



Big Data Processing for Generic Clarification of Heterogeneous Images

Olanrewaju E. Abikoye¹, Abdullateef O. Alabi^{2*} and Y. O. Olaboye¹

¹Department of Mechanical Engineering, Federal Polytechnic Offa, P.M.B.420, Offa Kwara State, Nigeria.

²Department of Computer Engineering Technology, Federal Polytechnic Offa, P.M.B.420, Offa Kwara State, Nigeria.

Authors' contributions

This work was carried out in collaboration among all authors. Author OEA designed the study, performed the statistical analysis, wrote the protocol, and wrote the first draft of the manuscript. Author AOA managed the analyses of the study. Author YOO managed the literature searches. All authors read and approved the final manuscript.

Article Information

DOI: 10.9734/AJRCOS/2020/v6i430167

Editor(s):

(1) Dr. Hasibun Naher, BRAC University, Bangladesh.

Reviewers:

(1) Ravindra Babu Jaladanki, PVP Siddhartha Institute of Technology, India.

(2) Arvind K. Sharma, India.

Complete Peer review History: <http://www.sdiarticle4.com/review-history/64020>

Method Article

Received 10 October 2020
Accepted 16 December 2020
Published 02 January 2021

ABSTRACT

Most industries around the globe make use of image processing to improve their productions. On the other hand Big Data Processing is a big dataset; this required fast method to processing irrespective of Generic nature, therefore Clarification of heterogeneous images can improve the integrity of any system design. To avoid waste of time and energy, it is necessary to classify images. Big Data Processing for Generic Clarification of heterogeneous images provides fast, accurate and objectives results. In this study, the researchers classified into three category using resnet50 techniques for training dataset images. The outcome of the research is analyzing these techniques and comparison analysis on different existing image data sets as pre-trained data and test data as sample images for decision making based on their limitations and strengths.

Keywords: Confusion; big data processing; generic clarification; images.

*Corresponding author: E-mail: Kitfad4u@gmail.com;

1. INTRODUCTION

Image processing techniques are widely used in various applications today, since its relevance in industries to increase production, enhance security and deployment to automation systems. In its simplicity, the application of convolutional neural networks provides autonomous ways to dealing with large dataset for making kind of decisions based on classification support vectors which play a vital role [1].

In a clear perspective, the objective of image classification is complex characteristic modalities to categories and label group of pixels or vector within an image according to Pralhad [2] based on specific rules. The algorithm of CNN depend on conditions, in a figurative aspect of deep convolutional neural network for Image Classification on CUDA Platform according to Chen et al. [3] done in large-scale categories and was experimented by classified image using hierarchical from the database. According to Deng Jia et al. [4], interpreted image classification in medical perspective, which focused on medical image, the image was classified into three “high, medium and low” the result obtained is very précised classification layers scores 2000 and above.

Image Classification is more relevant in Remote Sensing, this involves the process of assigning pixels in the image and this aids for the classes of interest or categorise under certain categories. Instances of this categories according to Alex et al. [5] includes water, urban, forest, agriculture, rocky areas, cloud, shadow, grassland, built-up areas, water body, green vegetation, bare soil, and many more.

Latest research in deep gaining knowledge of has been largely inspired by way of the manner our mind works. While thinking of it, it is good to know that with a given entrance, human brain processes functions that allow us to recognize of the sector that surrounds us. Therefore the evolution of image classifications varies and its go along applications. To recap from high-level perspective, the key changes that were brought in the evolution yield tremendous success and each methods really matter because their architectures are crucial in image classification.

A Le Net, method is concerned with the used a raw multi-layer perceptron connected to each image pixel. AlexNet, VGGNet, GoogLeNet, ResNet, DenseNet, as this proves as a major

pattern observed shown in Fig. 1. and these networks are designed to be deeper and deeper. In this view, its’ architectures are critical for us, not best due to the fact many challenges rely upon the tasks we will perform with them. in fact, the design of the networks themselves factors us out to the representation that researchers have been looking for, which will better analyze from the data. Therefore, another major point in CNN is the increasing use of connections between each layers of the network and this help for producing diverse features, increases productivity output which is more useful for gradient propagation.

The research is to use a pre-trained convolutional neural network for generic classification of heterogeneous image sets. A dataset is collected which contains objects categories for the data sets and this demands the needful use of a pre-trained convolution neural network to pacify the images of the data sets. Convolution Neural Network (CNN) is a powerful machine learning technique, which requires a lot of collection of images for full train in the network according to Karen and Zisserman [6].

