International Journal of Future Generation Communication and Networking
Vol. 13, No. 1s, (2020), pp. 08- 16

Fauna Image Classification using Convolutional Neural Network

Kavish Sanghvi[1], Adwait Aralkar[2], Saurabh Sanghvi[3] and Ishani Saha[4]

Students, Department of Computer Engineering, NMIMS University's Mukesh Patel School of Technology Management & Engineering, Mumbai, India
[1]sanghvi_kavish@yahoo.in, [2]adwaitaralkar@gmail.com, [3]sanghvis1000@gmail.com, [4]ishani.saha@nmims.edu

Abstract

Today, with the increasing volatility, necessity and applications of Artificial Intelligence, fields like Neural Networks, and its subsets, Machine Learning, and Deep Learning have gained immense momentum. It has become a data centric model where neural network developers are "training" the network to be "intelligent" and "independent". The training needs softwares and tools such as classifiers, which feed huge amounts of data, analyze them and extract useful features. These features are then used to observe a pattern and train the network to use similar data again the next time it is fed data. Convolutional Neural Network remains to be the most sought-after choice for computer scientists for image recognition, processing and classification. This paper proposes a fauna image classifier using convolutional neural network, which will be used to classify images of different species and animals captured in dense forest environments to achieve desired accuracy, and aid ecologists and researchers in neural network, artificial intelligence & zoological domains to further study and/or improve habitat, environmental and extinction patterns. A convolutional neural network is trained and developed for efficiently classifying these images with accurate results. Our model was successfully trained with 91.84% accuracy, and classified images with 99.77% accuracy. Complimentary technologies like VGG16, TensorFlow, Leaky ReLU, etc. have been used in training the model.

Keywords: *Image Classification, Convolutional Neural Network, VGG16, ReLU, Transfer Learning, TensorFlow*

1. Introduction

Image classification is one of the common and basic tasks in computer vision, and it has drawn a lot of attention in recent years. Data transfer in the form of images is one of the most convenient forms of presenting information for users. Images transmitted in can have background noise, distortion, occlusion, etc. Noise reduce image quality and can lead to erroneous interpretation of useful information. Noisy images are difficult to analyze both programmatically and by humans. Efficient and reliable monitoring of wild animals in their natural habitats is essential to inform conservation and management decisions regarding wildlife species, migration patterns, habitat protection, and is possible, rehabilitation and grouping species of same animals together. Processing a large volume of images and videos captured from camera traps manually is extremely expensive, time-consuming and monotonous. This presents a major obstacle to scientists and ecologists to monitor wildlife in an open environment.

Images captured in a field represent a challenging task while classifying since they appear in a different pose, cluttered background, different lighting and climatic conditions, human photographic errors, different angles, and occlusions. All these challenges necessitate an efficient algorithm for classification with most optimum accuracy. Convolutional neural network is a special algorithm of artificial neural networks and deep learning, aimed at effective image processing. The name of the network architecture is due to the presence of convolution operation. Convolution consists in that each fragment of the image is multiplied by convolution matrix, the result is summed and written to the

ISSN: 2233-7857 IJFGCN

analogous position of the output image. The work of a convolutional neural network is usually interpreted as a transition from specific image features to more abstract details, and further to even more abstract details, up to highlighting high-level concepts.

Convolutional Neural Network is a deep learning algorithm which can take an input image, assign importance to various aspects in the image and be able to differentiate one from another. Convolutional Neural Networks are typically used for image classification and recognition because of its high accuracy. Convolutional Neural Networks are also much more flexible and can adapt to the new incoming data as the dataset matures. Classification using convolutional neural network can also be applied to object detection. In recent years, multilayer neural networks have been used successfully in classification, pattern recognition, learning and decision making.

2. Literature Survey

2.1. Convolutional Neural Network

The authors of this paper[1] recommends method for automated underwater fish species classification. Popular approaches emphasis on classification of fishes outside of water because underwater classification carries several challenges such as background noises, distortion, object discrimination, image quality and occlusion. The proposed method suggested implementation of removing the noise in the dataset. Implementation of image processing before the training step helps to eliminate the underwater obstacles, dirt and non-fish bodies from the images. The following step uses Deep Learning approach by implementation of Convolutional Neural Network for fish species classification. Comparison of ReLU, SoftMax and tanh activation functions was performed and ReLU activation function was found to be highly accurate.

