International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878 (Online), Volume-9 Issue-2, July 2020

Empirical Examination of Color Spaces in Deep Convolution Networks

Urvi Oza, Pankaj kumar

Abstract: ***In this paper we present an empirical examination of deep convolution neural network (DCNN) performance in different color spaces for the classical problem of image recognition/classification. Most such deep learning architectures or networks are applied on RGB color space image data set, so our objective is to study DCNNs performance in other color spaces. We describe the design of our novel experiment and present results on whether deep learning networks for image recognition task is invariant to color spaces or not. In this study, we have analyzed the performance of 3 popular DCNNs (VGGNet, ResNet, GoogleNet) by providing input images in 5 different color spaces(RGB, normalized RGB, YCbCr, HSV , CIE-Lab) and compared performance in terms of test accuracy, test loss, and validation loss. All these combination of networks and color spaces are investigated on two datasets- CIFAR 10 and LINNAEUS 5. Our experimental results show that CNNs are variant to color spaces as different color spaces have different performance results for image classification task.***

Keywords: CIFAR 10, Color spaces, Convolution neural networks, LINNAEUS 5, Object recognition.

I. INTRODUCTION

Object Recognition or image classification is a process of identifying objects in given image or video sequence. It is one of the key components in many advanced computer vision technologies such as Activity recognition, Smart parking systems, Robotic vision, Surveillance, Autonomous car and many more. Object Recognition requires machine to identify objects in image irrespective of its alignment, scale, position, lightning or any other environmental changes. Over the years there were many techniques proposed for object recognition task such as Shape based recognition [1], Local features based recognition[2], Matching feature descriptors (SIFT)[3], Template matching [4], etc. General approach of such object recognition method is to create database of images and extract features from images such that it can describe the object classes well. As images of same class of object have similarity in their description, these images can be separated based on its features and this learned features information can be used to classify objects in new images. Color histograms[5], Moment invariants[6], MSER(Maximally Stable Extremal Regions) features[7], SIFT feature descriptor[3], Cluster signatures[8], Image gradients all are used as methods to compute good description of images which summarize objects well and have similarity between images of same class. An image has multiple features based on color, shape, texture, etc. Color features of images are often useful in various applications such as category based image retrieval, classification, etc.

Deep learning architectures like “Convolutional Neural Network (CNN)” have achieved less error rate compared to other traditional methods in many computer vision tasks and competitions such as object recognition, object detection, segmentation, etc. In past few years CNNs are highly used in computer vision applications due to availability of high-performance computing resources, better accuracy and speed and inventions of better architectures and methods day by day. In CNN, each layer helps the network to learn the inherent features of each class of object so that network can identify class of objects based on learnt features of image.

II. MOTIVATION

DCNNs have millions of parameters which are trained during model training process. It is quite difficult to inspect or understand that which image features are regarded highly by the network to perform given task. It is possible that some algorithms or models perform better in certain color spaces than others as color is also considered important feature of image. In fact, similar research and analysis work have been done of how color spaces influences various algorithms/methods of applications such as segmentation, object detection, recognition etc. Analytical study of color spaces for plant pixel detection[9] suggested HSV as best color space to classify plant pixels. Whereas analysis of several color spaces for skin segmentation task[10] shows that it is unaffected by choice of color space. However, more recent studies show that human skin pixel segmentation in YCbCr color space is better than HSV color space[11]. Study of Foreground and shadow detection for traffic monitoring application[12] suggested that YCbCr is best color space for such applications. For traffic signal classification application deep neural network is being used[13] where input images are converted from RGB to Lab color space as preprocessing step to achieve better results compared to RGB color space.

Manuscript received on May 25, 2020.
Revised Manuscript received on June 29, 2020.
Manuscript published on July 30, 2020.
***** Correspondence Author
Urvi Oza*, Dhirubhai Ambani Institute of Information and Communication Technology, Gandhinagar, India. E-mail: 201921009@daiict.ac.in
Prof. Pankaj Kumar, Dhirubhai Ambani Institute of Information and Communication Technology, Gandhinagar, India. E-mail: pankaj_k@daiict.ac.in

In comparative study on different color spaces of input images for image segmentation using automatic GrabCut technique[14] authors have showed that the YUV color space generates best segmentation accuracy on their image dataset. Idea of experimenting with different color spaces in DCNNs is worth doing as [15] proposes a network called ColorNet which takes input images in different color spaces in order to improve accuracy of image classification.

In this experiment we are examining performance of CNNs like VGGNet, ResNet and GoogleNet in different color spaces such as RGB, normalized rgb, YCbCr, HSV and CIE-Lab using transfer learning approach. In recent time multiple object detection and classification based applications uses transfer learning approach to train their network on different tasks such as monkey detection[25] and diabetic retinopathy classification [26] as it takes less training time.

