IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

BLACK AND WHITE IMAGE COLORIZATION USING DEEP LEARNING AND OPEN CV

Shivam Yadav [1], Shashank Yadav[2] ,Dr. Sadhana Rana[3]

[1] B.Tech student, Dept.of CSE, SRMCEM , Lucknow , Uttar Pradesh, India

[2] B.Tech student, Dept.of CSE, SRMCEM , Lucknow , Uttar Pradesh, India

[3] Asst. Professor,, Dept.of CSE, SRMCEM, Lucknow , Uttar Pradesh, India

Shri Ramswaroop Memorial College of Engineering and Management , Lucknow

Abstract : The process of colorizing black and white images has witnessed significant advancements in recent years, spurred by the intersection of computer vision, machine learning, and image processing techniques. This review paper provides a comprehensive overview of the state-of-the-art methodologies and technologies employed in black and white image colorization.

The review systematically categorizes existing methodologies based on their underlying principles, including deep learning-based methods, image segmentation, and rule-based algorithms. A critical analysis of the strengths and limitations of each category is presented, along with an examination of the impact of dataset characteristics on model performance. The paper also explores the role of convolutional neural networks (CNNs), and recurrent neural networks (RNNs) in enhancing colorization accuracy and realism.

The main objective of this paper is to colorize historic images, which are only in black and white form using concepts of convolutional neural networks in prototxt file to construct our desired model. Colorization requires considerable user intervention and remains a boring, sluggish, and extortionate task. So, in this paper we have build a model to colorize the grayscale images automatically by using CNN algorithm in Deep learning where we have used all 4 layers of CNN which are not used in existing papers, which resulted into 13.75% better output than deep hybrid model [1]. This layers will be written in prototxt file. In the existing papers, they only talk about images but we have tried our model on live feed taken from webcam and give the colourful live feed on the otherside.

Furthermore, the review addresses challenges associated with colorizing specific content types, such as faces, landscapes, and historical photographs, and discusses the ethical considerations surrounding the manipulation of visual content. The paper concludes by outlining potential avenues for future research and development in the field, including the integration of domain-specific knowledge and the exploration of novel training strategies.

Index terms: RGB, CNN, Python, Open CV , Deep learning , Python imaging Library(PIL)

1. INTRODUCTION:

Colorizing black and white movies is a very old idea since 1902. For years many movie creators resist the idea of colorizing their black and white films and thought of it as destruction of their art. Today it is accepted as an amplification to their work of art. The technology itself has changed from meticulous hand colorization to today's highly automated techniques. In the USA, Legend movies used its automated technology to color old classics. In 1960, the movie Mughal-E-Azam was famous in India, was colorized in 2004. People from various generations over-crowded the theatres to watch it in color and the movie was a successful for the second time. In this paper, we have trained a Convolutional Neural Network (CNN) to link a black & white image input to a colorful output. In this we are using a pre-trained caffemodel that can be retrieved by the use of opencv in python. Our idea is to use a fully automatic approach which produces decent and realistic colorizations. Deep learning is an existing function of AI that works similarly like a human brain. Example, it processes the data and creates patterns for the use in decision making. The black & white image we have to color can be thought as the L-channel of the image in the LAB color space

- L channel: to encode lightness intensity only,
- a channel: to encodes green-red,
- b channel: to encode blue-yellow

and our objective is to find the a and b components that is luminance and chrominance. Which is done by CNN layers. The Lab image we got can be modified to the RGB color space by the use of standard color space transforms.

- To identify different objects in the image.
- To predict the possible colour distribution for each pixel in the image.
- To calculate the loss and loss function.
- To predict the possible colour distribution for each pixel in the image

