Journal of Engineering Sciences
Vol 11, Issue 4 , April/2020
ISSN NO:0377-9254

IMAGE COLORIZATION WITH DEEP CONVOLUTIONAL OPEN CV

K. Yamini[1], K. Sai Swetha[2] , P. Lakshmi Prasanna[3], M. Rupa Venkata Swathi[4] , Venkata Rao Maddumala[5]

[1,2,3,4]IV B.Tech, Department of Information Technology, Vignan's Nirula Institute of Technology & Science for Women, Peda Palakaluru, Guntur-522009, Andhra Pradesh, India.
[5]Asst.Professor, Department of Information Technology, Vignan's Nirula Institute of Technology & Science for Women, Peda Palakaluru, Guntur-522009, Andhra Pradesh, India.
venkatmaddumala@gmail.com

ABSTRACT

Image re-colorization is a challenging topic of ongoing research in Computer Vision. We present a convolutional neural network based system using OpenCV that faithfully colorizes black and white photographic images without direct human assistance. We explore various color spaces & image weights. We take a grayscale image as input and attempt to produce a coloring scheme. The goal is to make the output image as realistic as the input and converting the gray scale image to colorized image by OpenCV mechanisms. We give one image as input and system will automatically convert the black and white image into color image.

Keywords: RGB, CNN, Python, Open CV.

1. INTRODUCTION

Automated colorization of black and white images has been subject to much research within the computer vision and machine learning communities[1]. Beyond simply beingfascinating from an aesthetics and artificial intelligence perspective, such capability has broad practical application ranging from video restoration to image enhancement forimproved interpretability[2].

In recent years, convolutional neural network (CNN) have emerged as the de facto standard for solving image classification problems, achieving error rates lower than 4% in the ImageNet challenge[3][4]. CNNs owe much of their success to their ability to learn and discern colors, patterns, and shapes within images and associate them with object classes.[5][6] We believe that these characteristics naturally lend themselves well to colorizing images since object classes, patterns, and shapes generally correlate with color choice[7][41].

Image colorization is the process of taking an input grayscale (black and white) image and then producing an output colorized image that represents the semantic colors and tones of the input (for example, an ocean on a clear sunny day must be plausibly[42][43] "blue" — it can't be colored "hot pink" by the model)[8].The technology itself has moved from painstaking hand colorization to today's largely automated techniques. In the United States, Legend Films used its automated technology to color old classics[9][44]. In India, the movie Mughal-e-Azam, a blockbuster released in 1960 was remastered in color in 2004[10][11]. People from multiple generations crowded the theatres to see it in color and the movie was a huge hit for the second time!Many colorization papers have been published using traditional computer vision methods. One of my favorites is a paper titled Colorization using Optimization by Anat Levin, DaniLischinski, and Yair Weiss[12]. It used a few colored scribbles to guide an optimization problem for solving colorization[45][46].

Let's first define the colorization problem in terms of the CIE Lab color space. Like the RGB color space, it is a 3-channel color space, but unlike the RGB color space, color information is encoded only in the a(green-red component) and b (blue-yellow component) channels[13][14][47]. The L (lightness) channel encodes intensity information only[15][48].

The grayscale image we want to color can be thought as the L-channel of the image in the Lab color space and our objective to find the a and b components[49][50]. The Lab image so obtained can be transformed to the RGB color space using standard color space transforms. For example, in OpenCV[16], this can be achieved using cvtColor with COLOR_BGR2Lab option.

2. METHODS

In this article, we will take a practical approach to some basic image manipulation and transformations using OpenCV and Python. OpenCV is one of the best computer vision libraries and it is easy to use as well[17]. If you want to become a master in the field of computer vision and deep learning, then OpenCV is one library that you must check out[18][51][52]. So, in this blog post, we will check out some basic image manipulation and transformation techniques[53][54]. This will help you get started really easily whenever you find a new image dataset for your own project.

2.1.Install OpenCV

To follow along with this article, you will need the OpenCV library. If you have it already, then it's well and good. Else, you can easily install it using pip command[19].

pip install opencv-python

That's it. You are all set now to follow along. Just one more thing. We will carry out all the image manipulations on one image only[20]. This will keep things really simple to follow. So, you can either choose one image of your choice or download the following image.