Rest of the papers is described as follows - In chapter II we have briefly introduced all the color spaces used, their relationship with RGB color space and reason for choosing them in our study. In Chapter III architecture of CNNs which we are examining is given. Information regarding data set and methodology is described in chapter IV. Analysis and comparison of results are given in chapter V and conclusion is given in chapter VI.

III. CHOICE OF COLOR SPACES

A color model is an abstract mathematical model of describing a color, whereas color space is specific color organization which can be displayed or reproduced in a medium. Images are represented as multidimensional matrix where each cell represents intensity values at that particular pixel location.

There are multiple color spaces available but for the experiment purpose we have considered few of them such that it can summarize major available color spaces. We have considered major color models which are CIE, RGB, YUV and Hue based color model. Following are color spaces based on these models [16] -

A. RGB color model

RGB color model stores red, green and blue light intensity values separately. Multiple color spaces are derived on RGB color model and used for different purposes such as - Standard RGB, Adobe RGB, CIE RGB etc. We have considered the standard RGB color space as it is the most commonly used in web-based application and to render images on computer screens. We have also considered normalized RGB color space (rgb) as it is used in robotics vision and object recognition tasks.

1) RGB - Most of the images are stored in standard RGB color space where each pixel intensity value is stored in terms of red, green and blue values.

2) rgb - It is normalized RGB color space which preserves the color values and remove illumination dependence such as shadows and lighting information. Because of this characteristic of normalized RGB color space, it is used in object detection and recognition task to detect objects irrespective of illumination condition.

$$r = \frac{R}{R+G+B}$$
$$g = \frac{G}{R+G+B} \quad (1)$$
$$b = \frac{B}{R+G+B}$$

B. Luma and Chroma model

Color space based on Luma and Chroma model stores color information and illumination information separately. YUV and YIQ are color spaces belongs to Luma and chroma model which are used in television broadcasting systems whereas YPbPr is used in analog videos for transmission. YCbCr is digital form of YPbPr, which is used in all digital systems as an encoding scheme. We have chosen YCbCr for our study as it is the most commonly used color space on luma chroma model in digital video and image systems and it is widely used in compression schemes such as MPEG, JPEG for digital encoding of color images.

1) YCbCr - Here Y represents luma value and Cb is Y(luma) – b(blue color value) and Cr is Y(luma) – r(red color value). Human eyes are more sensitive to brightness value than color value and YCbCr uses this fact while representing images. Conversion of RGB to YCbCr is shown in (2).

$$Y = 0.299R + 0.587G + 0.114B$$
$$Cb = 128 - 0.168736R - 0.331264G + 0.5B \quad (2)$$
$$Cr = 128 + 0.5R - 0.418688G - 0.081312B$$

C. Hue based color model

HSV, HSL, HSB all are color models derived by applying cylindrical transformation to RGB color model. These color models are more natural to humans as it is based on how humans perceive color. For HSV (also known as HSB) H stands for hue which is color or chrominance value, S stands for saturation which indicates gray value in color and V stands for brightness value. In HSV color space V represents brightness of color or power of source which can be of any color whereas in HSL color space L represents brightness value in terms of white. Same as YCbCr color space, Hue based color spaces stores luminance and chrominance values separately.

1) HSV - Transformation from RGB to HSV can be done easily. We have chosen HSV color space for our study as it has been reported that input in HSV color space works better than RGB color space in some of the applications such as face recognition. RGB to HSV is cylindrical transformation as shown in (3).

$$C\max = \max(R,G,B)$$
$$C\min = \min(R,G,B)$$
$$C = C\max - C\min$$
$$H = \begin{cases} 0°, C = 0 \\ 60° \times \dfrac{G-B}{C} \bmod 6, C = R \\ 60° \times \dfrac{B-R}{C} + 2, C = G \\ 60° \times \dfrac{R-G}{C} + 4, C = B \end{cases} \quad (3)$$
$$S = \begin{cases} 0, C\max = 0 \\ \dfrac{C}{C\max}, C\max \neq 0 \end{cases}$$
$$V = C\max$$

D. CIE color model

The CIE color model is created by the International Commission on Illumination (CIE) to map all the colors which can be perceived by human eye. There were multiple color spaces introduced which belong to CIE model such as CIE-XYZ, CIE – LUV, CIE-Lab, etc. CIE-XYZ was first color space of CIE model which has used tristimulus (a combination of 3 color values that are close to red/green/blue values) to produce any color. CIE-LUV is modification of CIE-XYZ model. CIE-Lab can be derived from CIE-XYZ color space but it is more perpetually uniform than CIE-XYZ color space.

We have chosen CIE-LAB as it includes all perceivable colors and its gamut exceeds those of other color spaces like RGB.

