Does processing in the canine vision spectrum retain performance?

Project report for IITB EE610 Image Processing 2021

Aaryan Gupta
Electrical Engineering
19D070001

Eeshaan Jain
Electrical Engineering
19D070022

Vipin Singh
Electrical Engineering
19D070069

***Abstract*—One of the fundamental tasks of computer vision can be stated to be image classification. With the advent of faster processing and construction of deeper neural networks, many architectures have broken all previous records in this field. It is a standard to take in the images in the RGB color format to transform in the neural networks, but it has been shown various times that color spaces can indeed affect the performance of networks in the task of image classification.**

The current advances in robotic animals have attracted significant attention from the computer vision domain to re-enact the behavior of animals in response to stimuli from the environment. The canines view the world differently and not in the same RGB sense humans are used to. For now, the approaches to image-related tasks for such robots are done in the RGB spectrum, but we explore the possibilities of doing the same in the canine vision spectrum. Especially, we aim to review popular classification models and test the performance of the models in the canine vision spectrum. The canine vision spectrum is dichromatic in contrast to the RGB spectrum.

We found out that DenseNet performs marginally better in the canine vision spectrum in both - low parameter and high parameter settings. In ResNet and EfficientNet, the RGB data performed slightly better than the transformed data. This indicates that processing in the canine vision spectrum does not hinder the performance by a large margin and with further advances, it might be a viable option to work with. Although, a drastic improvement cannot be expected if viewed from the canine vision only.

***Index Terms*—image classification, canine vision spectrum, color spaces, DenseNet, ResNet, EfficientNet**

I. Introduction

One of the most fundamental applications in the field of computer vision is image classification. Since the advent of deep learning techniques to achieve the state-of-the-art goals in this task, it has been a norm to resort to the RGB color space for processing and forward implementations. The RGB color space, often represented by a 24-bit implementation (each R, G, B component is 8 pixels), is one of the simplest color spaces and is readily understood by humans. A question which arises is, *why do we stick to the RGB color space as a standard in training such models?*.

It is known that most animals view the world differently than we humans do (Fig 1). A natural consequence is a fact that they might be able to perceive notions and objects in a more enhanced or restricted way as compared to what we do. Apart from that, the recent advances in animal robotics have opened interest in creating robotic animals, completely enacting the real animals in all aspects. With advances in this field, one must question all the key aspects which differentiates animals and try to bridge the gap between them. One of the key aspects is vision, and as stated above, animals see the world differently. In particular, robotic dogs currently present have their vision in the RGB domain. That is something that brings in photo unrealism, as dogs, in particular, can capture wavelengths that we cannot. For example, the `Sony Aibo` is known to have visual programming done in the RGB domain. This project benchmarks the performance of standard deep learning models for the task of image classification with 2 classes: Dogs and Cats. Especially, we aim to check if processing in the canine vision spectrum harms the performance of these models. It is known that the transformation of images from the RGB domain to the canine vision spectrum is non-linear, and hence it is possible that the CNN learns differently. Section II-D elucidates more on this.