2.2. Transfer Learning

The paper[2] uses transfer learning to fine-tune the pre-trained network parameters for image classification. The authors have performed two experiments on two state of the art databases viz.: GHIM10K and CalTech256. The first experimentation analyzes the performance of VGG19 architecture for image classification task. Along with VGG19, the performance analysis of AlexNet and VGG16 on GBHIM10K database was compared. The second experimentation comprises the usage of CalTech256 database for performance evaluation of VGG19 architecture for image classification task. The performance of the AlexNet, VGG16, and VGG19 model was compared along with Recall, Precision, and F1-Score and with the results of both experimentations, VGG19 was found to be the most accurate.

2.3. Support Vector Machine

The paper[3] compares the performance of Support Vector Machine with Neural Network. The neural networks cannot discharge from its own limitations including the local optimum or the dependence on the input sample data. The authors suggest to implement support vector machine, main idea is to build a hyperplane as the decision surface, is introduced to solve the problems. The algorithm creates a line or hyperplane which separates the data into classes. Support vector machine is a sophisticated and powerful algorithm. The results of the experiments proved that the accuracy of the classification by support vector machine is more exceptional than the neural network. Support vector machine reveals better effect on the single classification than neural networks.

2.4. K-Nearest Neighbor

ISSN: 2233-7857 IJFGCN

The paper[4] discusses about implementation of K-Nearest Neighbors for text classification which is based on shared nearest neighbor technique, combining the BM25 similarity calculation method and the neighborhood information of samples. The proposed method obtained the best score in English text classification, but the results are still not perfect as there are few considerations that can be correctly for better results. Primarily, large structured set of text processing is too uneven without careful feature selection, and the large feature space confused the topic information of patent, which consequently weakened performance of the system to a certain extent. Furthermore, the problem of uneven density of corpus is not considered, resulting in wrong category decision-making for topics whose training data are inadequate. However, the system considered samples neighborhood information to amend the weight of each search result so as to objectively assign higher weight to the sample which is more similar with the topic, and at the same time avoided the phenomenon that the similar samples with lower ranks are severely punished because of the location. It is more reasonable for category decision-making.

2.5. Random Forest Algorithm

The paper[5] dealt with detection of transactional frauds caused related to credit cards. The proposed model uses RFA. In proposed system, it was used for finding and reporting such fraudulent transactions and the accuracy associated with them. Decision trees are used for classification of the dataset (supervised learning). For comparison and analysis, dataset was split into Training and Testing categories. When Random Forest Algorithm was applies again, a confusion matrix is obtained in which entire dataset is classified into 4 categories. Performance analysis is the next step after classification. The authors arrived at the accuracy of these fraudulent transactions and represented it graphically. The confusion matrix basically takes and collates most accurate values directly from different decision trees, thereby reducing computational complexity of making multiple decision trees. Random Forest Algorithm is a lucrative technique due to its efficiency and the ability to perform classification and regression, both.

2.6. VGG16

The paper[6] discusses VGG16 applications for Plant Image Classification along with Data Augmentation and Transfer Learning, where it uses transfer learning and convolutional neural network to classify the plant species. Leaf images are used instead of its flower counterparts as its low-level features such as color, shape, etc; and are typically used in other plant recognition models. This acts a major disadvantage is we use only leaf images as a sole feature/parameter to classify/recognize different species of plants. Data Augmentation, dropout and transfer learning can effectively help in reducing one of convolutional neural networks most computationally cumbersome problem - overfitting in small datasets. It uses a VGGnet model which was trained on ImageNet data-set. Generation of more samples to help model train better is supported by Data Augmentation. Existing training samples undergo some basic transformations too. Data augmentation ensures that the model does not "see" the same picture twice during training, thereby reducing overhead and the model is exposed to many other aspects of data. This, finally helps in generalizing model better.

ISSN: 2233-7857 IJFGCN

2.7. TensorFlow and Denoising Convolutional Neural Network

In this proposed infocommunication system[7] a DnCNN-denoiser is used. Such a denoiser uses the signal sent by receiver for filtering. Denoising Convolutional Neural Network can handle blind Gaussian noise with unknown noise levels too. Its ability to be easily retrained makes it a convenient denoiser. It predicts the residual image accurately. Convolutional Neural Network's advantage lies in dynamic, real-time filtering of images transmitted to a model through the network. Fine-tuning a large and diverse image dataset for special tasks can be achieved using a pre-trained network. Thus, Convolutional Neural Network provide flexible solutions to modern denoising problems in infocommunication systems. Comparative noise cancellation can be achieved using a combination of such Convolutional Neural Network/Denoising Convolutional Neural Network tools.