2.LITERATURE SURVEY :-

Colorization requires allocating realistic colors to black and white image. Convolutional neural networks are explicitly structured to deal with image data. A lot of work on this idea was done by many authors. Richard Zhang [2] proposed an optimized solution by using enormous dataset and single feed-forward pass in Convolutional Neural Network. Their major motive lies on training part. Human subjects were used to test the output and was able to fool 32% of them and had multiple number of neurons. The many attempts used various architectures. Domonkos Varga [3] suggested the goal of automated coloring of cartoon images, because they are distinct from natural images, they faced problems as their colors rely on animator to animator. The data-set was especially trained for lakhs of cartoon images, 30% of which were used in documentation and rest for training. But drastically, the color unpredictability in cartoons is higher than in natural images and consideration is personalized and moderate. Shweta Salve [4] suggested another similar perspective, utilizing the Google's image classifier, Inception ResNet V2. The system model is divided into 4 parts which are Encoder, Feature extractor, Fusion layer and Decoder. The system is capable of producing satisfactory results, given adequate resources like CPU, Memory, and large data-set. This is generally proof of concept implementation. Yu Chen [5] suggested an approach to mainly solved the problem of coloring Chinese movies from old time. For tuning the overall model, they used existing dataset with their dataset of Chinese images. The network uses multiscale convolution kernels and combines low and middle features which are extracted from VGG-16. V.K. Putri [6] has proposed a method to transform ordinary sketches into colorful images. It makes use of color prediction in CIELab color space and sketch inversion model. This perspective is capable of handling hand-drawn sketches as well as various geometric transformations. The drawbacks found was that, dataset is bounded but it performs well for uncontrolled situations. In few papers, number of neurons is same as the dimension of the feature descriptor extracted from each pixel coordinates in a gray-scale image.

3. PROPOSED WORK

Considering the pixel color is highly reliant on the features of its adjacent pixels, use of CNN is a satisfactory option for image colorization. The condition of having only a grayscale or black and white image, detecting the exact color is complicated. The information is not enough for a network to evaluate the pixel colors. For instance, consider a car image which is in gray form, there are number of acceptable options for car color. To guess a suitable color, we require more information to study the model to match a grayscale input image to the equivalent color of the output image. In the past few years, Convolutional neural network is one of

the most successful learning-based models. CNN verified spectacular capabilities in image processing. In such manner, CNN-based model is proposed by us for automatic image colorization. By using Convolutional Neural Networks, we decided to ambush the issues of image colorization to "hallucinate" what an input black & white image would appear after colorization. For training the network started with the ImageNet dataset and all images were transformed from the RGB color space to the Lab color space. Like the RGB color space, the Lab color space has three channels. But unalike the RGB color space, Lab encodes color information differently.

Since intensity was encoded by L channel, we have used the L channel as our grayscale input to the network. As a result, the network must learn to predict the a channel that is chrominance and b channel which is luminance. By giving L channel as input and the predicted ab channels, we can now form our final output image.

The whole (clarified) process could be synopsized as:

I. • Convert all training images in Lab color space from RGB color space using opencv.

II. • For the input to the network, use L channel and then train the network to predict the ab channels that is the combination of chrominance and luminance. Which is done by the caffemodel that is pretrained with imagenet dataset and prototxt file where we have written the 8 layers using the 4 basic layers of CNN which are convolution layer, pooling, flattening and fully connection.

III. • Merge the input L channel and predicted ab channels with the help of fully connected layer.

IV. • Transform the Lab image back to RGB by using opencv.

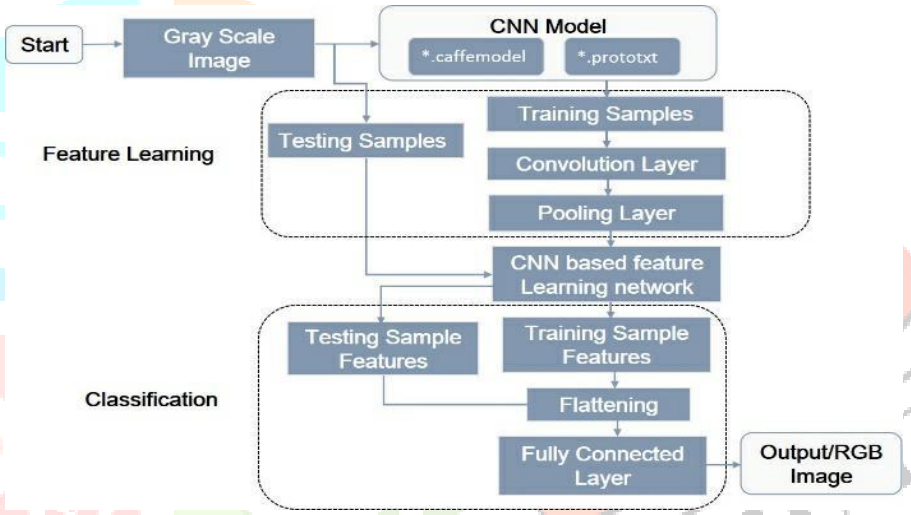

Fig 3.1: System Architecture

3.1. Colorizing black and white images:-

Our colorizer script only requires three imports: NumPy, OpenCV, and argparse [7].Let's go ahead and use argparse to parse command line arguments. This script requires that these four arguments be passed to the script directly from the terminal:

- image: The path to our input black/white image.
- prototxt : Our path to the Caffe prototxt file.
- model : Our path to the Caffe pre-trained model.
- points : The path to a NumPy cluster center points file.

Command line arguments are an elementary skill that you must learn how to use, especially if you are trying to apply more advanced computer vision,

3.2. To build an application with Open CV

An applications purpose you need to do two things:

• Tell to the compiler how the OpenCV library looks.You do this by showing it the header files.

• Tell to the linker from where to get the functions or data structures of OpenCV, when they are needed.

If you use the lib system you must set the path where the library files are and specify in which one of them to look[8]. During the build the linker will look into these libraries and add the definitions and implementation of all used functions and data structures to the executable file.

Open CV's :- functionality that will be used for facial recognition is contained within several modules[9]. Following is a short description of the key namespaces.

CV:- namespace contains image processing and camera calibration methods. The computational geometry functions are also located here[10].

ML:-namespace contains machine-learning interfaces.

High GUI :-namespace contains the basic I/O interfaces and multi-platform windowing capabilities [11][12].OpenCV's functionality that will be used for facial recognition is contained within several modules.

4.EXPERIMENTS

To improve the recognition performance, there are many things that can be improved here, some of them being fairly easy to implement. For example ,you could add colour processing, edge detection, etc[13][14]. You can usually improve the face recognition accuracy by using more input images, at least 50 per person, by taking more photos of each person, particularly from different angles and lighting conditions[15]. If you can't take more photos, there are several simple techniques you could use to obtain more OpenCV has the advantage of being a multi-platform framework; it supports both Windows and Linux, and more recently, Mac OS X[16][17]. Open CV has so many capabilities it can seem overwhelming at first[18]. A good understanding of how these methods work is the key to getting good results when using Open CV. Fortunately, only a select few need to be known beforehand to get started.

4.1. Reading and Visualizing Image using OpenCV

Let's start with very basic stuff. That is reading an image and visualizing it using Open CV[18]. First of all, we import cv2. Then we read the image using Open CV's I'm read() method. This method takes the image path as an argument[19]. Here, the path is Images/audi-640.jpg

The following is the code snippet to change the colour format of the image from BGR to RGB

```
1. image_rgb = cv2.cvtColor(image,cv2.COLOR_BGR2RGB)
2. plt.imshow(image_rgb)
3.plt.title('RGB Image')
4.plt.axis('off')
5.plt.show()
```

In the above code, cv2.cvtColor() takes two arguments. One is the original image reference that we have read using OpenCV's imread() method and the second one is the argument to change the color format. We have usedcv2.COLOR_BGR2RGB.

4.2. IMAGE TRANSLATION USING OPENCV

Translation of an image is moving or relocating an image or object from one location to another [20]. We can relocate the image in any direction using a transformation matrix. The following is a transformation matrix for translation.

$$\begin{bmatrix} 1 & 0 & t_x \end{bmatrix}$$

$$\begin{bmatrix} 0 & 1 & t_y \end{bmatrix}$$

In the above matrix:

- tx : the units we want to move the image to move to the right (along the x-direction).
- ty: the units we want to move the image downwards (along the y-direction).

We can use the warp Af fine() method in Open CV to carry out the image translation operation[21].