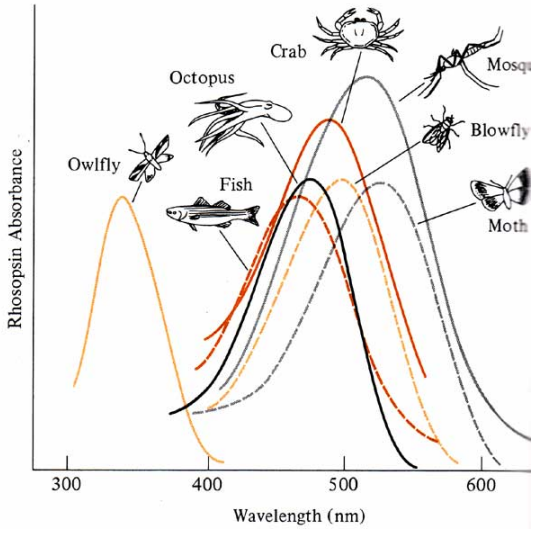


Fig. 1: Vision spectrum of different species. Credits: Vision2

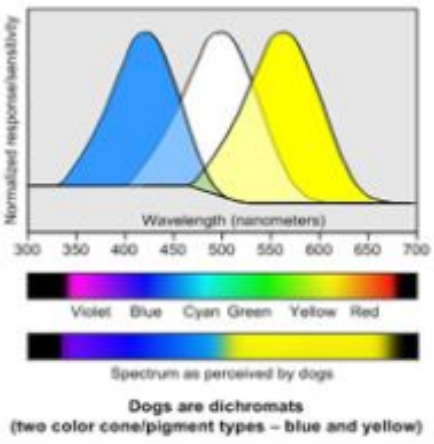


Fig. 2: USCB MAT: Canine Vision Spectrum‘

‘

II. Background and Prior Work

A. Canine Vision Spectrum

It is known that dogs possess two-color receptors, one sensitive to long/medium wavelength (555 nm spectral sensitivity; red/green) and the other sensitive to short-wavelength (429 nm spectral sensitivity; blue). The presence of these two discrete color receptors indicates a potential dichromatic vision.
Siniscalchi et al. [1], showed that dogs perceive images similar to a person suffering from deuteranopia color blindness (green color-blindness). The lens of the human eye blocks ultraviolet light, but in dogs, ultraviolet light reaches the retina, which converts the light into nerve signals that travel to the brain, where the visual system perceives them.
Deuteranopia color blindness is the most common type of color deficiency. In this color blindness, the medium wavelength sensitive cones (green) are missing. The deuteranope can only distinguish 2 to 3 different hues, whereas somebody with normal vision sees seven different hues. The transformation of an image from normal vision to canine vision is a nonlinear transformation due to residual clipping.

B. Classical methods for classification

Some of the machine-learning methods commonly used for Binary classification include decision trees, random forest, Logistic Regression, Naive Bayes, Support Vector Machines, etc.
SVMs work well in higher dimensions & are well-suited for extreme case binary classification; however, they are very slow & show poor performance with overlapping classes. Naive Bayes is very fast & scalable with large datasets; however, it is a bad estimator of probability. Logistic Regression is simple, effective without any requirement of hyper-parameter tuning; however, it's not a very powerful algorithm & shows poor performance with non-linear data. Random Forest is an ensemble of decision trees. It minimizes overall variance & reduces error. It works well with large-imbalanced datasets & shows more generalization with lesser over-fitting; however, it needs a very specific kind of data & is difficult to interpret.

C. Deep Learning for classification

Machine Learning Algorithms work best for small-size datasets; however, as the size of the dataset increases, deep Learning techniques outperform other techniques. Whenever there is a lack of domain understanding for feature introspection, deep learning techniques outshines others as you don't have to worry much about feature engineering.
The LeNet architecture was first introduced by LeCun et al. [2]. It is a fairly straightforward Convolution Neural Network that works to extract meaningful & low-dimensional features to work on. These high-level features are flattened and fed to fully connected layers, which will eventually yield class probabilities through a softmax layer.
As the application of CNNs in solving classification problems started to grow, AlexNet was introduced [3] in order to leverage the full potential of CNNs. It is a deeper CNN with more convolutional & pooling layers, consequently capturing finer features useful for image classification. It used the ReLU activation layer, which proved better for gradient propagation & thus showed better non-linearity. It also introduced dropout as regularization as well as data augmentation, which increased the robustness of the model.
After a certain point of time, it was seen that stacking more layers didn't improve the performance of the model; the opposite occurred. To counter this effect, the concept of Residual Networks was introduced [4]. After every 2 layers, there is an identity mapping via an element-vise addition which proved to be helpful in gradient propagation & reducing error. It also helped combine different levels of features at each step of the network. Extension of this reasoning was proposed as DenseNet [5] wherein entire blocks of layers were connected, diversifying a lot more features within those blocks.