The larger the collection, the richer that the CNN learnt, this performs hand-crafted features such as HOG, LBP, or SURF. Training as a CNN of a large collection of divergent images is not an easy task however, there is an easy way. The use of pre-trained CNN saves a huge amount of time and effort and this is the main objective of this research work.

The implementation workflow begins with creation of a directory folder that constrains the sub-folder that holds the entries image categories of the data-sets. Working with the MATLAB environment, the main folder must be loaded with a script function fullfile() to enable script access to all data-sets. In the case, three categories data-set is selected out of 101 data-set in the main path file for CNN pre-train and this done by enabling path files for this data-sets “airplanes, ferry and laptop are selected for the pre-train network. *IMDS* helps to manage data since it operates in file location, it does not level the images into the memory until they are read and this makes it efficient to use. The syntax MATLAB functions for managing data as follow;

```
%Load training data
categories = {'airplanes','ferry','laptop'};
%.....code 1
```

```
rootFolder = 'cifar10Train';
imds = imageDatastore(fullfile(rootFolder,
categories), %%...code 2
'LabelSource', 'foldernames');
```

The images are located in the categories according to code 1 and the IMDS for data management taking as code 2.

The imageDatastore has two arguments, the first is rootFolder, categories for the images location and the second categories is name and value pair 'LabelSource', 'folder names' that means the images will be level as their folder names. So IMDS variables contain the images and their corresponding categories levels.

The next action to determine the numbers of images at categories levels, then a function countEachLabel() using the script function as follow;

```
Tb1 = functioncountEachLabel(imds)
%%.....code 3
```

This line code returns airplanes 800, ferry 67 and laptop 81 respectively. In this regard, the number of images in each category is not the same and this is not an application of an equal matrix. Therefore to resolve to make all categories equal in the data set, apply the “min” function trimmed the categories follow by a function.

Split Eachlevel(); % equal as written below;

```
minSetCount = min(Tb1); %.....code 4
minSetCount = min(:,2); %.....code 5
splitEachlevel(imds, minSetCount,
'randomize'); %.....code 6
```

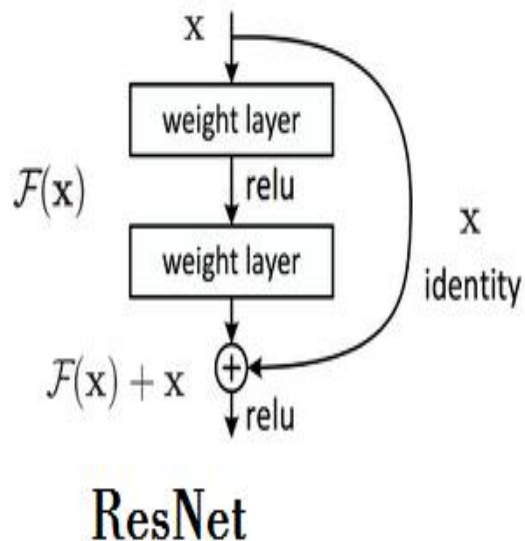
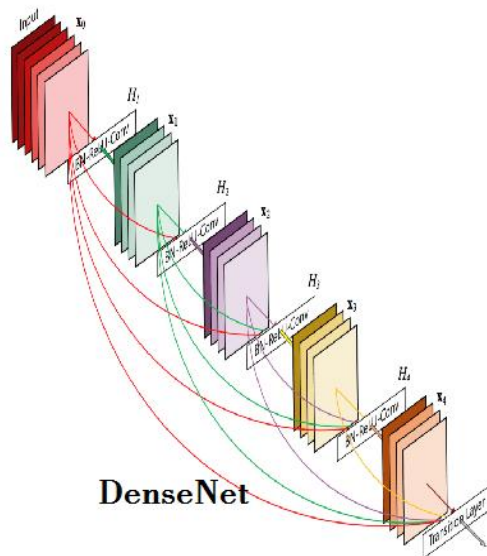
This code 5 is only referred to as the second column on the TB1 results while the code 6 split each level randomly with equal data set.