2.8. TensorFlow and Convolutional Neural Network

The paper[8] proposed removal of nonuniform spatial trends acting on the background which were superimposed on mammographic images, and two representative features were selected and were calculated - entropy of an image and the standard deviation. These features represent the texture photo. Instead of directly inputting the image to the neural network. These representative features were presented to the network instead of passing the entire data-set directly to the neural net to learn and test. This helps in reducing computational overhead of training vis-à-vis early error control or inconsistency in dataset even after preprocessing of data.

3. Proposed Methodology

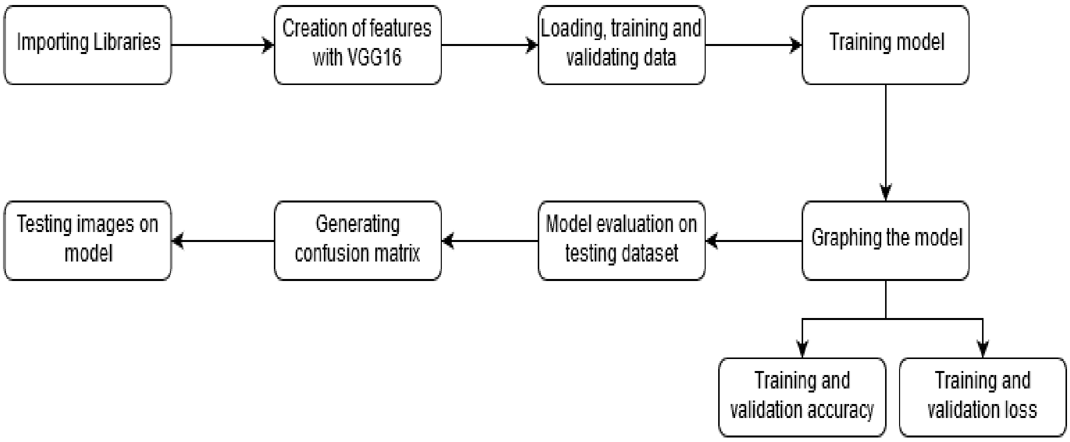

Figure 1. Design of proposed neural network

Here, we present a methodology for the classification of fauna images, which will help ecologist and scientists to further study and/or improve habitat, environmental and extinction patterns. Figure 1 displays the proposed design of model for Fauna Image Classification using Convolutional Neural Network. The Aminal-10 dataset[9] used for the classification is taken from Kaggle. We are using Convolutional Neural Network with Leaky ReLU activation function and VGG16 architecture for our model. The initial step aims at creation of features with VGG16 model. Application of Image Processing along with Loading, Testing, Training, and Validating the dataset before the training step helps to remove the noise, obstacles, distortion and dirt from the images. The next step uses Convolutional Neural Network along with Leaky ReLU to train the model to accurately and precisely classify animal classes. In order to avoid the problem of Dying ReLU, where some ReLU neurons essentially die for all inputs and remain inactive no matter what input is supplied, here no gradient flows and if large number of dead neurons are there in a neural

ISSN: 2233-7857 IJFGCN

network its performance is affected. To resolve this issue, we make use of what is called Leaky ReLU, where slope is changed left of x=0 and thus causing a leak and extending the range of ReLU. After training the model, we graph the model's training and validation accuracy and loss to have insights about how well the model is trained. Lesser the loss, more the accuracy. The next step is to generate confusion matrix to have exact details about how correctly the model is trained and classifying, as we cannot only rely on the accuracy. Lastly, we tested our model with sample data and found it to be accurately classified. Below is the detailed explanation of each modules implemented.

3.1. Importing the libraries

This module is used for importing the required libraries for the neural network model. We made use of various libraries such as Panda library is used for providing high-performance, easy-to-use data structures and data analysis, NumPy for mathematical and logical operations on arrays can be performed, Keras is designed to enable fast experimentation with deep neural networks, it focuses on being user-friendly, modular, and extensible, and many more.