1) CIE – Lab - In CIE-Lab color space L is lighting component which closely matches with human perception of light, a and b components are (green \(-\) red) and (blue \(-\) yellow) color values respectively. CLE-LAB is device independent model as colors are defined independent of device they are displaying on. It is uniform with human perception, which means that change in any values L,a or b brings the same amount of change in perception value.

Conversion of RGB to CIE-LAB requires RGB to CIE-XYZ conversion followed by CIE-XYZ to CIE-LAB conversion as shown in (4).

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.4124564 & 0.3575761 & 0.1804375 \\ 0.2126729 & 0.7151522 & 0.0721750 \\ 0.0193339 & 0.1191920 & 0.9503041 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$
$$L = 116 * f\left(\frac{Y}{Yn}\right) - 16$$
$$a = 500 * \left(f\left(\frac{X}{Xn}\right) - f\left(\frac{Y}{Yn}\right) \right) \quad (4)$$
$$b = 200 * \left(f\left(\frac{Y}{Yn}\right) - f\left(\frac{Z}{Zn}\right) \right)$$
$$f(t) = \begin{cases} \sqrt[3]{t}, t > \delta^3 \\ \dfrac{t}{3\delta^2} + 4/29, otherwise \end{cases}$$
$$\delta = \frac{6}{29}$$

Here Xn, Yn and Zn are the CIE XYZ tristimulus values of the reference white point.

IV. CHOICE OF DEEP LEARNING NETWORKS

To build high performance DCNN architecture it requires multiple hyperparameters to be tuned such as - number of hidden layers, convolution filters dimensions, stride and padding, pooling layer parameters, activation functions, dropout, learning rate, etc. All such hyperparameters define CNN model structure and its training procedure. One can tune these hyperparameters and have different structure of DCNN by considering its application, results obtained by tweaking hyperparameters values or amount of data available. There were multiple classical networks introduced which have reported good performance in object recognition task over the years. These classic networks provide inspiration to build network in terms of tuning hyperparameters, such as number of layers and their position, filter size, etc. Our goal here is to analyze whether DCNNs are invariant to color space or not and to be able to give such conclusion it is important to experiment with networks which can generalize multiple CNNs architectures. So we have chosen to experiment with classic convolution networks (discussed in this chapter), as variation of these architecture designs are widely used in many computer vision applications and many networks have reported best performance in object recognition tasks by following the same pattern of architecture as these networks.

Following are some of the CNN architectures we have used in our study –

A. VGGNet

VGGNet was introduced in paper “Very Deep Convolutional Neural Networks for Large-Scale Image Recognition”[17]. VGGNet has very simple, deep and uniform architecture. Its architecture was similar to “Alexnet”[18] but VGGNet has improved performance by replacing large sized kernels (filters) to multiple small sized kernels. They replaced 5*5 and 7*7 kernel with two 3*3 kernels and three 3*3 kernels respectively, which decreased number of parameters to be trained and increased depth of network.

VGGNet only have 3*3 sized kernels as it is smallest sized kernel to capture notion of left-right, up-down, center. VGGNet has many convolutional layers with RELU function as activation function and lot many 3*3 size filters in each convolutional layer which makes the network very deep and helps to learn more complex feature. There are different models of VGGNet available with varying number of depth or layers. For example VGG16, VGG19 and VGG22 have depth of 16, 19 and 22 layers in VGGNet. We have used VGG16 for our experiment.

VGGNet has introduced pattern regarding kernel dimensions and how different layers of CNNs should be arranged. Due to simple, uniform and deep architecture it is preferred network for many computer vision tasks and that is why we have decided to analyze VGGNet for our study. Architecture of VGGNet is as shown in Fig. 1.

VGG architecture with 16 weight layers
Input Layer (224*224 RGB Image)
Convolution (3*3)-64 Convolution (3*3)-64
Maxpool
Convolution (3*3)-128 Convolution (3*3)-128
Maxpool
Convolution (3*3)-256 Convolution (3*3)-256 Convolution (3*3)-256
Maxpool
Convolution (3*3)-512 Convolution (3*3)-512 Convolution (3*3)-512
Maxpool
Convolution (3*3)-512 Convolution (3*3)-512 Convolution (3*3)-512
Maxpool
Fully Connected– 4096 nodes
Fully Connected– 4096 nodes
Fully Connected– 1000 nodes
Softmax

Figure 1 VGG16 architecture with representation of each layer[17]

B. GoogLeNet

GoogLeNet was introduced in paper "Going Deeper with Convolutions"[19]. GoogLeNet was first CNN model which had different architecture from general approach of CNNs (e.g. stacking up convolution and pooling layers followed by fully connected layers and softmax layer). Increasing depth (number of levels in network) and width (number of units at each level) of network for better performance lead to high computation requirement. GoogLenet suggested sparsely connected architecture as solution of this problem and introduced concept of using inception module (shown in Fig. 2) as optimal local sparse structure. GoogLenet is a network constructed by stacking up multiple inception modules with occasional max-pooling layers.