4.2.1. TRANSLATION

Translation is the shifting of object's location[22]. If you know the shift in (x,y) direction, let it be (tx,ty), you can create the transformation matrix M as follows:

$$M=\begin{bmatrix}1 & 0 & t_x\\ 0 & 1 & t_y\end{bmatrix}$$

We use the function:
cv.warpAffine (src,dst, M, dsize, flags = cv.INTER_LINEAR,
border Mode = cv.BORDER_CONSTANT,
borderValue = new cv.Scalar())

4.2.2. PARAMETERS

src: input image.
dst: output image that has the size dsize and the same type as src.
Mat: 2 × 3 transformation matrix(cv.CV_64FC1 type).
dsize: size of the output image.
flags: combination of interpolation methods(see cv. Interpolation Flags) and the optional flag WARP_INVERSE_MAP that means that M is the inverse transformation (`dst→src`)

4.3. CODING

But you must keep in mind that, OpenCV follows BGR convention and PIL follows RGB color convention, so to keep the things consistent you may need to do use cv2.cvtColor() before conversion.Pillow and Open CV use different formats of images[23][24]. So you can't just read a image in Pillow and use it manipulate the image in Open CV. Pillow uses the RGB format as @ZdaR highlighted, and Open CV uses the BGR format. So to you need convertor to convert from one format to another .

To convert from OpenCV image to PIL image use:

```
import cv2
import numpy as np
from PIL import Image
opencv_image=cv2.imread("demo2.jpg") # open image using openCV2

# convert from openCV2 to PIL. Notice the
COLOR_BGR2RGB which means that
# the color is converted from BGR to RGB
pil_image=Image.fromarray(cv2.cvtColor(opencv_i
mage, cv2.COLOR_BGR2RGB)
letdst = new cv.Mat();
let M = cv.matFromArray(2, 3, cv.CV_64FC1, [1, 0,50, 0, 1, 100]);
letdsize = new cv.Size(src.rows, src.cols);
// You can try more different parameters
cv.warpAffine(src, dst, M, dsize,
cv.INTER_LINEAR, cv.BORDER_CONSTANT,
new cv.Scalar());
cv.imshow('canvasOutput', dst);
src.delete(); dst.delete(); M.delete();
```

5.RESULTS

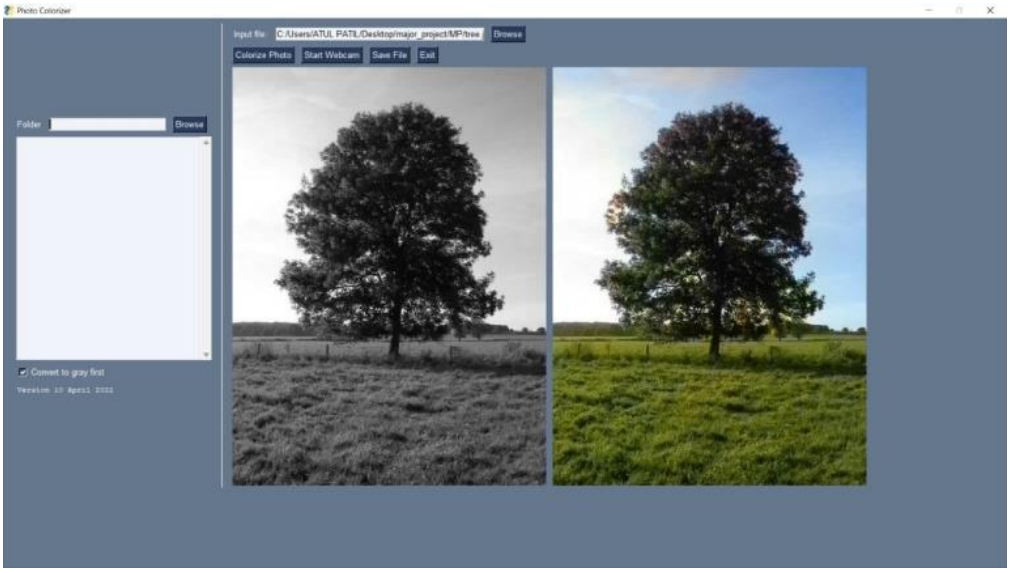

Fig: Results of our model

IN THE ABOVE FIGURE 4, LEFT COLUMN IS THE INPUT GRAYSCALE IMAGE AND THE RIGHT SIDE IS THE OUTPUT OBTAINED FROM OUR MODEL. OUR MODEL GENERATES 84.43% ACCURACY WHICH IS 13.75% MORE ACCURATE AS COMPARED TO DEEP HYBRID MODEL WITH RESPECT TO SOME OF THE PROPERTIES LIKE QUALITY, COLOR AND ACCURACY ETC[4].

Fig: Sample test set input and output from ImageNet.

