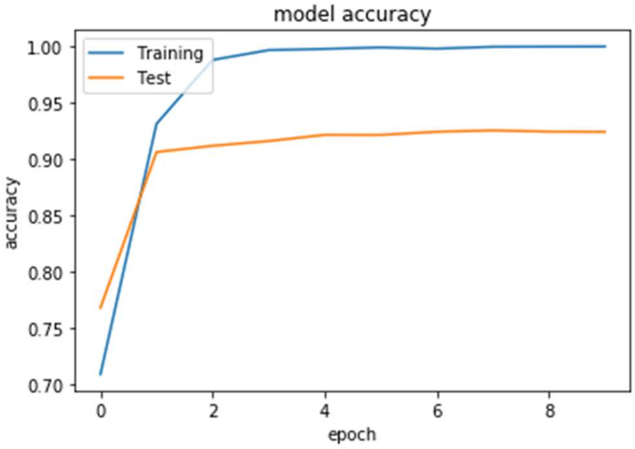


Fig 5: Model accuracy of training and test over epochs

The graph between training accuracy and test accuracy can be seen above. This graph can be seen where the training and test accuracy increased for an interval and then remained constant. This is a perfect example when our training data is very less we get such kind of graphs.

The model gave an overall accuracy of over 99% on the training set and validation accuracy was 92.4% on our test set. In the next section we will discuss the performance of our test using transfer learning on DenseNet.

Evaluation of Transfer Learning on pre-trained DenseNet

In this section we will evaluate the output produced by our model which was created by using transfer learning on pre-trained DenseNet model. The model used a single dense layer and initial results showed overfitting of the data like the previous test. The model was later tuned in order to get a good accuracy

Use of L2 Regularization.

Since the model showed overfitting so a L2 kernel regularizer was used in order to avoid overfitting.[31] L2 regularizer is basically a technique which helps in avoiding the overfitting by adding penalty which is equal to sum of squares of coefficients. The loss of L2 regularization can be explained

$$Loss = Error(y, \hat{y}) + \lambda \sum_{i=1}^{N} w_i^2$$

Use of Early stopping

Early stopping is done in Keras with the use of callback parameter. This is also an effective parameter to avoid overfitting of data by stopping the training of the dataset when validation loss reaches to minimum.[32] The graphs obtained after implementing these techniques are as follows.

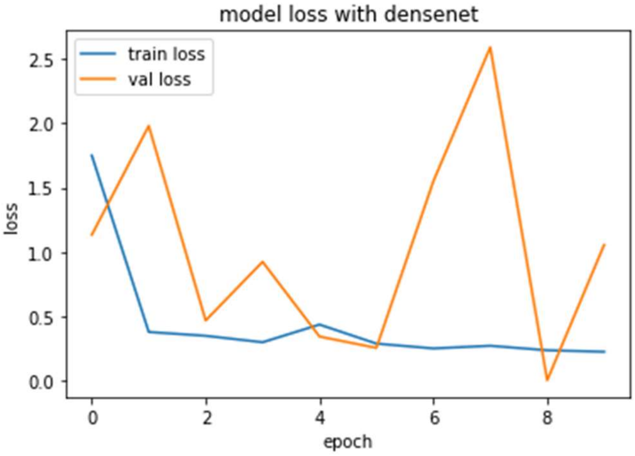


Fig 6: Model loss on train and test set over epochs.

We can observe in this image that model loss for training set came down to some extent in the beginning and then it kind of became stationery. Whereas for the test set it reached to a maximum point and then got executed when it came to lowest point. When tried to remove the call back as it was getting executed after 4 epochs, we get the above graph.

Next, we will see the graph for the accuracy of our model trained by using transfer learning on DenseNet.

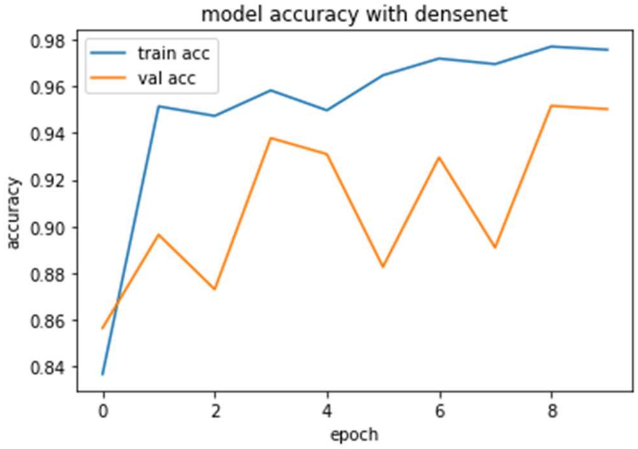


Fig 7: Model accuracy on train and test set over epochs

The graph between training and test shows an increase in accuracy for both the training and test set at the execution of the model. The model gave an accuracy of 95.03% after execution.