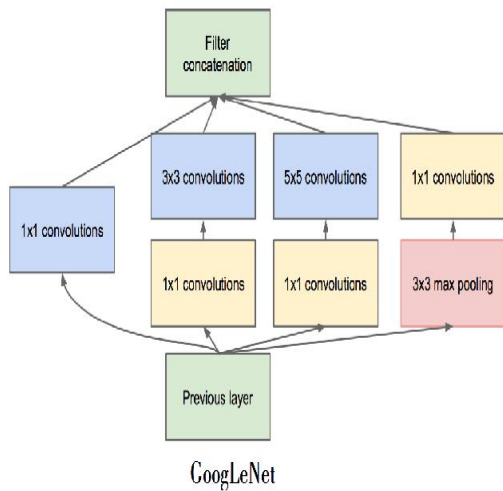
Therefore the sequence to find the first categories label in the data set for airplanes, ferry and laptop using a function find() as written below;

```
Airplanes = find(imds,Labels=='airplanes', 1);
%%.....code 6
Ferry = find(imds,Labels=='ferry', 1);
%%.....code 7
Laptop = find(imds,Labels=='laptop', 1);
%%.....code 8
```

Next, is rendering the result in figure format using subplot () function to plot airplanes, ferry and laptop respectively and this is presented in Fig. 1.

```
Figure
Subplot (2,2,1);
Imshow (readimage(imds, Airplanes));
Subplot (2,2,2);
Imshow(readimage(imds, Ferry));
Subplot (2,2,3);
Imshow(readimage(imds, Laptop));
```

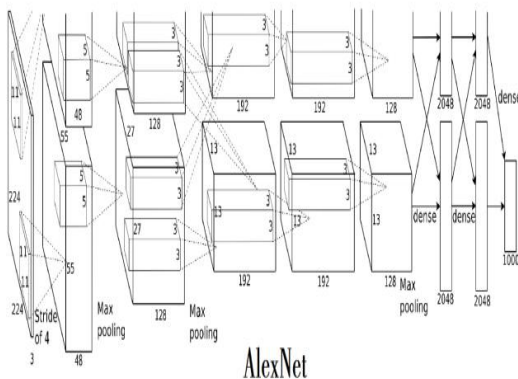




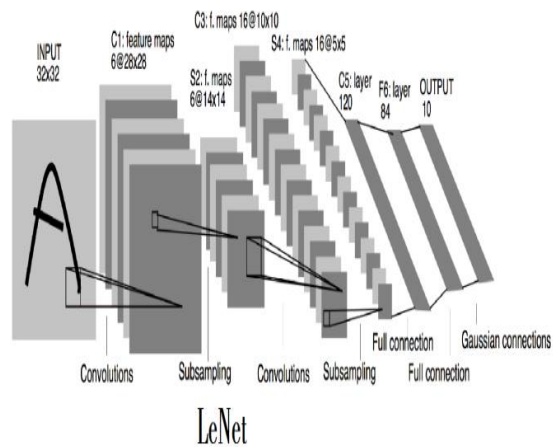
GoogLeNet

input (224 × 224 RGB image)					
conv3-64	conv3-64	conv3-64	conv3-64	conv3-64	conv3-64
	LRN	conv3-64			
maxpool					
conv3-128	conv3-128	conv3-128	conv3-128	conv3-128	conv3-128
		conv3-128			
maxpool					
conv3-256	conv3-256	conv3-256	conv3-256	conv3-256	conv3-256
conv3-256	conv3-256	conv3-256	conv1-256	conv3-256	conv3-256
maxpool					
conv3-512	conv3-512	conv3-512	conv3-512	conv3-512	conv3-512
conv3-512	conv3-512	conv3-512	conv3-512	conv3-512	conv3-512
maxpool					
conv3-512	conv3-512	conv3-512	conv3-512	conv3-512	conv3-512
conv3-512	conv3-512	conv3-512	conv1-512	conv3-512	conv3-512
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

VGGNet



AlexNet



LeNet

Fig. 1. CNN classes architectures structures with layer patterns

Therefore an equal number of images in the data set is achieved in the result above, now time to load pre-trained convolutional neural networks. Therefore several methods of CNN that have gained popularities on network training and resnet50 is one of them, a deep convolutional activation for generic feature recognition according to Donahue Jeff et al. [7]. Help to train a full version. It has been trained on images net data set. This has 1000 object categories and 1.2 million trained dataset. The resnet50() was release with MATLABb 2016 version, so using resnet50() function to begin CNN as follow;

```
net = Resnet50();
Figure
Plot(NET)
```

Title('Architecture of ResNet-50')
Set(gca, 'Ymim', [150, 170]);

The line code presents the architecture of resnet 50() as shown in Fig. 2.