As shown in Fig. 2, Inception module has convolutions of different sizes (5*5, 3*3, 1*1) and pooling filters to execute in parallel. Output of all these layers are concatenated as single output vector forming input for next inception module. Varying dimension of convolutions helps to extract information in detail and provides abstract features from different scales to the next layer. GoogLeNet uses global average pooling layer instead of fully connected layer before softmax layer.

We have considered GoogLeNet for our study as it has unique structuring of layers than other sequential CNN models and analyzing GoogLenet can help to cover CNNs having inception modules or similar kind of architecture. There are multiple version of Inception network is available such as InceptionV1, InceptionV2, InceptionV3, etc. We have used InceptionV3 for our experiment.

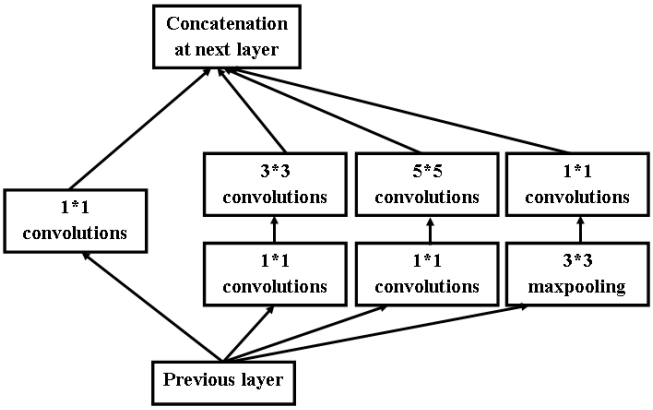


Figure 2 Inception module which is used in architecture of Inception CNNs[19]

C. ResNet

ResNet was introduced in paper "Deep Residual Learning for Image Recognition"[20]. Theory says that increasing number of layers in CNNs gives more accurate results, but sometimes very deep neural networks face degradation problem which decreases the accuracy of model. ResNet is CNN model which solved such problem by introducing skip connection to the network. ResNet has architecture similar to VGGNet, except it introduced module called residual module shown in Fig. 3.

As shown in Fig. 3 skip connections perform identity mapping and add its output to the output of stacked layers. By introducing skip connections deeper model will have training error less than its shallower counterpart, thus increasing number of layers won't degrade the performance. It is worth noticing that ResNet uses Identity mapping in short connection, so it does not add any extra parameters to the network and it does not increase complexity of network. ResNet also uses global averaging layer after convolutions layers, same as GoogLeNet.

ResNet uses only 3*3 sized convolutions same as VGGNet, but it contains shortcut connections in architecture which helps the network to have large number of layers than VGGNet without degradation problem and have deeper architecture with better accuracy. Reason of choosing ResNet is that ResNet and its various versions have achieved benchmark error rate in object classification task on multiple database such as CIFAR10[21] ,CIFAR100[21], ImageNet[22].

There are multiple ResNet architecture versions available based on number of layers in model such as ResNet50, ResNet101, etc. We have used ResNet50 architecture which has 50 layers in it.

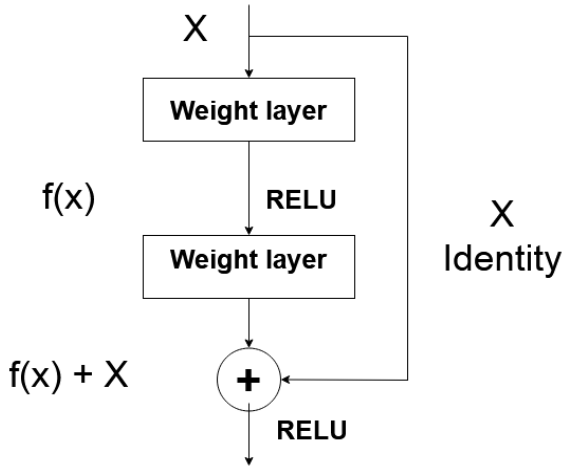


Figure 3 Building block of Residual learning network (ResNet) which has skip connections between layers[20]

V. EXPERIMENT

A. Dataset

We have considered two dataset for our study. All color space transformations are applied on each dataset and all three networks are trained on those data. Our Goal here is not to improve state of the art results but to analyze behavior of CNNs with respect to color spaces.

1) CIFAR 10 - CIFAR 10[21] data set is one of the classic datasets for object recognition application containing small sized color images. It has color images of 32*32 size each belongs to one of the ten classes:- airplanes, automobiles, birds, cats, deer, dogs, frogs, horses, ships, and trucks. It contains 50,000 training images and 10,000 test images, so each class has total 6000 images.