3.2. Creation of weights/features with VGG16

This module is used for resizing the images with width=224 and height=224 as per VGG16 model requirements. A bottleneck file was created to encourage the network to compress feature representations to best fit in the available space, in order to get the best loss during training. We categorized the image dataset into train, validate and test and loaded it in the network. We also loaded a pre-trained VGG16 model. This module creates weights/features with VGG16 model to finetune the neural network to perform well with the input data. The network successfully found 13412 train image which belonged to 6 animal classes in 95:04 minutes. This module creates weights/features with VGG16 model to validate and test data. Data quality is vital to ensure accuracy and reliability. The validate network successfully found 2549 validated images which belonged to 6 animal classes in 22:26 minutes, and the test network successfully found 1845 test images which belong to 6 animal classes in 13:43 minutes.

3.3. Loading Training, Validating, and Testing Data

This module loads training, testing and validation dataset for testing the model. Training data is the actual dataset that we use to train the model. The neural network model "*observes"* and "*learns*" own its own from the training data. Testing data is the sample of data that is used to provide an unbiased evaluation of the best final model on the training dataset. Validation data is the sample data that is used to provide an unbiased evaluation of a model on the training data while tuning model hyperparameters. The evaluation becomes more biased on the validation dataset is incorporated into the model configuration. The training network successfully found 13412 train images which belonged to 6 animal classes. The testing network successfully found 1846 train images which belonged to 6 animal classes. The validation network successfully found 2549 validated images which belonged to 6 animal classes.

ISSN: 2233-7857 IJFGCN

3.4. Training Model

This module trains the neural model for animal image classification. Training the neural network involves finding a set of weights to best map inputs to outputs. Loss function states how good our neural network works for a certain task. The problem of training is equal to the problem of minimizing the loss function. The procedure used to carry out the learning process in a neural network is termed as optimization algorithm. A single appearance of the entire data set is referred to as an "epoch". An epoch defines the number of times the algorithm sees the entire dataset. Each time the algorithm has seen all samples in the dataset, an epoch is completed. Neural network training algorithms involve making multiple presentations of the entire data set to the neural network, includes one forward pass and one backward pass of all the training. 7 epochs took 01 minutes 15 seconds, resulting in 91.84% accuracy and loss of 0.2707. Keras offers a method to summarize a model. The summary is textual and includes detailed information about:

- The layers and their order in model.
- The output shape of each layer.
- The number of parameters (weights) in each layer.
- The total number of parameters (weights) in the model.

The summary can be created by using the summary() function on the model that returns a string that in turn can be printed.

```
WARNING:tensorflow:From C:\Users\KAVISH\Anaconda3\lib\site-packages\tensorflow\python\ops\math_ops.py:3066: to
_int32 (from tensorflow.python.ops.math_ops) is deprecated and will be removed in a future version.
Instructions for updating:
Use tf.cast instead.
Train on 13412 samples, validate on 2549 samples
Epoch 1/7
13412/13412 [==============================] - 13s 976us/step - loss: 0.7251 - acc: 0.7372 - val_loss: 0.4256
- val_acc: 0.8474
Epoch 2/7
13412/13412 [==============================] - 10s 728us/step - loss: 0.4087 - acc: 0.8612 - val_loss: 0.3454
- val_acc: 0.8858
Epoch 3/7
13412/13412 [==============================] - 9s 708us/step - loss: 0.3293 - acc: 0.8879 - val_loss: 0.3806 -
val_acc: 0.8674
Epoch 4/7
13412/13412 [==============================] - 10s 715us/step - loss: 0.2827 - acc: 0.9043 - val_loss: 0.2674
- val_acc: 0.9157
Epoch 5/7
13412/13412 [==============================] - 10s 756us/step - loss: 0.2416 - acc: 0.9203 - val_loss: 0.2763
- val_acc: 0.9109
Epoch 6/7
13412/13412 [==============================] - 10s 737us/step - loss: 0.2036 - acc: 0.9295 - val_loss: 0.3774
- val_acc: 0.8847
Epoch 7/7
13412/13412 [==============================] - 10s 756us/step - loss: 0.1822 - acc: 0.9409 - val_loss: 0.2707
- val_acc: 0.9184
2549/2549 [==============================] - 0s 181us/step
[INFO] accuracy: 91.84%
[INFO] Loss: 0.27072709791651656
Time:  0:01:15.003557
```

Figure 2. Training model with epochs and accuracy

3.5. Graphing the model

This module graphs the training and validation accuracy and loss for each epoch. During an epoch, the loss function is calculated across every data item and it is guaranteed to give the quantitative loss measure at the given epoch and plotting curve across each iteration only gives the loss on a subset of the entire dataset.