2.2. Reading and Visualizing Image using Open CV

Let's start with very basic stuff. That is reading an image and visualizing it using OpenCV.

The following is the code for reading and showing an image using Open CVOur journal was inspired in part by Ryan Dahl's CNN based system for automatically colorizing images [21]. Dahl's system relies on several ImageNet-trained layers from VGG16 [22][23], integrating them with an autoencoderlike system with residual connections that merge intermediate outputs produced by the encoding portion of the network comprising the VGG16 layers with those produced by the latter decoding portion of the network[55][56]. Image colorization is the process of taking an input grayscale (black and white) image and then producing an output colorized image that represents the semantic colors and tones of the input (for example, an ocean on a clear sunny day must be plausibly "blue" — it can't be colored "hot pink" by the model)[24].

Previous methods for image colorization either:

1. Relied on significant human interaction and annotation
2. Produced restarted colorization

The novel approach we are going to use here today instead relies on deep learning. We will utilize a Convolutional Neural Network capable of colorizing black and white images with results that can even "fool" humans!

Similar to the RGB color space, the Lab color space has three channels*[25]*. But unlike the RGB color space, Lab encodes color information differently:

- The *L* channel encodes lightness intensity only
- The *a* channel encodes green-red
- And the *b* channel encodes blue-yellow

Since the *L* channel encodes only the intensity, we can use the *L* channel as our grayscale input to the network.

From there the network must learn to predict the *a* and *b* channels[26]. Given the input *L* channel and the predicted *ab* channels we can then form our final output image.

The entire (simplified) process can be summarized as:

- Convert all training images from the RGB color space to the Lab color space.
- Use the *L* channel as the input to the network and train the network to predict the *ab* channels.

- Combine the input *L* channel with the predicted *ab* channels.
- Convert the Lab image back to RGB.

3. PROPOSED METHOD

The proposed technique colorizes gray scale images by transferring colors from a gray scale image to a destination reference color image[27]. A feed forward convolutional neural network (CNN) is constructed and trained by mapping the pixels from a gray scale space(gray scale version) into a color space of the selected reference.

In proposed system we change the black and white or gray scale image into a recolorized image[28][29]. Here we also transform the color spaces of the image from BRG color space to RGB color space. We use scales and pixel values to map the input image with the models we used to change the color.The proposed algorithm has two main phases of action. Phase one is setting up the (CNN) and training it using machine learning models.

3.1. Colorizing black and white images with OpenCV

Our colorizer script only requires three imports: NumPy, OpenCV, and argparse [30].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, image processing, or deep learning concepts.

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[31]. 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[32]. 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[33].

MLnamespace contains machine-learning interfaces. HighGUI namespace contains the basic I/O interfaces and multi-platform windowing capabilities [34][35].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 color processing, edge detection, etc[36][37]. You can usually improve the facerecognition 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[38]. 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[39][40].

Open CV has so many capabilities it can seem overwhelming at first[41]. 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.

Journal of Engineering Sciences
Vol 11, Issue 4 , April/2020
ISSN NO:0377-9254

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[42]. First of all, we import cv2. Then we read the image using Open CV's im read() method. This method takes the image path as an argument[43]. Here, the path is Images/audi-640.jpg

The following is the code snippet to change the color 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. There are many other options as well. To know about other color space conversions in OpenCV, you can visit this link

4.2. IMAGE TRANSLATION USING OPENCV

Translation of an image is moving or relocating an image or object from one location to another [44]. We can relocate the image in any direction using a transformation matrix. The following is a transformation matrix for translation.

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

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[45]. Let's write the code first.

4.2.1. TRANSLATION

Translation is the shifting of object's location[46]. 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, borderMode = 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[47][48]. 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 a convertor to convert from one format to anotherTo convert from PIL image to OpenCV use:

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

Journal of Engineering Sciences
Vol 11, Issue 4 , April/2020
ISSN NO:0377-9254

```
# 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_image, 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();
```

RESULTS

4.4.1.REQUIRED PACKAGES

- CMake 2.8.8 or higher
- Xcode 4.2 or higher

Now to read the image, use the imread() method of the cv2 module, specify the path to the image in the arguments and store the image in a variable as below:

```
img = cv2.imread("pyimg.jpg")
```

The image is now treated as a matrix with rows and columns values stored in img.

```
>>>print(type(img))
<class 'numpy.ndarray'>
```

If you only want to apply contrast in one image, you can add a second image source as zeros using NumPy.