2) Linnaeus 5 - Linnaeus 5[23] dataset contains images of size 128*128, each belongs to one of the 5 classes - berry, bird, dog, flower, other (negative set). It has total 8000 images where each class has 1200 training images and 400 test images.

B. Methodology

All the DCNNs which we are using for our analysis have made their trained and optimized weights for imagenet[22] dataset freely available for use. So, we have taken advantage of those existing models by using transfer learning approach for training in our experiment as it is computationally efficient and requires less training data. Following are details regarding models, data preprocessing and training of these models –

1) Model - All three models are loaded with pretrained weights on Imagenet dataset. Fully connected layers are removed and input layer shape of these models are set to (75*75*3) for CIFAR10 dataset and (128*128*3) for Linnaeus 5 dataset.

In VGG16 model we have added fully connected layer of 512 neurons with RELU as activation function, while in Resnet50 and InceptionV3 network we have added global average pooling layer followed by fully connected layer of 512 neurons with RELU as activation function. Output layer of all these models are set to fully connected layer with softmax activation function for output class predictions. After loading network with pretrained weights and modifying its architecture according to our dataset, all the networks are retrained during experiment.

2) Preprocessing - Following step wise preprocessing has been applied on data before loading it to the network.

- Resizing – Among the networks we have considered for our experiments, Inception network requires dimensions of input images no less than 75*75. So for uniformity, We have resized input images of CIFAR10 from 32*32 to 75*75 for all three networks, whereas input images of Linnaeus 5 are taken with original dimensions.
- Color space transformation
- Normalization
- Data augmentation - For data-augmentation on training set we have applied
  - Random rotation to input images at 15˚ in the range of 0˚ to 180˚
  - Random horizontal shift in the range of 0.1 fraction of image width
  - Random vertical shift in the range of 0.1 fraction of image height
  - Random flip to images horizontally

3) Training - We have used Adam optimizer[24] and categorical cross-entropy as loss function to compile the model. We have set parameters values of Adam optimizer same as defined in its original paper. It was noted that model with higher learning rate was not able to give us good results in terms of loss function. Reason can be understood as networks we are using here are already initialized with Imagenet weights so we require small learning rate to retrain them. So, we have used 0.0001 as default learning rate. For CIFAR10 we have used batch size of 32, whereas for Linnaeus5 we have used batch size of 128.

To decide the number of epochs for training of all the three networks we considered the training accuracy, validation accuracy and also the training loss and validation loss at each epoch. Based on those values, we decided to train all the networks with 25 epochs for CIFAR10 dataset and 30 epochs for Linnaeus5 dataset.

4) Validation and Verification - We have manually specified dataset for verification at each epoch during training process of models. 25% of training images are considered as validation dataset to monitor validation accuracy and loss while training models.

5) Testing - To measure performance of each trained network we have computed three scores - test accuracy (TA), test loss (TL), and 0/1 validation loss (0/1 VL).

VI. EXPERIMENTAL RESULTS AND ANALYSIS

In this section we report and discuss the results of networks performances on each dataset for various color spaces. Fig. 4, 5 and 6 shows performances of networks during training in terms of validation loss and validation accuracy for each color spaces. Scores of test accuracy, test loss and validation loss of networks for every dataset and color spaces are reported in table 1, 2 and 3.

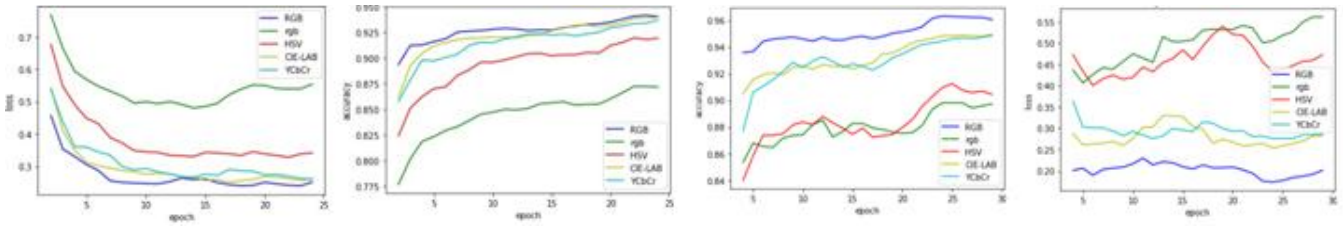
A. VGG16

Graph shown in Fig. 4 shows that each color space has similar trend of improvement in accuracy as well as in loss with increasing number of epochs for both the dataset.

However rgb and HSV performance is significantly lower both in terms of validation accuracy and loss when compared with RGB, CIE-LAB and YCbCr. Among all the color spaces, rgb color space is the worst performer in both the dataset. While comparing test accuracy, test loss and validation loss from table 1 it is clear that for CIFAR10 dataset RGB, CIE-LAB and YCbCr performed almost similarly during training, but CIE-LAB is clearly the best performer during testing. While for Linnaeus5 dataset RGB color space performed best among all color spaces, closely followed by CIE-LAB color space. But both the dataset results show that VGG16 is variant to color spaces.