D. Prior Work

The way CNN works is that they take the input data to an abstract multidimensional latent space, with feature maps beyond comprehension after a particular stage. The goal is to learn those feature maps, and it might seem that since the representation is abstract, it will be independent of the various representations of input values. However, that is something that has been shown not to be true. Shreyank N Gowda et al. [6] showed that we show that certain classes of images are better represented in particular color spaces, and sometimes it is even better to merge the color spaces to get higher dimension inputs to improve performance. It has also been noted that passing RGB values through strong hash functions makes classifying the images much harder. Although almost one-to-one mapping exists, it implies that the input space matters for the CNN output.
The CaffeNet benchmarking done on ImageNet-2012 for 19 different color spaces showed the performance variation, as depicted in Fig 3. As of now, there has been no benchmarking

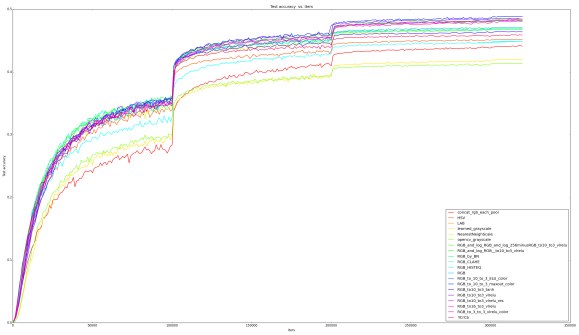


Fig. 3: CaffeNet Benchmarking

of classification tasks in the canine vision spectrum, hence this is the first time that such a benchmarking is being done.

III. Data and Methodology

A. Dataset Source

The dataset used is a subset of the ASIRRA (Animal Species Image Recognition for Restricting Access) dataset, which consisted of three million images of cats and dogs, and the images present have been manually classified into the two classes we are interested in: `Cats` and `Dogs`.
The dataset had been provided to Microsoft Research and is available online on the following link: Kaggle Cats and Dogs Dataset.
Note that the original dataset had 25000 images in the training set. That has been reduced to 20000, with 3000 images used in the test set and 2000 images discarded. Purposely, noisy images (those with texts or humans) have been kept in the dataset to increase robustness to a certain extent.

B. Data Preprocessing

To transform an image in the RGB color space from normal vision to deuteranopic vision, we can apply the following matrix transformations [7].
Transform the image from RGB color space to XYZ color space using the following matrix transformation.

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.4124 & 0.3756 & 0.1805 \\ 0.2126 & 0.7152 & 0.0722 \\ 0.0193 & 0.1192 & 0.9505 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{1}$$

Transform the image from XYZ color space to LMS color space using the following matrix transformation.

$$\begin{bmatrix} L \\ M \\ S \end{bmatrix} = \begin{bmatrix} 0.1551 & 0.5431 & -0.03286 \\ -0.1551 & 0.4568 & 0.03286 \\ 0 & 0 & 0.01608 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \tag{2}$$

The LMS color space makes it easier to simulate the color model of color vision defective by adjusting the reaction value of the LMS cone.
The transformation given below is used to convert the image from the human perceived color spectrum to canine perceived color spectrum.

$$\begin{bmatrix} L_d \\ M_d \\ S_d \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0.49 & 0 & 1.25 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} L \\ M \\ S \end{bmatrix} \tag{3}$$

The image seen by a deuteranope can be transformed from the LMS color space into RGB color space using the following matrix transformation.

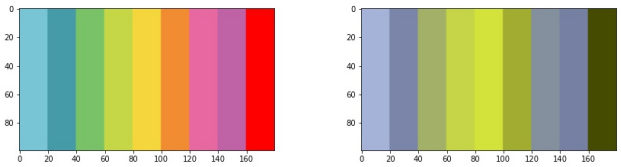
$$\begin{bmatrix} R_d \\ G_d \\ B_d \end{bmatrix} = \begin{bmatrix} 8.0655 & -13.1980 & 12.2918 \\ -0.9953 & 5.3625 & -10.3650 \\ -0.0380 & -0.4059 & 66.4792 \end{bmatrix} \begin{bmatrix} L_d \\ M_d \\ S_d \end{bmatrix} \tag{4}$$

As the above transformations are linear, they can be implemented in fewer steps by the following transformations.

$$\begin{bmatrix} X_d \\ Y_d \\ Z_d \end{bmatrix} = \begin{bmatrix} 0.1884 & 0.6597 & 0.1016 \\ 0.2318 & 0.8116 & -0.0290 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \tag{5}$$

$$\begin{bmatrix} R_d \\ G_d \\ B_d \end{bmatrix} = \begin{bmatrix} 3.2979 & -1.6484 & -0.5010 \\ -0.9861 & 1.9090 & 0.0422 \\ 0.0570 & -0.2061 & 1.0569 \end{bmatrix} \begin{bmatrix} X_d \\ Y_d \\ Z_d \end{bmatrix} \tag{6}$$

Note : In the end, residual clipping is performed to keep the intensity level between 0 and 255, making it a non-linear transformation.
Figure 4 is an intensity transformation of normal vision to canine vision.