The model was later tested with an image of a deer and it was able to classify the deer from the image. The codes for that are present in the codes uploaded.

VI. Conclusion and Future Work

The main objective of this project was to perform multiclass image classification over an image dataset which had 5 classes of wildlife animal. The aim was to obtain high accuracy in classifying these animals using deep learning technique. This was also an attempt to learn the concepts of deep learning techniques. This project was designed based on 2 experiments. In one experiment a simple convolutional neural network was used to train our training dataset and then accuracy was calculated over test set. On the other experiment a transfer learning approach was used to train the image data using a pre-trained DenseNet model and to compare the accuracies of both the model. The model created using the pre trained DenseNet model gave better accuracy than the conventional Convolutional Neural network. This project also aimed towards visualising the performance using metrics like accuracy and loss. After performing the tests we came to a conclusion that both CNN and using a pre-trained DenseNet can be used for image classification. The CNN model gave an test accuracy of 92.40% whereas pre-trained DenseNet gave an accuracy of 95.03%. This shows that a pre-trained DenseNet gives a better accuracy on the same dataset in 10 epochs each.

Even though we have received some optimitic result from this research there is a lot which can be done to increase the performance of the model. This research paper used a single convolution layer and dense layer while training the model with CNN. Adding more convolutional layer and dense layer will definitely increase the performance. Similarly because of hardware and time constraints this research couldn’t use a very large dataset. The dataset were comparatively small and the results shows the consequence of using a very small dataset. More number of training data would help in improving the performance of the result. The time which the model training takes were very high which reduced the possibility of applying different optimisation tuning during the execution. Tuning of few parameters would help in increasing the performance of the model.Converting the images into grey scale and then trying to find out the important characteristics might also work however this technique was not used in first place because there were few animal which are differentiated from other only on the basis of their color. This project also left a lot of questions which needed to be answered but where actually not clear due to lack of resources. If given more time the research will look around to explore other techniques to improve the performance and also how to remove over and underfitting.

VII. References

[1] Suguitan, Agnes & Dacaymat, Lucille. (2019). Vehicle Image Classification Using Data Mining Techniques. 13-17. 10.1145/3339363.3339366.

[2] N. Manohar, Y. H. S. Kumar and G. H. Kumar, "Supervised and unsupervised learning in animal classification," 2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Jaipur, 2016, pp. 156-161.

[3] Murata, Kenya & Mito, Masataka & Eguchi, Daisuke & Mori, Yuichiro & Toyonaga, Masahiko. (2018). A Single Filter CNN Performance for Basic Shape Classification. 139-143. 10.1109/ICAwST.2018.8517219.

[4] C. Luo, X. Li, L. Wang, J. He, D. Li and J. Zhou, "How Does the Data set Affect CNN-based Image Classification Performance?," 2018 5th International Conference on Systems and Informatics (ICSAI), Nanjing, 2018, pp. 361-366.

[5] A. Şengür, Y. Akbulut and Ü. Budak, "Food Image Classification with Deep Features," 2019 International Artificial Intelligence and Data Processing Symposium (IDAP), Malatya, Turkey, 2019, pp. 1-6.

[6] Deepa, R. & Lalwani, Kiran. (2019). Image Classification and Text Extraction using Machine Learning. 680-684. 10.1109/ICECA.2019.8821936.

[7] B. Jiang, W. Huang, W. Tu and C. Yang, "An Animal Classification based on Light Convolutional Network Neural Network," 2019 International Conference on Intelligent Computing and its Emerging Applications (ICEA), Tainan, Taiwan, 2019, pp. 45-50.

[8] G. Chen, T. X. Han, Z. He, R. Kays and T. Forrester, "Deep convolutional neural network based species recognition for wild animal monitoring," 2014 IEEE International Conference on Image Processing (ICIP), Paris, 2014, pp. 858-862.

[9] D. Li, X. Qiu, Z. Zhu and Y. Liu, "Criminal Investigation Image Classification Based on Spatial CNN Features and ELM," 2018 10th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC), Hangzhou, 2018, pp. 294-298.

[10] B. H. Curtin and S. J. Matthews, "Deep Learning for Inexpensive Image Classification of Wildlife on the Raspberry Pi," 2019 IEEE 10th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON), New York City, NY, USA, 2019, pp. 0082-0087.

[11] Jaswal, Deepika & Vishvanathan, Sowmya & Kp, Soman. (2014). Image Classification Using Convolutional Neural Networks. International Journal of Scientific and Engineering Research. 5. 1661-1668. 10.14299/ijser.2014.06.002.

[12] Jaworska, Tatiana. (2019). The Fusion of Two NN Architectures for the Improvement of Image Classification. AIAM 2019: Proceedings of the 2019 International Conference on Artificial Intelligence and Advanced Manufacturing. 1-8. 10.1145/3358331.3358352.