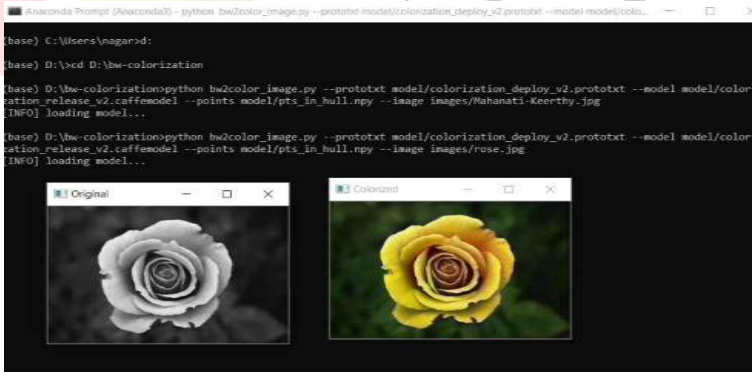

Fig: Sample test set input and output from ImageNet

The regression network outputs are somewhat reasonable. Roads, Rose flower, Medical and nature are different colorization area on road and green of the image, with foliage, and there seems to be a slight amount of color tinting in the sky. We, however, note that the images are severely and generally unattractive. These results are expected given their similarity to Dahl's sample outputs and our hypothesis.

Texture characteristics are used to recognize several objects provided by user's reference image and also narrate a color transfer method. We generated micro scribbles at high confidence areas to accomplish better colorization and the complete colorization is performed using CNN algorithm. The combination of CNN with Opencv gives highly optimized colorization of images. We introduced a method of automatic colorization of unique grayscale images combining CNN techniques in prototxt file. The method is proficient to yield a effective and effervescent colorization of some parts of individual images even when applied to a

ordinary sized dataset that has properties which make it difficult to colorize than natural images, but does not better results when applied to video sequences this is achieved by using the Opencv and color representation.

6. CONCLUSION AND FUTURE WORK

To improve the recognition performance, there are many things that can be improved here, some of them being fairly easy to implement .For example, you could add color processing, edge detection, etc. Through our experiments, we have demonstrated the efficacy and potential of using deep convolutional neural networks to colorize black and white images. In particular, we have empirically shown that formulating the task as a classification problem can yield colorized images that are arguably much more aesthetically-pleasing than those generated by a baseline regression-based model, and thus shows much promise for further development.

7.REFERENCES

[1] K.Kiani, R.Hemmatpour, and R.Rastgoo. "Automatic Grayscale Image Colorization using a Deep Hybrid Model". May 2021.

[2] R.Zhang, P.Isola, and Alexei A. Efros. "Colorful Image Colorization", Berkeley. Oct.5,2016.

[3] C. A. S. Domonkos Varga and T. Szirfffdfffdnyi, Automatic cartoon colorization based on convolutional neural network, https://core.ac.uk/download/pdf/94310076.pdf, 2017.

[4] S. Salve, T. Shah, V. Ranjane, and S. Sadhukhan,Automatization of coloring grayscale images using convolu- tional neural network, Apr. 2018. DOI: 10.1109/ICICCT. 2018.8473259.

[5] Automatic colorization of images from Chinese black and white films based on cnn, 2018. DOI: 10.1109/ICALIP. 2018.8455654

[6] V. K. Putri and M. I. Fanany, "Sketch plus colorization deep convolutional neural networks for photos generation from sketches," in 2017 4th International Conference on Electrical Engineering, Computer Science and Informat- ics (EECSI), Sep. 2017, pp. 1–6. DOI: 10.1109/EECSI. 2017.8239116.

[7]. K.Santhi Sri, P.R.S.M.Lakshmi, and M.V.Bhujanga Ra0,(2017), "A Study of Security and Privacy Attacks in Cloud Computing Environment", IJASTEMS,Vol.3,No.1,pp. 235-238.

[8]R S M Lakshmi Patibandla and N. Veeranjaneyulu, (2018), "Explanatory & Complex Analysis of Structured Data to Enrich Data in Analytical Appliance", International Journal for Modern Trends in Science and Technology, Vol. 04, Special Issue 01, pp. 147-151.