(a) (b)

Fig. 4: Sample Image: (a) Image seen by Human, (b) Image seen by Dog

A transformation example from dataset is given in figure 5.

(a) (b)

Fig. 5: Sample Image: (a) Image seen by Human, (b) Image seen by Dog

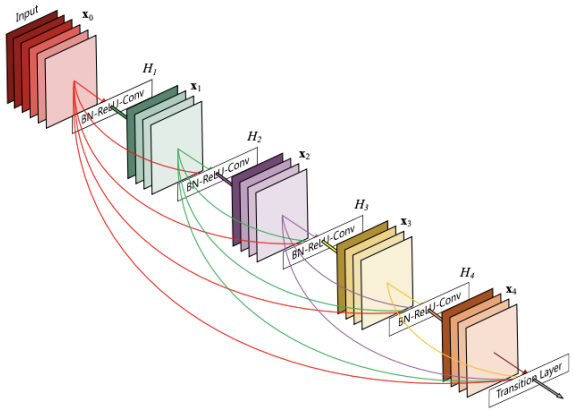


Fig. 6: Architecture of DenseNet Model Credits: Pytorch

The above transformation is applied to each image in the dataset to produce a canine vision spectrum dataset of cats and dogs.

C. Deep Learning Models

All the deep learning models mentioned forth have been benchmarked twice on different scales - larger image and number of parameters (L_p), and smaller image and number of parameters (S_p) (specific details mentioned ahead). This allows the comparison of models based on more and less trainable parameters. This has been done by reducing the layer parameters in the fully connected layers, and reducing the iamge size. The L_p set has the input size as 224×224 while the S_p set has the input size as 160×160.

1) DenseNet (Transfer Learning): DenseNet was first introduced by Gao Huang et al [5]. In DenseNet, each layer in the model receives input from all preceding layers, and the layer passes on its feature maps to all subsequent layers. Therefore, each layer receives knowledge from all preceding layers. As a result, the network can be made thinner or compact, i.e., the number of channels can be fewer.

DenseNets are divided into DenseBlocks where the size of feature maps does not change within the block, but the number of filters between them changes. The number of channels is reduced to half of the existing layer in the transition layer between the blocks.

In DenseNet-121 architecture (Fig 6), dense blocks have varying numbers of layers (repetitions) featuring two convolutions each; a 3x3 kernel to perform the convolution operation and a 1x1 sized kernel as the bottleneck layer. Also, each transition layer has a 2x2 average pooling layer with a stride of 2 and a 1x1 convolutional layer. Thus, the layers present are as follows:

- Basic convolution layer with 64 filters of size 7X7 and a stride of 2
- Basic pooling layer with 3x3 max pooling and a stride of 2
- Dense Block 1 with two convolutions repeated six times
- Transition layer 1 (1 Conv + 1 AvgPool)
- Dense Block 2 with two convolutions repeated 12 times
- Transition layer 2 (1 Conv + 1 AvgPool)
- Dense Block 3 with two convolutions repeated 24 times
- Transition layer 3 (1 Conv + 1 AvgPool)
- Dense Block 4 with two convolutions repeated 16 times
- Global Average Pooling layer- accepts all the feature maps of the network to perform classification
- Output layer

Therefore, DenseNet-121 has the following layers:

- One 7x7 Convolution
- 58 3x3 Convolution
- 61 1x1 Convolution
- 4 AvgPool
- 1 Fully Connected Layer

In short, DenseNet-121 has 120 Convolutions and 4 AvgPool. All layers, i.e., those within the same dense block and transition layers, spread their weights over multiple inputs, allowing deeper layers to use features extracted early on.