ISSN: 2233-7857 IJFGCN

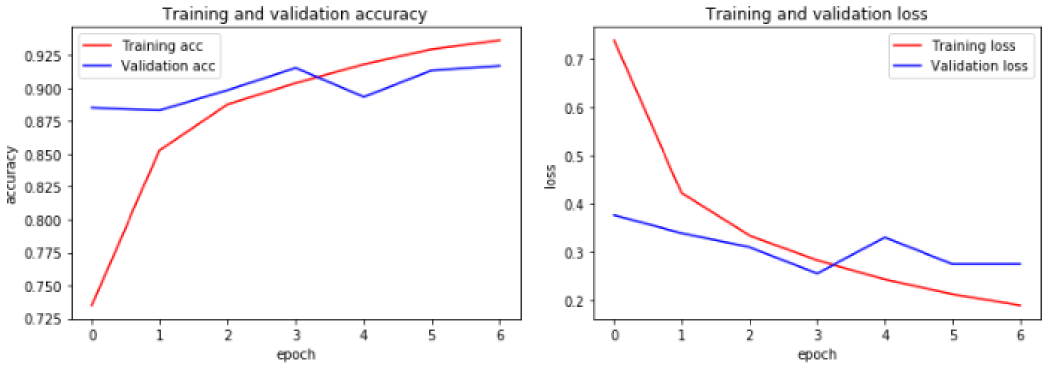

Figure 3. Training and Validation accuracy and loss graph

3.6. Model evaluation on testing dataset

Classification metrics lists the precision, recall, F1-score, and support for fauna species, along with their micro and macro average and weighted and samples average.. Precision is a good measure when classification accuracy is not a good indicator of your model performance, when the class distribution is imbalanced. Recall is the fraction of samples from a class which are correctly predicted by the model. The combination of precision and recall is called F1-score.

- $Precision = \frac{True\_Positive}{(True\_Positive + False\_Positive)}$
- $Recall = \frac{True\_Positive}{(True\_Positive + False\_Negative)}$
- $F1 - Score = \frac{2*Precision*Recall}{(Precision + Recall)}$

3.7. Generating confusion matrix

We implemented confusion matrix as we cannot only rely on the accuracy. Confusion matrix is a tabular visualization of the model predictions as opposed to the ground-truth labels. Each row of confusion matrix represents instances in a predicted class and each column of confusion matrix represents the instances in an actual class.

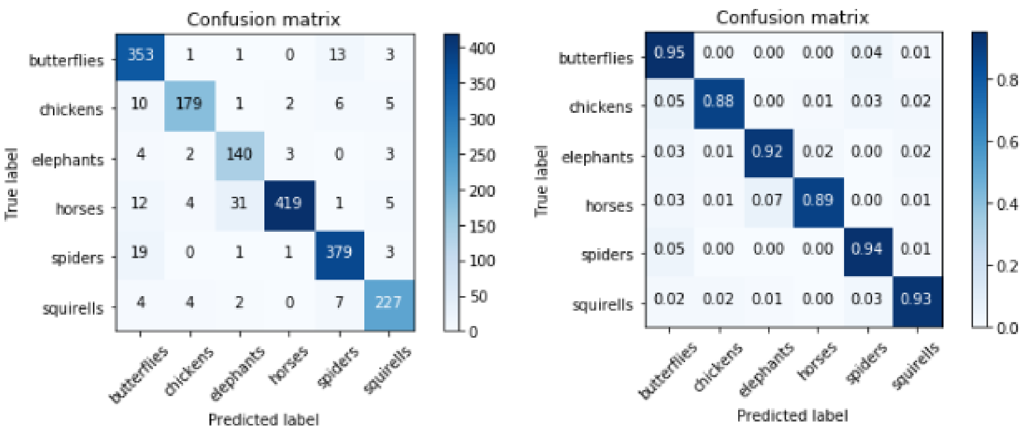

Figure 4. Confusion matrix with and without Normalization

3.8. Testing images on model

ISSN: 2233-7857 IJFGCN

Finally, the last stage is the testing of the trained model on a sample image to check whether the neural network is trained accurately and is working error free. The image is fed in the neural network model and the model accurately classifies the animal class.