[9]. R S M Lakshmi Patibandla, Santhi Sri Kurra, Ande Prasad and N.Veeranjaneyulu, (2015), "Unstructured Data: Qualitative Analysis", J. of Computation In Biosciences And Engineering, Vol. 2,No.3,pp.1-4.

[10]. R S M Lakshmi Patibandla, Santhi Sri Kurra and H.-J. Kim,(2014), "Electronic resource management using cloud computing for libraries", International Journal of Applied Engineering Research, Vol.9,pp. 18141-18147.

[11]. Ms.R.S.M.Lakshmi Patibandla Dr.Ande Prasad and Mr.Y.R.P.Shankar,(2013), "SECURE ZONE IN CLOUD",International Journal of Advances in Computer Networks and its Security,Vol.3,No.2,pp.153-157.

[12]. Patibandla, R. S. M. Lakshmi et al., (2016), "Significance of Embedded Systems to IoT.", International Journal of Computer
Science and Business Informatics, Vol.16,No.2,pp.15-23.

[13]. AnveshiniDumala and S. PallamSetty. (2019), "Investigating the Impact of Network Size on LANMAR Routing Protocol in a Multi-Hop Ad hoc Network", i-manager's Journal on Wireless Communication Networks (JWCN), Volume 7, No. 4, pp.19- 26.

[14]. AnveshiniDumala and S. PallamSetty. (2019), "Performance analysis of LANMAR routing protocol in SANET and MANET",
International Journal of Computer Science and Engineering (IJCSE) – Vol. 7,No. 5, pp.1237-1242.

[15]. AnveshiniDumala and S. PallamSetty. (2018), "A Comparative Study of Various Mobility Speeds of Nodes on the Performance of LANMAR in Mobile Ad hoc Network", International Journal of Computer Science and Engineering (IJCSE) – Vol. 6, No. 9, pp. 192-198.

[16]. AnveshiniDumala and S. PallamSetty. (2018), "Investigating the Impact of IEEE 802.11 Power Saving Mode on the Performance of LANMAR Routing Protocol in MANETs", International Journal of

Scientific Research in Computer Science and Management Studies (IJSRCSMS) – Vol.7, No. 4.
[17]. AnveshiniDumala and S. PallamSetty. (2016),"Analyzing the steady state behavior of RIP and OSPF routing protocols in the context of link failure and link recovery in Wide Area Network", International Journal of Computer Science Organization Trends (IJCOT) – Vol. 34 No 2, pp.19-22.
[18]. AnveshiniDumala and S. PallamSetty. (2016),"Investigating the Impact of Simulation Time on Convergence Activity & Duration of EIGRP, OSPF Routing Protocols under Link Failure and Link Recovery in WAN Using OPNET Modeler", International Journal of Computer Science Trends and Technology (IJCST) – Vol. 4 No. 5, pp. 38-42.
[19]. VellalacheruvuPavani and I. Ramesh Babu (2019) ,"Three Level Cloud Storage Scheme for Providing Privacy Preserving using Edge Computing",International Journal of Advanced Science and Technology Vol. 28, No. 16, pp. 1929 – 1940.
[20]. VellalacheruvuPavani and I. Ramesh Babu," A Novel Method to Optimize the Computation Overhead in Cloud Computing by Using Linear Programming" International Journal of Research and Analytical Reviews May 2019, Volume 6, Issue 2,PP.820-830..
[21]. Anusha Papasani and Nagaraju Devarakonda,(2016),"Improvement of Aomdv Routing Protocol in Manet and Performance Analysis of Security Attacks", International Journal Of Research in Computer Science & Engineering ,Vol.6,No.5, pp.4674-4685.
[22]. Sk.Reshmi Khadherbhi,K.Suresh Babu , Big Data Search Space Reduction Based On User Perspective Using Map Reduce ,International Journal of Advanced Technology and Innovative Research Volume.07, IssueNo.18, December-2015, Pages: 3642-3647
[23]. B.V.Suresh kumar,Sk.Reshmi Khadherbhi ,BIG-IOT Framework Applications and Challenges: A Survey Volume 7, Issue VII, JULY/2018 pg.no 1257-1264
[24]. P.Sandhya Krishna,Sk.Reshmi Khadherbhi,V.Pavani, Unsupervised or Supervised Feature Finding For Study of Products Sentiment ,International Journal of Advanced Science and Technology, Vol 28 No 16 (2019).