Modification: The fully-connected layer has been modified in the end as follows:

- L_p: $\binom{1024}{512} \triangleright ReLU \triangleright \binom{512}{256} \triangleright ReLU \triangleright \binom{256}{2} \triangleright LogSoftmax$
- S_p: $\binom{1024}{128} \triangleright ReLU \triangleright \binom{128}{2} \triangleright LogSoftmax$

2) ResNet (Transfer Learning): ResNet is an innovative neural network first introduced by Kaiming He et al [4]. A residual neural network (ResNet) is an artificial neural network (ANN) that builds a network by stacking residual blocks on top of one another.

ResNets use residual blocks to improve the accuracy of the model. Skip connection, which lies at the core of the residual block, is the strength of residual neural networks. Skip connections combine the outputs of previous layers with the outputs of stacked layers, allowing deeper networks to be trained.

The skip connection in ResNet works to alleviate the vanishing gradient issue by setting an alternate and shot path for the gradient to pass through to the previous layer. Also, they enable the model to learn an identity function, ensuring that the higher layers of the model perform at par with the lower layers of the model. As a result, the model's efficiency improves with more neural layers while minimizing the percentage of errors.

ResNet-50 is a deep learning model that is 50 layers deep. ResNet-50 architecture (Fig 7) is based on the ResNet-34 model; there is one significant difference. In ResNet-50, the building block is modified into a bottleneck design due to concerns over training the layers. Therefore, ResNet-50 used a stack of 3 layers instead of 2.

The above modification helped ResNet-50 to achieve much higher accuracy than the 34-layer ResNet model.

Modification: The fully-connected layer has been modified in the end as follows:

- L_p: $\binom{2048}{512} \triangleright ReLU \triangleright \binom{512}{256} \triangleright ReLU \triangleright \binom{256}{2} \triangleright LogSoftmax$
- S_p: $\binom{2048}{256} \triangleright ReLU \triangleright \binom{256}{2} \triangleright LogSoftmax$

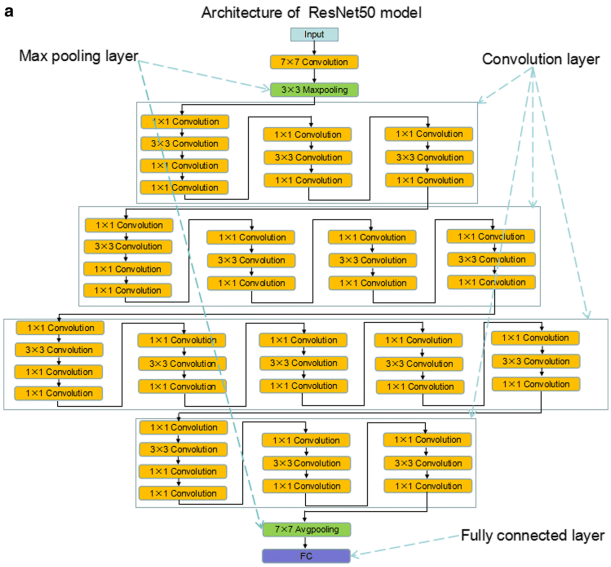


Fig. 7: Architecture of ResNet-50 Model, Credits: Springer

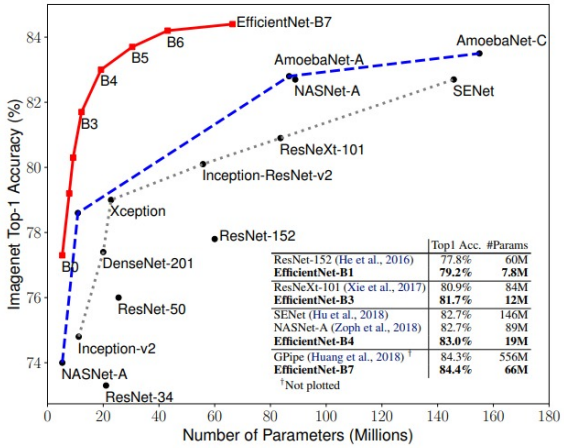


Fig. 8: ModelSize v/s Accuracy, Credits: [8]

3) EfficientNet (Transfer Learning): EfficientNet was first introduced by Google AI [8]. It is a convolutional neural network architecture and scaling method that uniformly scales all dimensions of network width, depth, and resolution with a fixed set of scaling coefficients, unlike conventional practice that arbitrarily scales these factors. The scaling method is justified by the intuition that if the input image is bigger, then the network needs more layers to increase the receptive field and more channels to capture more fine-grained patterns on the bigger image.