4.4.2. EXECUTION

```
python bw2color_image.py --prototxt model/colorization_deploy_v2.prototxt --model model/colorization_release_v2.caffemodel --points model/pts_in_hull.npy --image images/Mahanati-Keerthy.jpg
```

```
python bw2color_video.py --prototxt model/colorization_deploy_v2.prototxt --model model/colorization_release_v2.caffemodel --points model/pts_in_hull.npy
```

Figure1. Sample test set input and output from ImageNet.

Vol 11, Issue 4 , April/2020
ISSN NO:0377-9254

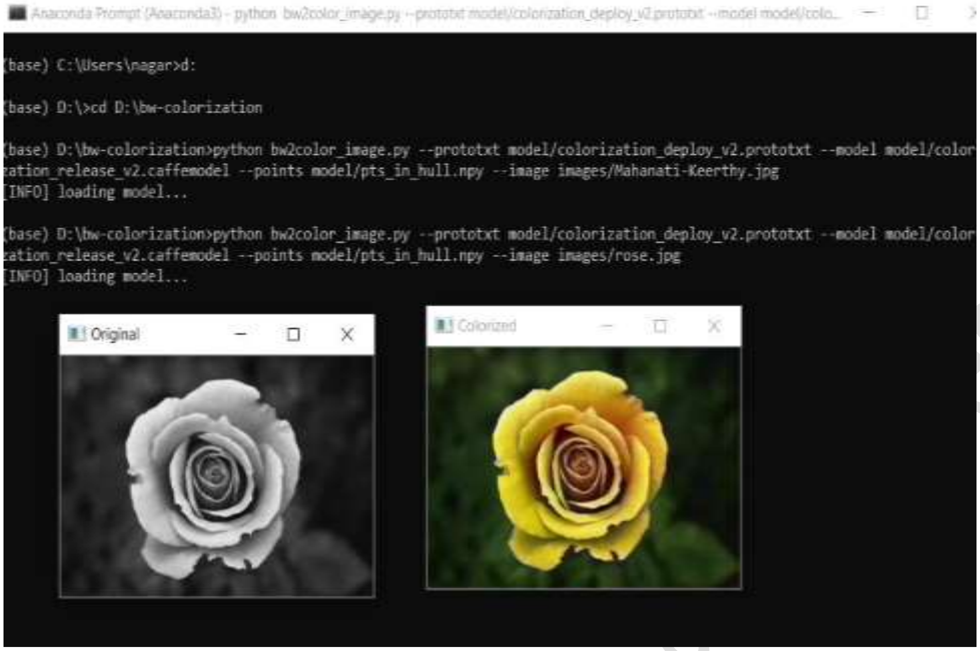

Figure 2. Sample test set input and output from ImageNet.