Table I Performance comparison of VGG16[17] on CIFAR10[21] and Linnaeus5[23] dataset for all five color spaces

Color space	CIFAR10			Linnaeus5		
	TA	TL	0/1 VL	TA	TL	0/1 VL
RGB	0.9385	0.2832	0.0615	**0.9595**	**0.2028**	**0.0405**
HSV	0.9216	0.3265	0.0784	0.8920	0.6270	0.1080
YCbCr	0.9396	0.2725	0.0604	0.9460	0.3166	0.0540
CIE-LAB	**0.9417**	**0.2621**	**0.0583**	0.9525	0.2855	0.0475
rgb	0.8704	0.5727	0.1296	0.9085	0.5137	0.0915



(a) (b) (c) (d)

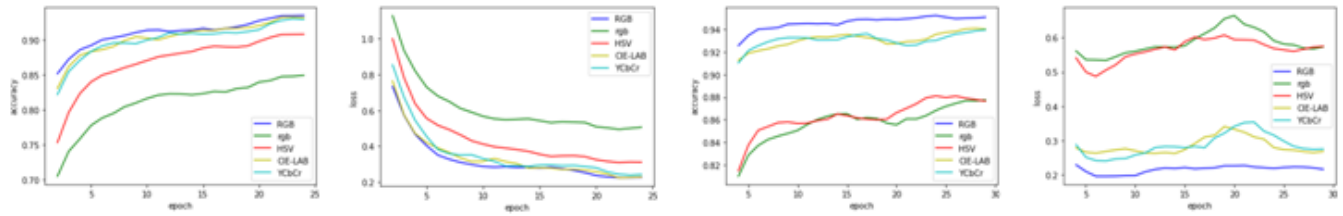
Figure 4 Performance of VGG16[17] on different color spaces in terms of accuracy and loss during training as function of epochs - figure (a) and (b) for CIFAR10[21] and figure (c) and (d) for Linnaeus5[23]

B. Inceptionv3

Graphs in Fig. 5 also shows that rgb and HSV performed quite poorly compared to RGB,YCbCr and CIE-Lab. While comparing TA, TL, and 0/1 VL reported in table 2, it is clear that InceptionV3 is also variant to color spaces. For CIFAR10 dataset RGB is the best performer for all the three scores and closely followed by CIE-LAB and YCbCr respectively. Linnaeus5 also has similar performance results with RGB turned out to be best performer, closely followed by YCbCr and CIE-LAB respectively.

Table II Performance comparison of Inception V3 [19] on CIFAR10[21] and Linnaeus5[23] dataset for all five color spaces

Color space	CIFAR10			Linnaeus5		
	TA	TL	0/1 VL	TA	TL	0/1 VL
RGB	**0.9362**	**0.2236**	**0.0638**	**0.9435**	**0.2615**	**0.0565**
HSV	0.9085	0.3131	0.0915	0.8685	0.5858	0.1315
YCbCr	0.9286	0.2486	0.0714	0.9405	0.2709	0.0595
CIE-LAB	0.9316	0.2298	0.0684	0.9330	0.3033	0.0670
rgb	0.8498	0.5152	0.1502	0.8820	0.5784	0.1180

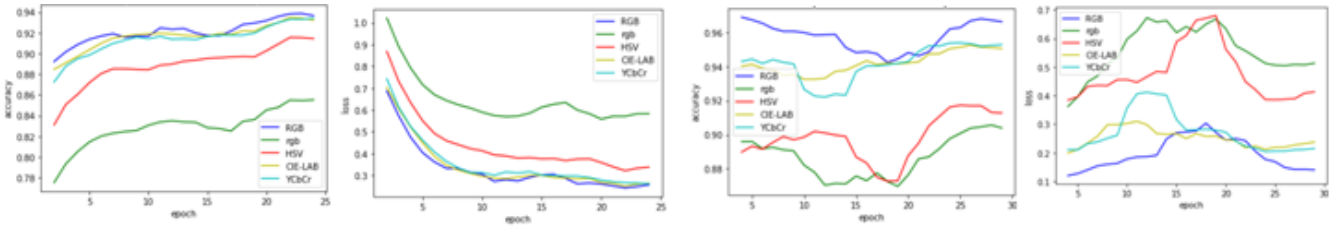


(a) (b) (c) (d)

Figure 5 Performance of InceptionV3[19] on different color spaces in terms of accuracy and loss during training as function of epochs - figure (a) and (b) for CIFAR10[21] and figure (c) and (d) for Linnaeus5[23]

Retrieval Number: B4038079220/2020©BEIESP
DOI:10.35940/ijrte.B4038.079220
Journal Website: www.ijrte.org