From Fig 8 it can be inferred that with considerably fewer numbers of parameters, the family of models are efficient and also provide better results. Flowchart representation of EfficientNet-B1 Architecture has been shown in Fig 9.

Modification: The fully-connected layer has been modified in the end as follows:

- L_p: $Dropout_{0.2} \triangleright \binom{1280}{512} \triangleright ReLU \triangleright \binom{512}{256} \triangleright ReLU \triangleright \binom{256}{2} \triangleright LogSoftmax$
- S_p: $Dropout_{0.2} \triangleright \binom{1280}{128} \triangleright ReLU \triangleright \binom{128}{2} \triangleright LogSoftmax$

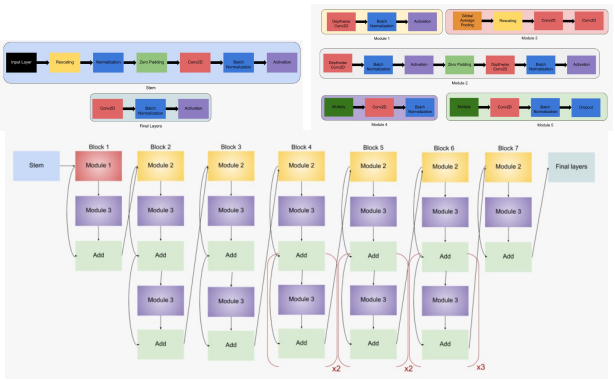


Fig. 9: Architecture of EfficientNet-B1 Model, Credits: TDS

D. Rest of the architecture

1) Adam: Adam [9] is a first-order gradient-based technique for optimizing stochastic objective functions based on adaptive estimations of lower-order moments. Adam is simple to implement, computationally efficient, has low memory requirements, is insensitive to gradient re-scaling, and is well suited for vital data or parameters problems.

$$m_t = \beta_1 + (1 - \beta_1)\left[\frac{\delta L}{\delta w_t}\right] \tag{7}$$

$$v_t = \beta_2 + (1 - \beta_2)\left[\frac{\delta L}{\delta w_t}\right]^2 \tag{8}$$

where,

- ϵ - a small +ve constant to avoid 'division by 0' error when $(v_t \to 0)$. (10^{-8})
- β_1 & β_2 - decay rates of average of gradients in the above two methods. (β_1 = 0.9 & β_2 = 0.999)
- α — Step size parameter / learning rate (0.003)

2) Negative Log Likelihood Loss: Negative Loss Likelihood (NLL) [10] is a cost function used as loss for machine learning models, telling us how bad it is performing; the lower, the better. Any loss consisting of a negative log-likelihood is a cross-entropy between the empirical distribution defined by the training set and the probability distribution defined by the model. Mathematically, NLL can be represented as:

$$NLLLoss = -\frac{1}{N}\sum_{k=1}^{N} y_k(LogSoftmax) \tag{9}$$

3) Batch Size: Batch size is a term used in machine learning and refers to the number of training examples utilized in one iteration. The Batch size can be one of three options:

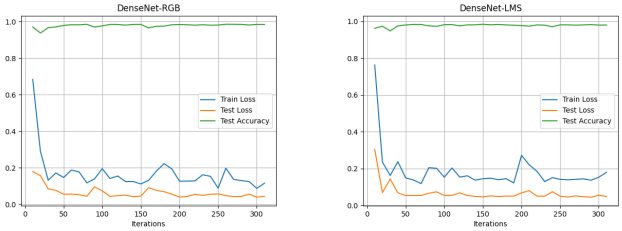
- Batch mode, where the batch size is equal to the total dataset, thus making the iteration & epoch values equivalent.

- Mini-batch mode, where a batch size is usually a number that can be divided into the total dataset size.
- Stochastic mode, where the batch size is equal to 1. Therefore, the gradient & network parameters update after each sample.