In the Fig. 2b shown the input, convolutional layer, activation and relu function respectively, the architecture in the output figure is presented as the network architecture. Each network is accessible to determine the type of input it accepts in respect to image size and type. Taken an instances of the first layer in the network, the command function net.Layers(1) allow access to the first layer only, this reveal associate properties as input layer properties as presented below;

```
ImageInputLayerwithproperties;
Name: 'input_1'.....indicate name of
the image
InputSize: [224 224 3].....this implies that
is 224 by 224 of 3 channel image i.e RGB
image
Hyperparameters
DataAugmentation: 'none'
Normalization: 'zerocenter'
Average Image: [224 x 224 x 3 single]
```

Therefore to inspect the last layer in the network, a function name `net.Layers(end)`, the return of the function is given that;

```
ClassificationOutputLayerwithproperties:
Name: 'ClassificationLayer_fc1000'
Classes: [1000 x 1 categorical]...This means
there are 1000 classes to be train by the
network
Hyperparameters
```

2. RESEARCH METHODS AND MATERIALS

In other way, determine the classes can be derive with a function `numel(net.Layers(end).ClassName)` and the return value 1000 on command window. In this scenario, the research focus to solve different classification tasks on `cifar10Train` dataset. In order to achieve this task, there is need to prepare the training and the test images, therefore dividing the `cifar10Train` dataset into `trainingSet` and `testSet` using the function called `splitEachLabel` using the images stored in `imds`. Therefore, 30% of the dataset is set aside for the training while 70% for validation using random selection as written below;

```
[trainingSet, testSet] = splitEachLabel(imds,
0.3, 'randomize'); %.....code 9
```

The code 9 in above is to perform training of a dataset, It's a remark that different CNN operators accept different image dimensions, in this regard. There is a need to perform sequential image processing before sending the result to CNN. The `resnet 50()` take 224 by 224 with three channel image is used to achieve the function written in code 10 bellow;

```
imageSize = net.Layer(1). InputSize;
%.....code 10
```

The return of the function gives image size as 224 by 224 with 3 channels and many engineers override this image while processing and its the requirement before dataset training. In other to prevent such occurrence, a function name `augmentedImageDatastore()` is employment to resize and congress any gray scale image to RGB image of the line. There are other usefulness function of augmented Image Data Store "`augmentedImageDatastore()`" as used in this work, this is used to render additional data augmentation when use for network training. In this view, its requirement to pass arguments to resize image set;

```
augmentedTrainingSet =
augmentedImageDatastore(imageSize,
trainingSet, 'ColorPreprocessing', 'gray2rbg')
%.....code 10
```

In the code 10 written above, the arguments required image resize, training of dataset, the use of color pre-processing to convert gray image to RGB. In the other way round, code 11 shown function to the data set image to be test.

```
augmentedTestSet =
augmentedImageDatastore(imageSize,
testSet, 'ColorPreprocessing', 'gray2rbg')
%.....code 11
```

Each layer of the convolutional neural network produces a response, however not all layers on the network are suitable for image extraction. The layer at the beginning captures basic image features such as aging and blurs. This can be seen through visualization of the network filter weight from the first convolutional layer and this more understanding that features extracted from the first convolutional neural network work out perfectly for image recognition. So to get the weight from the first convolutional neural network;

```
w1 = net.Layers(2).Weights; %.....code 12
```

The response from code 12 in above store value in `w1` as a matrix data set, there is needed to convert it to image in order to visualise it. This is done using a MATLAB function `mat2gray()` as written in code 13. Thereafter plot to visualise weight (`w1`) of the first convolutional layer.

```
w1 = mat2gray(w1); %.....code 13
```


The features in the output Fig. 4 are processed by a deeper network layer which combine the early features to wall the high level image features. These high level features are better suited for recognition cast simply they combine all primitives features into a retired image representation as done according to Kim et al. [8] findings. Using activation method which can easily extract features from one of the deeper layers and can choose any layer however; this research method will extract features from the layer right before classification layer. In resnet50(), this layer is named fc1000.

```
featureLayer = 'fc1000';
trainingFeatures =
activations(net,augmentedTrainingSet,
featureLayer, 'MinBatchSize', 32, 'outputAs',
'columns'); %.....code 13
```

The activation function in code 13 actually uses GPU, however if GPU is not available, then it uses CPU. Therefore the MiniBatchSize is set to 32 bit memory, so if the GPU runs out of memory, there is a possibility to lure the MiniBatchSize. In another study, according to Karen and Zisserman [6], precise on adaptive learning, the classification was based on weight for filtering processing and speeding yielded up 97.99 since all depends on GPU. The activation output is arranged as columns and this will speed up the clock output as Support Vector Machine (SVM) training and so in other to achieve this, level of training set is required using function trainingSetLabels in code 14 bellow;

```
trainingLabels = trainingSet.Labels;
%.....code 14
```

In order to train the SVM , a function named Fit-Class Error Correcting Output Codes(ECOC) as written on code 15 and this return full trained multiclass in output coding model.