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication



(a) (b) c) (d)

Figure 6 Performance of ResNet50[20] on different color spaces in terms of accuracy and loss during training as function of epochs - figure (a) and (b) for CIFAR10[21] and figure (c) and (d) for Linnaeus5[23]

C. Resnet50

Fig. 6 shows the plots of validation accuracy and loss and the trend is similar to other networks, rgb is the worst performer, HSV is better than rgb but RBG and CIE-LAB and YCbCr are all equally better. The 3 scores of table 3 clearly shows that for CIFAR10 dataset YCbCr is the best performing color space closely followed by RGB and CIE-LAB. While for Linnaeus5 dataset RGB is best performing color space followed by YCbCr and CIE-LAB.

Table III Performance comparison of ResNet50[20] on CIFAR10[21] and Linnaeus5[23] dataset for all five color spaces

Color space	CIFAR10			Linnaeus5		
	TA	TL	0/1 VL	TA	TL	0/1 VL
RGB	0.9348	0.2619	0.0652	**0.9690**	**0.1422**	**0.0310**
HSV	0.9138	0.3393	0.0862	0.9140	0.4398	0.0860
YCbCr	**0.9359**	**0.2584**	**0.0641**	0.9580	0.1909	0.0420
CIE-LAB	0.9292	0.2726	0.0708	0.9540	0.2119	0.0460
rgb	0.8538	0.5935	0.1462	0.9080	0.4714	0.0920

Here our goal was not to compare these CNN architectures performances on different dataset, but to compare performance of each network in different color spaces. Different scores of test results for both the datasets shows that all the networks have variant behavior to color spaces. Among all color spaces rgb performed least in all the network architectures for both the dataset. It is a justifiable result as by transforming RGB to rgb partially the intensity information is lost. RGB color space still wins in terms of performance as most of these networks have been optimized for RGB space. Test results of Linnaeus5 dataset shows that RGB is best color space for all the networks, whereas results of CIFAT10 dataset showed that YCbCr and CIE-Lab have slightly better performance on ResNet50 and VGG16 networks respectively.

VII. CONCLUSION

In this paper we reported experiment with DCNN architectures like VGG16, InceptionV3 and ResNet50 by changing color space of input image dataset from RGB to YCbCr, CIE-Lab, HSV and rgb to examine whether these DCNNs are variant to color spaces or not. Test accuracy, test loss and validation loss of these networks in different color spaces shows that color spaces are affecting DCNNs performance, so DCNNs are 'not' invariant to color space.

Different color models represent image differently, some color space represents image in terms of hue and saturation values, some represents images based on how humans perceive colors, some stores illumination and chrominance information separately, etc. All these color spaces have different applications in real world based on its characteristics. Performance of CNNs with respect to color spaces may depend on various factors such as dataset, application of CNN, etc. If dataset we are experimenting with have wide illumination variations among its classes, we may classify it better by considering illumination information rather than color information. In such cases YCbCr or HSV may outperform RGB color space.

So as DCNNs have varying performance results with respect to color spaces, experimenting with color spaces may help to get better results for targeted dataset.

So, at this point we are having a conclusion that color spaces do affect DCNNs performance and experimenting with color spaces may help to get better results.

REFERENCES

1. S. Belongie, J. Malik, and J. Puzicha, "Shape matching and object recog-nition using shape contexts,"IEEE Transactions on Pattern Analysis &Machine Intelligence, no. 4, pp. 509–522, 2002.W.-K. Chen, *Linear Networks and Systems* (Book style). Belmont, CA: Wadsworth, 1993, pp. 123–135.
2. D. G. Lowe, "Object recognition from local scale-invariant features," inComputer vision, 1999. The proceedings of the seventh IEEE interna-tional conference on, vol. 2. Ieee, 1999, pp. 1150–1157.B. Smith, "An approach to graphs of linear forms (Unpublished work style)," unpublished.
3. D. G. Lowe, Distinctive image features from scale-invariant keypoints,"Inter-national journal of computer vision, vol. 60, no. 2, pp. 91–110, 2004.
4. J . Talmi, R. Mechrez, and L. Zelnik-Manor, "Template matching withdeformable diversity similarity," in2017 IEEE Conference on ComputerVision and Pattern Recognition (CVPR). IEEE, 2017, pp. 1311–1319..
5. M. J. Swain and D. H. Ballard, "Color indexing,"International journalof computer vision, vol. 7, no. 1, pp. 11–32, 1991.
6. M. Mercimek, K. Gulez, and T. V. Mumcu, "Real object recognitionusing moment invariants,"sadhana, vol. 30, no. 6, pp. 765–775, 2005.
7. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust wide-baselinestereo from maximally stable extremal regions,"Image and visioncomputing, vol. 22, no. 10, pp. 761–767, 2004.
8. Kumar, P.; Dick, A. Adaptive earth movers distance-based Bayesian multi-target tracking. IET Computer Vision 2013, 7, 246–257.
9. P. Kumar and S. J. Miklavcic, "Analytical study of colour spaces forplant pixel detection,"Journal of Imaging, vol. 4, no. 2, 2018. [Online].Available: http://www.mdpi.com/2313-433X/4/2/42
10. S. L. Phung, A. Bouzerdoum, and D. Chai, "Skin segmentation usingcolor pixel classification: analysis and comparison,"IEEE transactionson pattern analysis and machine intelligence, vol. 27, no. 1, pp. 148–154, 2005.
11. K Shaik,., Packyanathan, G., Kalist, V., B.S, S., Merlin Mary Jenitha,J.: Comparative study of skin color detection and segmentation in hsv and ycbcr color space. Procedia Computer Science57, 41–48 (12 2015). https://doi.org/10.1016/j.procs.2015.07.362