Figure 5. Output of test sample images with accuracy

In order to get the best results for object feature identification and training of the convolutional neural network, it is important to provide input image with enhanced features as training sample. The goal of the training algorithm is to train a neural network such that the error is minimized between the network output and the desired output.

4. Result & Observations

The proposed model was coded in Python and tested in Jupyter Notebook on the Animal-10 dataset[9] which contains 26,179 images from 10 animal classes. The model could achieve an accuracy of 91.84% for 6 animal classes. The neural network could successfully identify the animal image and classified it to the correct animal class with an accuracy of 99.77%. The model successfully detected test 1845 images from 6 animal classes, 13412 training images from 6 animal classes, and 2549 validated images from 6 animal classes. Table 1 provides insights into the number of images for each animal classes.

Table 1. Animal Classes in the Dataset

Animal	Number of Images
Butterflies	2112
Chickens	3098
Elephants	1446
Horses	2623
Spiders	4821
Squirrels	1862
Total	**15,962**

ISSN: 2233-7857 IJFGCN

5. Conclusion & Future Work

The proposed method for classification of fauna images using convolutional neural network gives an accuracy of 91.84%. It addresses the implementation of convolutional neural network with Leaky ReLU for fauna image classification. The efficiency of various activation functions and convolutional neural network architectures were compared, and we found ReLU activation function and VGG16 model to be most accurate and appropriate for image classification. The neural network is trained to classify image of an animal and help identify animal class. We have trained our neural network in such a way that it can train new animal class by simply feeding the neural network with minimum 1000 labelled images for training dataset and more than 300 labelled images for validation dataset. Concluding, the proposed fauna image classification using convolutional neural network can be used extensively for fauna image classification which will aid ecologists and researchers to further study and/or improve habitat, environmental and extinction patterns.

Future work includes:

- Developing a simple yet efficient user-interface for the project for easy use for ecologist, photographers, computer researchers
- Improvising the classification accuracy, precision and reduction in terms of error, training and testing time
- The image classification model can be improved in future, by including low-level features such as shape and spatial location features apart from optimizing the weights and learning rate of the neural network.

When these improvements are incorporated in the classification system, it would help further improve the performance and be useful for applications meant for the explicit classification system.

References

[1] Dhruv Rathi, Sushant Jain, Dr. S. Indu, "Underwater Fish Species Classification using Convolutional Neural Network and Deep Learning", International Conference of Advances in Pattern Recognition, 2017.

[2] Manali Shaha, Meenakshi Pawar, "Transfer learning for image classification", 2nd International conference on Electronics, Communication and Aerospace Technology (ICECA), 2018.

[3] Xiaowu Sun, Lizhen Liu, Hanshi Wang, Wei Song, Jingli Lu, "Image Classification via Support Vector Machine", 4th International Conference on Computer Science and Network Technology (ICCSNT), 2015.

[4] Yun-lei Cai, Duo Ji, Dong-feng Cai, "A KNN Research Paper Classification Method Based on Shared Nearest Neighbor", NTCIR-8 Workshop Meeting, Tokyo, Japan, 2010.

[5] M. Suresh Kumar, V. Soundarya, S. Kavitha, E.S. Keerthika, E. Ashwini, "Credit Card Fraud Detection Using Random Forest Algorithm", 3rd International Conference on Computing and Communications Technologies (ICCCT), 2019.

[6] Mohamad Aqib Haqmi Abas, Nurlaila Ismail, Ahmad Ishan Mohd Yassin, Mohd Nasir Taib, "VGG16 for plant image classification with transfer learning and data augmentation", International Journal of Engineering & Technology, 2018.

[7] Oleksii Sheremet, Kateryna Shernemet, Oleksander Sadovoi, Yuliia Sokhina, "Convolutional Neural Networks for Image Denoising in Infocommunication Systems", International Scientific-Practical Conference (Problems of Infocommunications, Science and Technology-PICS&T), 2018.

[8] H. Fujita, K. Horita, T. Endo, C. Kido, T. Ishigaki, "Classification of Breast Tumors in Mammograms using a Neural Network: UtiIization of Selected Features", International Conference on Neural Networks (IJCNN-93-Nagoya, Japan), 1993.

[9] Kaggle.com, Animals-10 dataset [Online], Available at: https://www.kaggle.com/alessiocorrado99/animals10.

ISSN: 2233-7857 IJFGCN