$$K(K-1)/2 \text{ \%.....equation 1}$$

This function fitcecoc() uses a method called binary support vector machine model using one versus one coding design. Where K is a unique class labels denoted as $K = \#$ unique class labels. This required argument include training feature extracted from code 13, level of training of image level as store in a variable name trainingLabels in code 14 and the learner of SVM present to be lineal classification and taken the debug of coding to be onesall, the observation

is presented in columns as presented in code 15 bellow;

```
classifier = fitcecoc(trainingFeatures,
trainingLabs,'leanear', 'Coding', 'onesall',
'ObservationsIn', 'columns'); %.....code 15
```

The fitcecoc() returns a train model then stores the result in a variable name classifier, however, to evaluate classifier by extract features from the test set.

```
Test Features = activations
(net,augmentedTestSet, featureLayer,
'MinBatchSize', 32, 'outputAs', 'columns');
%.....code 16
```

The return value of code 16 will return a test set for the test images and demonstrate the accuracy of the trained classifier. Once the test image is passed over the classifier, compared with obtained features then can determine the accuracy of the classifier. In the scenario, a predict() function is employed and this returns a vector of predicted class level based on the trained classifier. Since classifier has been trained in code 15 above, the following arguments are required for the predict function and store the return function in a variable name *predictLabels* and this really works with Tensorflow algorithm according to [9] model.

There are other computer classifier used according to Khayatizada et al. [10] in the applications of 3D skeletons. This algorithm is used *detect convex ridges on voxel surfaces extracted from 3D scans. Using a surface classifier to separates convex ridges from quasi-flat regions*. This work prove partition techniques with best practise to used in a preprocessing step for the brain surface analysis. this proven to be a Robust from the perspective of used of CNN in Classification and Analysis of Anatomic Surfaces Using 3D.M. Khayatad [11] also use of an algorithm that for a research to quantifies and combines two visual aspects – roughness and color, this was done in order to locate the corroded area in a given image and based on this conditions, major lies on the roughness analysis include the uniformity metric which was calculated from the perspective of gray-level co-occurrence matrix. Based on the author findings, the algorithm adopted push to a large dataset of photographs of corroded material which proven to be efficient to corroded areas.

Therefore, to compare the result, the original level is obtained first using a function name `testSet.Labels`. So both `predictLabels` and `testLabels` can be compared in order to evaluate the performance of the classifier. In MATLAB, a function name `confusionmat()` is appropriate to generate the confusion matrix.

```
PredictLabels = predict (classifier,
testFeatures,'ObservationsIn', 'columns');
%.....code 17
ConfMat = confusionmat(testSet.Labels,
predictLabels); %.....18
```

The required arguments include `testSet.Labels` and `predictLabels` as used in code 18 and return highlighted value presented in Fig. 4;

When evaluating a classifier, the present state value is expected from the presented confusion matrix, therefore, a function name `bsxfun()` with

operational function divide and addition mathematical function for the confusion matrix as presented in cod 19.

```
confMat = bsxfun(@rdivide, confMat,
sum(confMat,2)); %.....code 19
```

`Sum (confMat,2)` function calculates the sum of entire rows of `confMat` matrix and puts the value of the first element 47 as shown in Fig. 5b and this gives the same result for each row. Therefore the `bsxfun()` is generated by converting the value of `confusionmat` into percentage. So, 100 percent presented in Fig. 5c. is denoted by 1, therefore calculate the mean value of the presented Fig. 5c, this can be presented in percentage accuracy in diagonal matrix. A function name `mean(diag(confMat))` returns accuracy level 0.992 for the *augmentedTestSet* classifier as presented in Fig. 5d, therefore, next to classify images on this fact.