12. P. Kumar, K. Sengupta, and A. Lee, "A comparative study of differentcolor spaces for foreground and shadow detection for traffic monitoringsystem," inIntelligent Transportation Systems, 2002. Proceedings. TheIEEE 5th International Conference On. IEEE, 2002, pp. 100–105.
13. D. Cires¸An, U. Meier, J. Masci, and J. Schmidhuber, "Multi-columndeep neural network for traffic sign classification,"Neural networks,vol. 32, pp. 333–338, 2012.
14. D. Khattab, H. M. Ebied, A. S. Hussein, and M. F. Tolba, "Color imagesegmentation based on different color space models using automaticgrabcut,"The Scientific World Journal, vol. 2014, 2014.
15. S.N Gowda , C Yuan, ColorNet: Investigating the importance of color spaces for image classification, 2019, [arXiv:cs.CV/1902.00267].
16. W. contributors, "List of color spaces and their uses — Wikipedia,the free encyclopedia," https://en.wikipedia.org/w/index.php?title=Listofcolorspacesandtheiruses&oldid=886629037, 2019.
17. K. Simonyan and A. Zisserman, "Very deep convolutional networks forlarge-scale image recognition,"arXiv preprint arXiv:1409.1556, 2014.
18. A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classificationwith deep convolutional neural networks,"Commun. ACM, vol. 60,no. 6, pp. 84–90, May 2017. [Online]. Available: http://doi.acm.org/10.1145/3065386
19. C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan,V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions,"arXiv preprint arXiv:1409.4842, 2014.
20. K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for imagerecognition,"2016 IEEE Conference on Computer Vision and PatternRecognition (CVPR), pp. 770–778, 2016.
21. A. Krizhevsky, "Learning multiple layers of features from tiny images,"Tech. Rep., 2009.
22. J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet:A Large-Scale Hierarchical Image Database," inCVPR09, 2009.
23. G.K.L. Chaladze, "Linnaeus 5 Dataset for Machine Learning",Tech. Rep., 2017.
24. D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," arXiv preprint arXiv:1412.6980, 2014.
25. Kumar P., Shingala M. (2021) Native Monkey Detection Using Deep Convolution Neural Network. In: Hassanien A., Bhatnagar R., Darwish A. (eds) Advanced Machine Learning Technologies and Applications. AMLTA 2020. Advances in Intelligent Systems and Computing, vol 1141. Springer, Singapore
26. Doshi N, Oza U, Kumar P. Diabetic Retinopathy Classification using Downscaling Algorithms and Deep Learning. In2020 7th International Conference on Signal Processing and Integrated Networks (SPIN) 2020 Feb 27 (pp. 950-955). IEEE.

AUTHORS PROFILE

Urvi Oza received the BE degree in information technology from Vishwakarma Government Engineering College, Gandhinagar, Gujarat and Mtech degree in information and communication technology from Dhirubhai Ambani Institute of Information and Communication Technology (DAIICT), Gandhinagar, Gujarat. Currently she is pursuing her Ph.D degree from DAIICT. Her research interest includes Computer vision, Image Processing, Artificial intelligence and Medical image analysis.

Pankaj Kumar received the B.Tech degree in electrical and computer engineering from the Indian Institute of Technology, Delhi, India, and the M.Eng. and Ph.D. degrees from the National University of Singapore, Singapore. Currently he is an Associate Professor at DAIICT, Gandhinagar, Gujarat. Prior to joining DAIICT he was affiliated with University of South Australia and University of Adelaide and also worked as Scientist and Associate Scientist at Defense Science and Technology Organization, Australia and Institute of Infocomm Research, Singapore, respectively. His research interests include computer vision, artificial intelligence, large data mining and analysis, behavior analysis, multicamera tracking, and surveillance. He is program committee member of several IEEE conferences and is serving as guest editor for MDPI journal Technology.