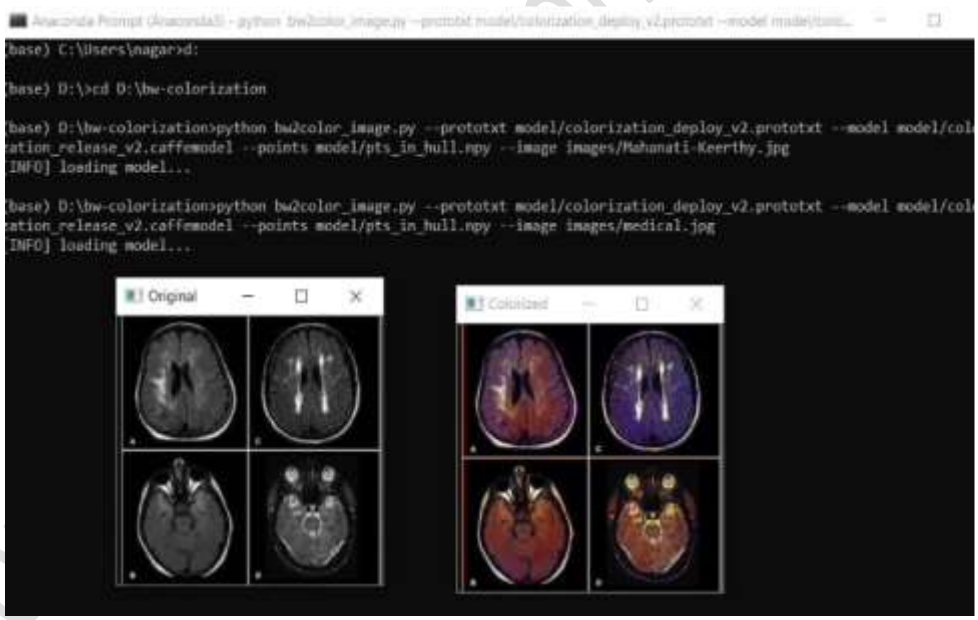

Figure 3. Test set input images (left column), regression network output (center column), and classification network output (rightcolumn).

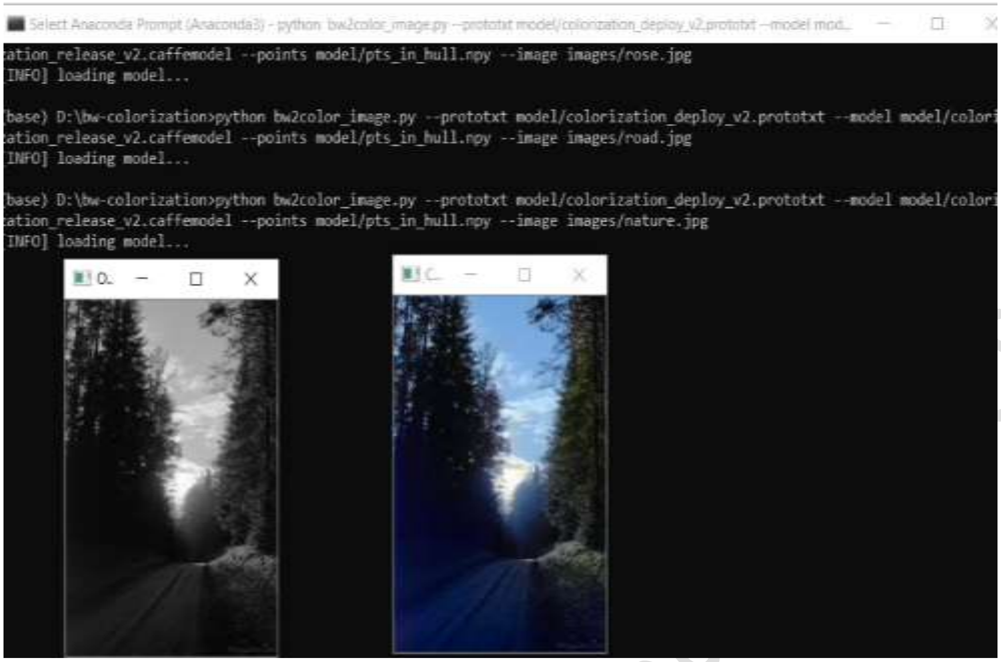

Figure 4. Sample outputs exhibiting color inconsistency issue.

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.

5. 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.

REFERENCES

[1]. Open CV Homepage, http://opencv.willowgarage.com

[2]. Wikipedia, Three-dimensional face recognition, http://en.wikipedia.org/wiki/Three-dimensional_face_recognition

[3]. http://en.wikipedia.org/wiki/Active_appearance_model

[4]. Computer Vision Papers, http://www.cvpapers.com/

[5]. https://stackoverflow.com/questions/43232813/convert-opencv-image-format-to-pil-image-format

[6]. Wikipedia, Histogram equalization, http://en.wikipedia.org/wiki/Histogram_equalization

[7]. http://tinyclouds.org/colorize/, 2016.

[8]. X. Glorot and Y. Bengio. Understanding the difficulty oftraining deep feedforward neural networks. In Internationalconference on artificial intelligence and statistics, pages249–256, 2010.

[9]. Lakshman Narayana Vejendla and A Peda Gopi, (2019)," Avoiding Interoperability and Delay in Healthcare Monitoring System

Using Block Chain Technology", Revue d'Intelligence Artificielle , Vol. 33, No. 1, 2019,pp.45-48.