In our work, we have employed mini-batch mode with a batch size of 64 for the training dataset & 128 for the test dataset. This results in 313 iterations on training data & 24 iterations on test data.

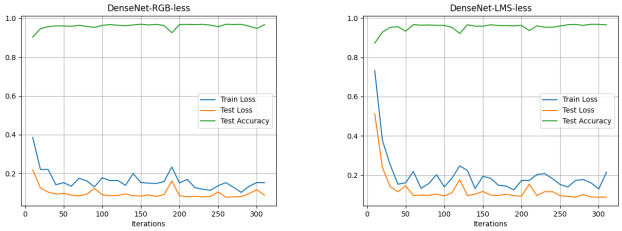
IV. Experiments and Results

A. DenseNet



(a) Human View (b) Canine View

Fig. 10: Metrics: Larger number of parameters

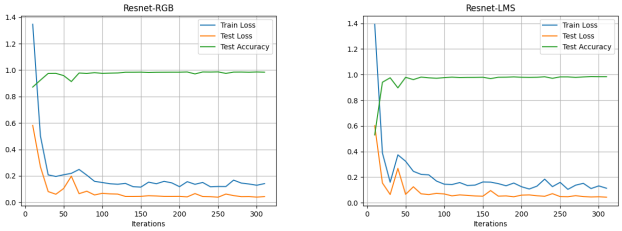


(a) Human View (b) Canine View

Fig. 11: Metrics: Lesser number of parameters

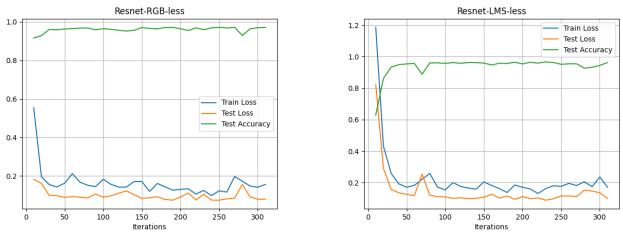
DenseNet-121 obtains maximum test accuracy of 98.6% and 96.9% on the L_p and S_p settings for images perceived by humans and maximum test accuracy 98.9% and 96.8% on the L_p and S_p settings for images perceived by canine.

B. ResNet



(a) Human View (b) Canine View

Fig. 12: Metrics: Larger number of parameters

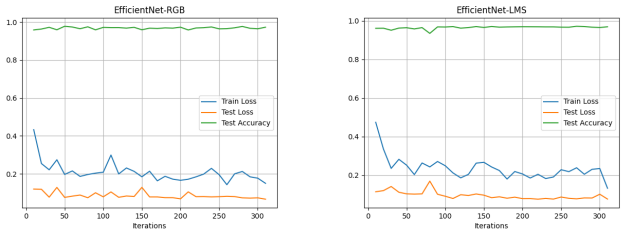


(a) Human View (b) Canine View

Fig. 13: Metrics: Lesser number of parameters

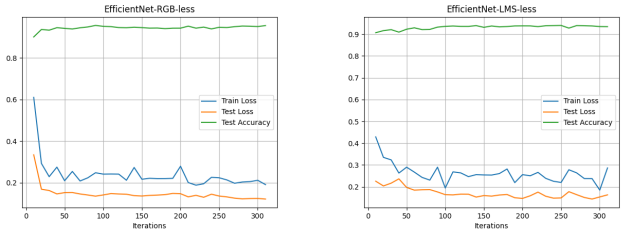
ResNet-50 obtains maximum test accuracy of 98.6% and 97.1% on the L_p and S_p settings for images perceived by humans and maximum test accuracy 98.4% and 96.6% on the L_p and S_p settings for images perceived by canine.

C. EfficientNet



(a) Human View (b) Canine View

Fig. 14: Metrics: Larger number of parameters



(a) Human View (b) Canine View

Fig. 15: Metrics: Lesser number of parameters

EfficientNet-B1 obtains maximum test accuracy of 97.7% and 95.6% on the L_p and S_p settings for images perceived by humans and maximum test accuracy 97.2% and 94.0% on the L_p and S_p settings for images perceived by canine.