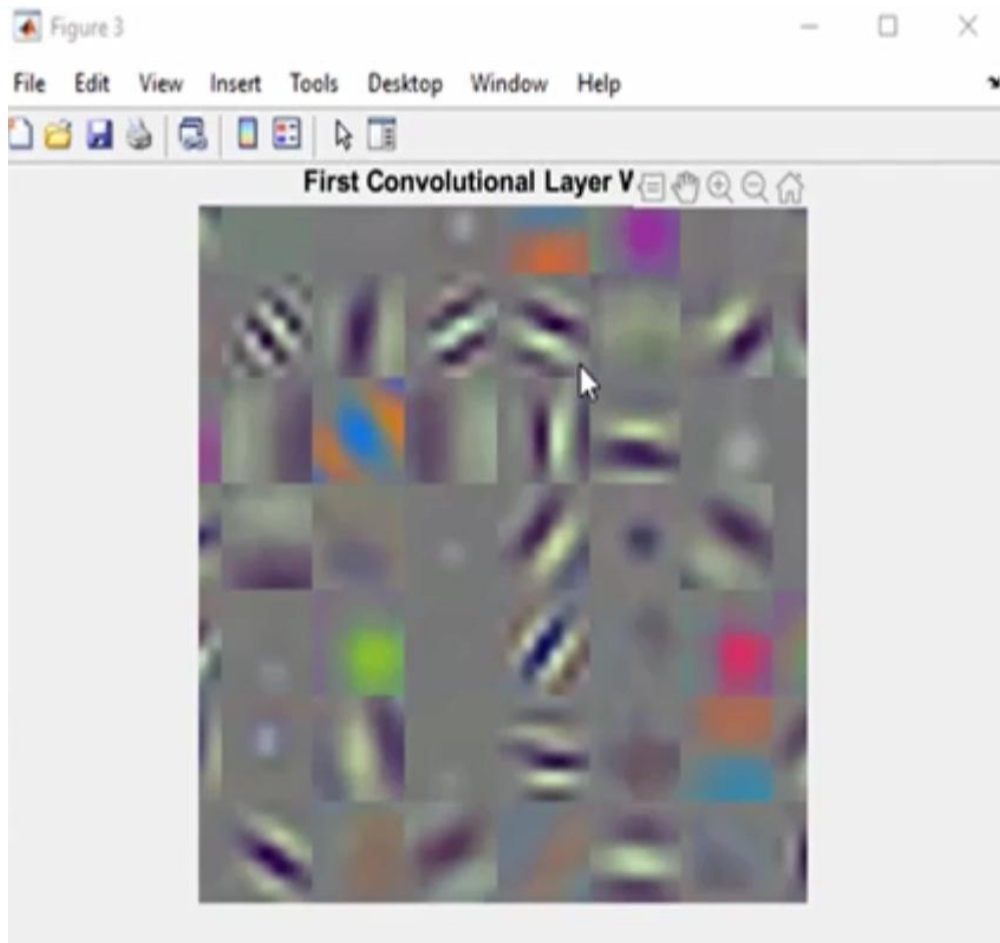


Fig. 4. First convolutional layer visual

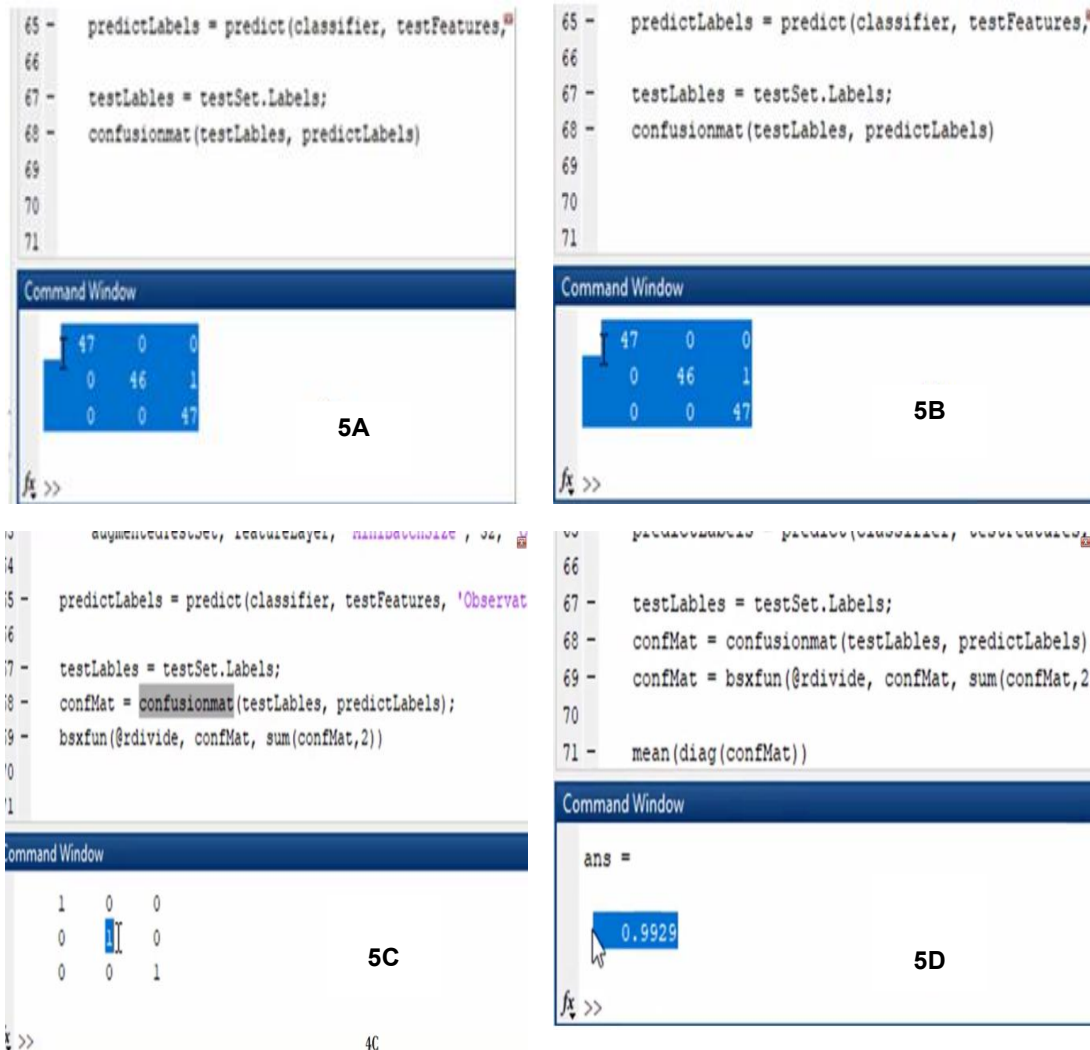


Fig. 5. a, b and c: Returned value of confusion method using test and prediction dataset

3. TESTING AND RESULTS

Recall from existing categories “aeroplanes, ferry and laptop”, a sample image of any of the categories is used for pilot test. A sample of aeroplane is downloaded from the internet and tested for train netnet. To achieve this, new sample image is store in MATLAB working directory then follow by the syntax bellow;

```

newImage= imread (fullfile('test101.png'));
ds      =      augmentedTestSet      =
augmentedImageDatastore(imageSize,
newImage, 'ColorPreprocessing', 'gray2rgb')
imageFeatures      =      activations(net,ds,
featureLayer, 'MinBatchSize', 32, 'outputAs',
'columns');

```

```

Label      =      predict(classifier,
imageFeatures, 'Observations', 'columns');

```

Finally, the output of these resultnce code testing segment can be display using sprint function as written below;

```

Sprint ('The loaded image belongs to %s
class', label);

```

Therefore, the result shown in Fig. 6a. proved that the image belongs to airplanes class, therefore other categories images i.e laptop and ferry also tested shown in Fig. 6 b and c respectively. The presentation model shown in Fig. 6d is presented as the overview of the whole network for the three classified images on convolutional neural networks.