[13] Xiang, Qian & Wang, Xiaodan & Li, Rui & Zhang, Guoling & Lai, Jie & Hu, Qingshuang. (2019). Fruit Image Classification Based on MobileNetV2 with Transfer Learning Technique. CSAE 2019: Proceedings of the 3rd International Conference on Computer Science and Application Engineering. 1-7. 10.1145/3331453.3361658.

[14] Phiphitphatphaisit, Sirawan & Surinta, Olarik. (2020). Food Image Classification with Improved MobileNet Architecture and Data Augmentation.

[15] Hafemann, Luiz Gustavo & Soares de Oliveira, Luiz & Cavalin, Paulo & Sabourin, Robert. (2015). Transfer Learning between Texture Classification Tasks using Convolutional Neural Networks. 10.1109/IJCNN.2015.7280558.

[16] S. Y. Jo, N. Ahn, Y. Lee and S. Kang, "Transfer Learning-based Vehicle Classification," 2018 International SoC Design Conference (ISOCC), Daegu, Korea (South), 2018, pp. 127-128.

[17] M. A. Russo, L. Kurnianggoro and K. Jo, "Classification of sports videos with combination of deep learning models and transfer learning," 2019 International Conference on Electrical, Computer and Communication Engineering (ECCE), Cox'sBazar, Bangladesh, 2019, pp. 1-5.

[18] T. Kaur and T. K. Gandhi, "Automated Brain Image Classification Based on VGG-16 and Transfer Learning," 2019 International Conference on Information Technology (ICIT), Bhubaneswar, India, 2019, pp. 94-98.

[19] P. Pooyoi, P. Borwarnginn, J. H. Haga and W. Kusakunniran, "Snow Scene Segmentation Using CNN-Based Approach With Transfer Learning," 2019 16th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), Pattaya, Chonburi, Thailand, 2019, pp. 97-100.

[20] P. Borwarnginn, K. Thongkanchorn, S. Kanchanapreechakorn and W. Kusakunniran, "Breakthrough Conventional Based Approach for Dog Breed Classification Using CNN with Transfer Learning," 2019 11th International Conference on Information Technology and Electrical Engineering (ICITEE), Pattaya, Thailand, 2019, pp. 1-5.

[21] "Deep Learning Toolbox Model for AlexNet Network" https://in.mathworks.com/matlabcentral/fileexchange/59133-neural-network-toolbox-tm--model-for-alexnet-network.

[22] "H. Lee, R. Grosse, R. Ranganath, and A.Y. Ng. Convolutional deep belief networks for scalable unsupervised learning of hierar-chical representations. In Proceedings of the 26th Annual Interna-tional Conference on Machine Learning, pages 609–616. ACM, 2009" *(PDF) Image classification using Deep learning*. Available from: https://www.researchgate.net/publication/325116934_Image_classification_using_Deep_learning [accessed Apr 30 2020].

[23] "How Do Convolutional Layers Work in Deep Learning Neural Networks?" by Jason Brownlee.

[24] "Insights and approaches using deep learning to classify wildlife" by Zhongqi Miao, Kaitlyn M. Gaynor, Jiayun Wang, Ziwei Liu, Oliver Muellerklein, Mohammad Sadegh Norouzzadeh, Alex McInturff, Rauri C. K. Bowie, Ran Nathan, Stella X. Yu & Wayne M. Getz.

[25] "A Study of Various Text Augmentation Techniques for Relation Classification in Free Text" by Praveen Badimala, Chinmaya Mishra and Reddy Kumar Modam Venkataramana.

[26] "Densely Connected Convolutional Networks" by Gao Huang of Cornell University, Laurens van der Maaten of Facebook AI Research, 2016

[27] W. Hua, S. Wang, W. Xie, Y. Guo and X. Jin, &quot;Dual-Channel Convolutional Neural Network for Polarimetric SAR Images Classification,&quot; IGARSS 2019 - 2019 IEEE International Geoscience and Remote Sensing Symposium, Yokohama, Japan, 2019, pp. 3201-3204.

[28] Ferentinos, Konstantinos. (2018). Deep learning models for plant disease detection and diagnosis. Computers and Electronics in Agriculture. 145. 311-318. 10.1016/j.compag.

[29] A. Nandy, &quot;A Densenet Based Robust Face Detection Framework,&quot; 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW), Seoul, Korea (South), 2019, pp. 1840-1847.

[30] "Wang, Muhammad, Hong "Alcoholism identification via convolutional neural network based on parametric ReLU, dropout, and batch normalization" 2018 Intelligent Biomedical Data Analysis and Processing.

[31] Cosovic, jankovic, "CNN Classification of the Cultural Heritage Images", 2020, IEEE conference.

[32] Gilberto Luis De Conto Junior, "Diabetic Retinopathy detection by retinal image recognizing", 2020, Electrical engineering and system sciences.

[33] "A survey on Image Data Augmentation for Deep Learning" by Connor Shorten & Taghi M. Khoshgoftaar.