V. Learning, Conclusions, and Future Work

A. Learning

- ◇ Learnt about new color spaces like XYZ and LMS color spaces and how to model canine vision spectrum using simple matrix transformation and clipping.
- ◇ Came to know about the evolution of image classification and the development of various deep CNNs in order to reduce errors & provide better accuracy while maintaining the complexity of the model.

Model	Spectrum	#Parameters (total)	#Parameters (fc)	Test accuracy
DenseNet	RGB (L_p)	7.61M	656.6k	98.42%
	RGB (S_p)	7.085M	131.5k	96.22%
	LMS (L_p)	7.61M	656.6k	98.66%
	LMS (S_p)	7.085M	131.5k	96.60%
ResNet	RGB (L_p)	24.69M	1.18M	98.48%
	RGB (S_p)	24.03M	525k	96.04%
	LMS (L_p)	24.69M	1.18M	98.24%
	LMS (S_p)	24.03M	525k	94.36%
EfficientNet	RGB (L_p)	7.3M	787.7k	97.04%
	RGB (S_p)	6.67M	164.2k	95.26%
	LMS (L_p)	7.3M	787.7k	96.86%
	LMS (S_p)	6.67M	164.2k	93.70%

TABLE I: Comparing number of parameters and performance of the models on the two color spectrums

- ◇ Comparison of different neural nets and how can they be improved upon for better image classification along with prevention of over-fitting by involving huge amount of parameters.
- ◇ Came to know about the advances in robotic animals.
- ◇ Saw how colorspaces affect performance on computer vision related deep learning tasks, and how one can leverage the correct color space(s) for better performance.

B. Conclusion

The project aimed to benchmark standard models over the RGB color spectrum and canine vision spectrum and compare the performance. The results (average of last 5 test accuracies) have been mentioned in Table I. We notice that the CVS performed better on the DenseNet-121 model, while it lagged slightly behind on the ResNet-50 and EfficientNet-B1 models over both L_p and S_p settings. This indicates that the fact that the canines see objects in a different spectrum does not affect their object recognition capabilities to a large extent, and they rely on structural information as well. Although this does not suggest that CVS is a bad choice, incorporating extra information from the surroundings can still have the upper hand when implementing actual robotics.

C. Future Work

The following aspects were left, and can be checked when further expanding on the project:

- ◇ Incorporate UV information to sense information outside the visible spectrum to allow more tasks.
- ◇ Further the benchmarking and comparing with other color spaces.
- ◇ Compare spectrums of different animals to compare classification powers
- ◇ Stack multiple color spaces and then benchmark to see if that improves performance.

CONTRIBUTION

We all explored different color spectrum, literature reviews for various deep learning models and contributed equally to the report. Eeshaan Jain, Aaryan Gupta and Vipin Singh implemented like ResNet-50, DenseNet-121, and EfficientNet-B1 respectively with varying number of parameters. Eeshaan Jain explored different datasets for the problem statement. Vipin Singh transformed the dataset from human-perceived images to canine-perceived images after exploring the literature review of the canine vision spectrum. Aaryan Gupta made the power point presentation for the video.

ACKNOWLEDGEMENTS

REFERENCES

[1] M. Siniscalchi, S. d'Ingeo, S. Fornelli, and A. Quaranta, "Are dogs red–green colour blind?" *Royal Society Open Science*, vol. 4, no. 11, p. 170869, 2017.

[2] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[3] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, pp. 1097–1105, 2012.

[4] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[5] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.

[6] S. N. Gowda and C. Yuan, "Colornet: Investigating the importance of color spaces for image classification," in *Asian Conference on Computer Vision*. Springer, 2018, pp. 581–596.

[7] H. Orii, H. Kawano, N. Suetake, and H. Maeda, "Color conversion for color blindness employing multilayer neural network with perceptual model," in *Image and Video Technology*, T. Bräunl, B. McCane, M. Rivera, and X. Yu, Eds. Cham: Springer International Publishing, 2016, pp. 3–14.

[8] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *International Conference on Machine Learning*. PMLR, 2019, pp. 6105–6114.

[9] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[10] D. Zhu, H. Yao, B. Jiang, and P. Yu, "Negative log likelihood ratio loss for deep neural network classification," *CoRR*, vol. abs/1804.10690, 2018.