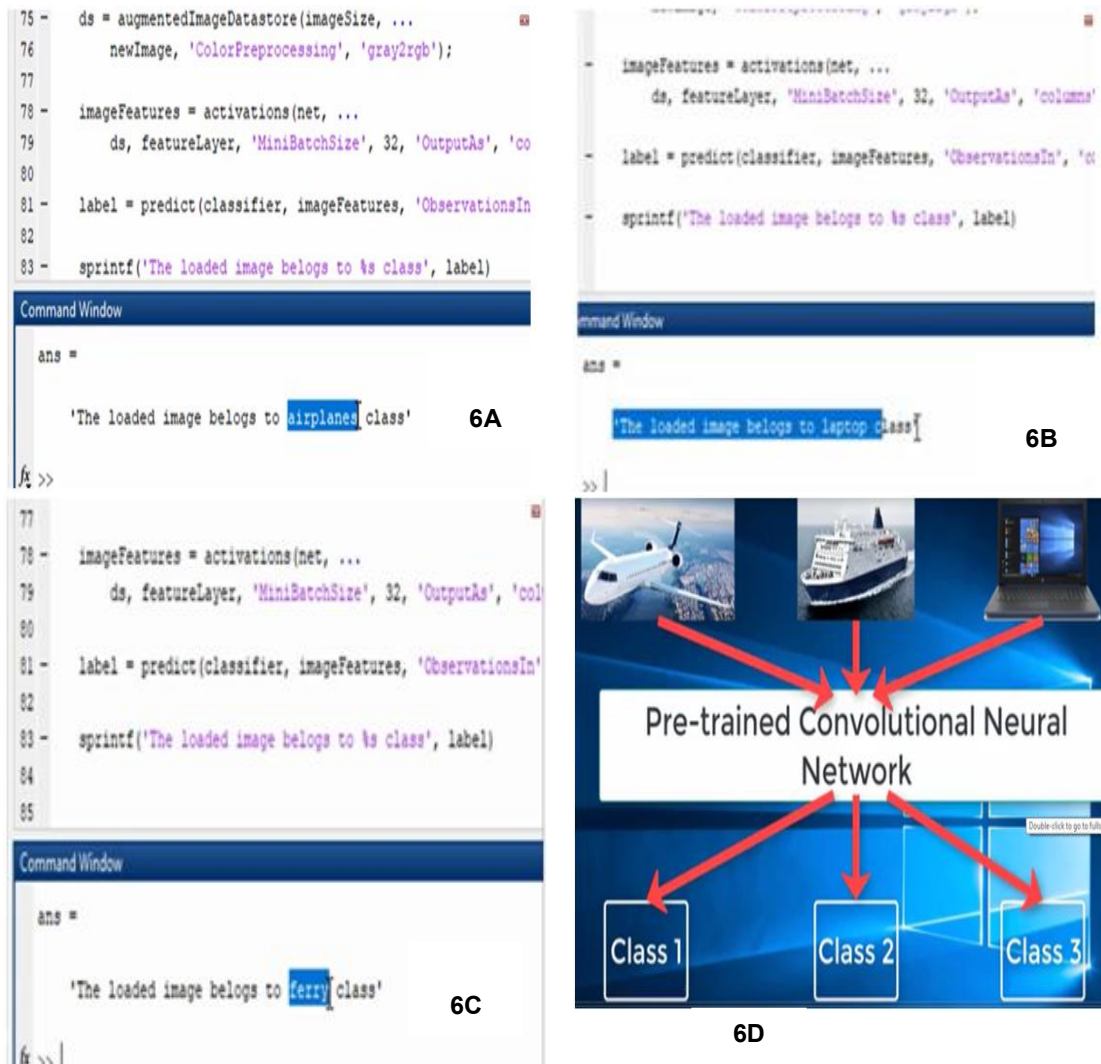


Fig. 6. Presentation model

4. CONCLUSION

This paper described general perspectives on image recognition and detection techniques to classify image dataset as presented as a pre-trained convolutional neural network. In this paper we made a comparison analysis on different existing image data set as pre-trained data and test data as test images for decision making and methods for classification of class 1, class 2 and class 3 images. So therefore, the researcher poses with the mindset that this paper can be a defined solution classifying images based on conditional approach in this field. This will provide a helpful glimpse and aid recast existing solutions for classification with their strength and limitations.

ACKNOWLEDGEMENTS

The authors acknowledge the support of Nigerian Tertiary Education Trust Fund (Tetfund) through its Institution Based Research Grant (Batch 5 RP Disbursement) that made the study possible.

COMPETING INTERESTS

Authors have declared that no competing interests exist.

REFERENCES

1. Neural networks, A comprehensive foundation, second edition by Simon Haykin; 2009.

2. Pralhad Gavali Me. Deep convolutional neural network for image classification on CUDA platform, deep learning and parallel computing environment for bioengineering systems; 2019.
Retrieved: 12-2020,
Available:<https://www.sciencedirect.com/topics/engineering/image-classification>
3. Chen PH, Chang YC, Chang LM, Doerschuk PC, Application of multiresolution pattern classification to steel bridge coating assessment. J Comput Civ Eng. 2002;16(4):244–251.
4. Deng Jia et al. Imagenet: A large-scale hierarchical image database. Computer Vision and Pattern Recognition, CVPR 2009. IEEE Conference on. IEEE; 2009.
5. Krizhevsky Alex, Ilya Sutskever, Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems; 2012.
6. Simonyan Karen, Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556; 2014.
7. Donahue Jeff et al. Decaf: A deep convolutional activation feature for generic visual recognition. arXiv preprint arXiv:1310.1531; 2013.
8. Kim HJ, Shrestha J, Kim HN, Jo GS. User action based adaptive learning with weighted bayesian classification for filtering spam mail. Lect Notes Comput. Sci Adv Artif Intell. 2006;790–798.
9. Tensor Flow Hub, Retraining an image classifier; 2020.
Retrieved: 13-12-2020,
Available:https://www.tensorflow.org/hub/tutorials/tf2_image_retraining
10. Khayatizada MB, De Pue a L, De Waele W. Detection of corrosion on steel structures using automated image processing, SOETE LABORATORY, Department of electromechanical, systems and metal engineering, faculty of engineering and architecture, Ghent University, 9052, Zwijnaarde, Belgium , Developments in the Built Environment. 2020;3:100022.
11. Reniers Jalba A, Telea A. Robust classification and analysis of anatomic surfaces using 3d skeletons, Eurographics Workshop on Visual Computing for Biomedicine D; 2008.

© 2020 Abikoye et al.; This is an Open Access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Peer-review history:

*The peer review history for this paper can be accessed here:
<http://www.sdiarticle4.com/review-history/64020>*