[10]. Gopi, A.P., Jyothi, R.N.S., Narayana, V.L. et al. (2020), "Classification of tweets data based on polarity using improved RBF kernel of SVM" . Int. j. inf. tecnol. (2020). https://doi.org/10.1007/s41870-019-00409-4.

[11]. A Peda Gopi and Lakshman Narayana Vejendla, (2019)," Certified Node Frequency in Social Network Using Parallel Diffusion Methods", Ingénierie des Systèmes d' Information, Vol. 24, No. 1, 2019,pp.113-117.. DOI: 10.18280/isi.240117

[12]. Lakshman Narayana Vejendla and Bharathi C R ,(2018),"Multi-mode Routing Algorithm with Cryptographic Techniques and Reduction of Packet Drop using 2ACK scheme in MANETs", Smart Intelligent Computing and Applications, Vo1.1, pp.649-658. DOI: 10.1007/978-981-13-1921-1_63 DOI: 10.1007/978-981-13-1921-1_63

[13]. Lakshman Narayana Vejendla and Bharathi C R, (2018), "Effective multi-mode routing mechanism with master-slave technique and reduction of packet droppings using 2-ACK scheme in MANETS", Modelling, Measurement and Control A, Vol.91, Issue.2, pp.73-76. DOI: 10.18280/mmc_a.910207

[14]. Lakshman Narayana Vejendla , A Peda Gopi and N.Ashok Kumar,(2018)," Different techniques for hiding the text information using text steganography techniques: A survey", Ingénierie des Systèmes d'Information, Vol.23, Issue.6,pp.115-125.DOI: 10.3166/ISI.23.6.115-125

[15]. A Peda Gopi and Lakshman Narayana Vejendla (2018), "Dynamic load balancing for client server assignment in distributed system using genetic algorithm", Ingénierie des Systèmes d'Information, Vol.23, Issue.6, pp. 87-98. DOI: 10.3166/ISI.23.6.87-98

[16]. Lakshman Narayana Vejendla and Bharathi C R,(2017),"Using customized Active Resource Routing and Tenable Association using Licentious Method Algorithm for secured mobile ad hoc network Management", Advances in Modeling and Analysis B, Vol.60, Issue.1, pp.270-282. DOI: 10.18280/ama_b.600117

[17]. Lakshman Narayana Vejendla and Bharathi C R,(2017),"Identity Based Cryptography for Mobile ad hoc Networks", Journal of Theoretical and Applied Information Technology, Vol.95, Issue.5, pp.1173-1181. EID: 2-s2.0-85015373447

[18]. Lakshman Narayana Vejendla and A Peda Gopi, (2017)," Visual cryptography for gray scale images with enhanced security mechanisms", Traitement du Signal,Vol.35, No.3-4,pp.197-208. DOI: 10.3166/ts.34.197-208

[19]. A Peda Gopi and Lakshman Narayana Vejendla, (2017)," Protected strength approach for image steganography", Traitement du Signal, Vol.35, No.3-4,pp.175-181. DOI: 10.3166/TS.34.175-181

[20]. Lakshman Narayana Vejendla and A Peda Gopi, (2020)," Design and Analysis of CMOS LNA with Extended Bandwidth For RF Applications", Journal of Xi'an University of Architecture & Technology, Vol. 12, Issue. 3,pp.3759-3765. https://doi.org/10.37896/JXAT12.03/319.

[21]. Chaitanya, K., and S. Venkateswarlu,(2016),"DETECTION OF BLACKHOLE & GREYHOLE ATTACKS IN MANETs BASED ON ACKNOWLEDGEMENT BASED APPROACH." Journal of Theoretical and Applied Information Technology 89.1: 228.

[22]. Patibandla R.S.M.L., Kurra S.S., Mundukur N.B. (2012), "A Study on Scalability of Services and Privacy Issues in Cloud Computing". In: Ramanujam R., Ramaswamy S. (eds) Distributed Computing and Internet Technology. ICDCIT 2012. Lecture Notes in Computer Science, vol 7154. Springer, Berlin, Heidelberg

[23]. Patibandla R.S.M.L., Veeranjaneyulu N. (2018), "Survey on Clustering Algorithms for Unstructured Data". In: Bhateja V., Coello Coello C., Satapathy S., Pattnaik P. (eds) Intelligent Engineering Informatics. Advances in Intelligent Systems and Computing, vol 695. Springer, Singapore

[24]. Patibandla, R.S.M.L., Veeranjaneyulu, N. (2018), "Performance Analysis of Partition and Evolutionary Clustering Methods on Various Cluster Validation Criteria", Arab J Sci Eng ,Vol.43, pp.4379–4390.

[25]. R S M Lakshmi Patibandla, Santhi Sri Kurra and N.Veeranjaneyulu, (2015), "A Study on Real-Time Business Intelligence and Big Data",Information Engineering, Vol.4,pp.1-6.

[26]. K. Santhisri and P.R.S.M. Lakshmi,(2015), " Comparative Study on Various Security Algorithms in Cloud Computing", Recent Trends in Programming Languages ,Vol.2,No.1,pp.1-6.

[27]. K.Santhi Sri and PRSM Lakshmi,(2017), "DDoS Attacks, Detection Parameters and Mitigation in Cloud Environment", IJMTST,Vol.3,No.1,pp.79-82.

[28]. P.R.S.M.Lakshmi,K.Santhi Sri and Dr.N. Veeranjaneyulu,(2017), "A Study on Deployment of Web Applications Require Strong Consistency using Multiple Clouds", IJMTST,Vol.3,No.1,pp.14-17.

[29]. P.R.S.M.Lakshmi,K.Santhi Sri and M.V.Bhujanga Ra0,(2017), "Workload Management through Load Balancing Algorithm in Scalable Cloud", IJASTEMS,Vol.3,No.1,pp.239-242.

[30]. 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.

[31]. 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.

[32]. 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.

[33]. 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.

[34]. 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.

[35]. 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.

[36]. AnveshiniDumala and S. PallamSetty. (2020),"LANMAR routing protocol to support real-time communications in MANETs using Soft computing technique", 3rd International Conference on Data Engineering and Communication Technology (ICDECT-2019), Springer, Vol. 1079, pp. 231-243.

[37]. 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.

[38]. 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.

[39]. 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.

[40]. 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.

[41]. 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.

[42]. 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.

[43]. 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.

[44]. 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..

[45]. 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.

[46]. 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

[47]. 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

[48]. 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).

[49]. K.Santhi Sri, Dr.Ande Prasad (2013), "A Review of Cloud Computing and Security Issues at Different Levels in Cloud Computing" , International Journal on Advanced Computer Theory and Engineering Vol. 2,pp 67-73.

[50]. K.Santhi Sri, N.Veeranjaneyulu(2018), "A Novel Key Management Using Elliptic and Diffie-Hellman for Managing users in Cloud Environment", Advances in Modelling and Analysis B,Vol.61,No.2,pp 106-112.

[51]. K.Santhi Sri, N.Veeranjaneyulu(2019), "Decentralized Key Management Using Alternating Multilinear Forms for Cloud Data Sharing with Dynamic Multiprivileged Groups", Mathematical Modelling of Engineering Problems,Vol.6,No.4,pp511-518.

[52]. S.Sasikala, P.Sudhakar, "interpolation of CFA color Images with Hybrid image denoising", 2014 Sixth International Conference on Computational Intelligence and Communication Networks, DOI 10.1109/.53 193 DOI 10.1109/CICN.2014.53, pp. 193-197.

[53]. Me. Jakeera Begum and M.Venkata Rao, (2015), "Collaborative Tagging Using CAPTCHA" International Journal of Innovative Technology And Research, Volume No.3, Issue No.5,pp,2436 – 2439.

[54]. L.Jagajeevan Rao, M. Venkata Rao, T.Vijaya Saradhi (2016), "How The Smartcard Makes the Certification Verification Easy" Journal of Theoretical

and Applied Information Technology, Vol.83. No.2, pp. 180-186.

[55]. Venkata Rao Maddumala, R. Arunkumar, and S. Arivalagan (2018)"An Empirical Review on Data Feature Selection and Big Data Clustering" Asian Journal of Computer Science and Technology Vol.7 No.S1, pp. 96-100.

[56]. Singamaneni Kranthi Kumar, Pallela Dileep Kumar Reddy, Gajula Ramesh, Venkata Rao Maddumala, (2019), "Image Transformation Technique Using Steganography Methods Using LWT Technique" ,Traitement du Signalvol 36, No 3, pp. 233-237.