Identifying Vehicle Exterior Color by Image Processing and Deep Learning

by

Somayeh Abniki

A Report Submitted in Partial Fulfillment of the Requirements for the Degree of

MASTER OF ENGINEERING

In the Department of Electrical and Computer Engineering

University of Victoria

SUPERVISORY COMMITTEE

Identifying Vehicle Exterior Color by Image Processing and Deep Learning

by

Somayeh Abniki

University of Victoria, 2022

Supervisory Committee

Dr. Kin Fun Li, Department of Electrical and Computer Engineering

Supervisor

Dr. Wei Li, Department of Electrical and Computer Engineering

Departmental Member

Abstract

The vehicle's color is one of the factors considered in car purchasing. Hence, color extraction and identification from online vehicle images play an important role in the vehicle e-commerce marketplace. In this project, we present a vehicle color identification methodology. Image processing techniques are employed to construct feature vectors, which are then used as input to deep neural networks to classify a vehicle's color into 14 classes. Local relative entropy is utilized as a measure of image segmentation to select the region of interest. Experiments are performed on an image dataset provided by an automobile e-commerce operator. Our implementation results are evaluated and discussed.

Table of Contents

List of Figures

LIST OF TABLES

Acknowledgments

I would like to thank my supervisor, Dr. Kin Fun Li, for his constant guidance, endless support, valuable feedback and motivation for my project and throughout my program.

Dedication

Dedicated to my wonderful children, Ali and Ilia and my supportive husband who encouraged me during this project.

Chapter 1: Introduction to Color Recognition

Color recognition plays a crucial role in many applications. Vehicle color recognition is an important research and application component in Intelligent Transport Systems [1] [2] [3] [4]. With the proliferation of the Web, there is a great potential in using this technology for electronic commerce. VINN is an automobile e-commerce platform that advances the vehicle buying industry by utilizing artificial intelligence and machine learning techniques to empower both dealerships and customers to advertise, sell, search, find and purchase their ideal vehicle online. The company's main goal is to provide buyers with a service that meets their lifestyle, hence the visual aspects of vehicles are of prime importance. VINN has been collecting vehicle images for the past years. The data consists of two categories: an image set containing different types of vehicle images (exterior, interior, engine, wheel, etc.) from various perspectives (front, rear, back, etc.), and a database consisting of vehicle information, such as make, model, trim, exterior color, interior color, etc., as entered by the sellers. However, the entered information is often missing or inconsistent with the vehicle's images. Amongst the many vehicle attributes, color is one of the primary factors considered by car buyers. Therefore, there are many advantages of being able to recognize a vehicle's color, including:

- Dataset cleaning: to resolve incomplete and dirty data issues.
- Ground truthing the image sets: to tackle the inconsistency between the images and the description of the vehicle as entered by the sellers or dealers.

- Auto filling the color field in the database: to fill the color data automatically when sellers or dealers upload the images of their vehicle, for a more streamlined C2C experience.

Image processing techniques and machine learning algorithms have become the most useful tools to achieve automation in color recognition [5] [6]. In this work, a hybrid approach is used to classify a vehicle's color from its images into 14 colors: Red, Black, Blue, Silver, White, Grey, Purple, Brown, Yellow, Orange, Green, Pink, Beige and Gold. Our method utilizes pre-trained models to localize the main vehicle in an image, then extracts the dominant color of the selected region of interest (ROI) by detecting the most frequent Hue, Saturation and Value component (HSV), and constructing feature vectors as input to various models to classify the color. We also experimented with relative local entropy as a measure of image segmentation. The segment with local relative entropy lower than a threshold is identified as the ROI, and its most frequent HSV component is extracted as dominant color.

However, due to the similarity between some of the colors (e.g., silver and white), the variations of vehicle colors, and the impact of environmental factors including lighting and reflections, it is difficult to identify the color precisely. It is a challenge to develop a robust and effective system for vision-based color recognition [7].

The rest of this report is organized as follows: related works of vehicle color recognition techniques and image segmentation from an entropy-based perspective are presented in section 2. Section 3 describes VINN's dataset and image set. The proposed methodology is detailed in section 4. Section 5 presents the experimental results of the models introduced in section 4. Lastly, in section 6, we conclude and propose future works.

Chapter 2: Related Works

In this section, we discuss prior work related to vehicle color recognition and image segmentation techniques with a focus on entropy as a measure of thresholding for ROI selection. A study conducted by Dule et al. [8] analyzes the performance of three classification methods (K-Nearest Neighbors, Artificial Neural Networks, and Support Vector Machines) using all possible combinations of sixteen color space components as feature sets on two ROIs (smooth hood piece and semi front vehicle). They use a Plate Recognition System (PRS) to retrieve plate position parameters. Feature are selected in the determined ROIs by three methods, histogram-based feature selection, pixel-based majority selection and pixel-based median selection. A vehicle is classified into one of the seven colors: black, grey, white, red, green, blue, and yellow.

Chen et al. [2] provide a BoW-based method to select the ROI for color recognition while focusing on vehicle localization. The BoW representation as introduced in [2] utilizes a large codebook quantized from color features to map these features into a higher dimensional subspace. Pre-processing using the haze removal method [9] and color contrast normalization method [10] is carried out to overcome image quality degradation. The ROI selection is performed by partitioning the vehicle image into subregions carrying different weights. A classifier is trained to learn the assigned weights. To deal with the multiclass issue, they train the classifier by a linear SVM to increase the efficiency and precision. Nonlinear SVM, though outperforms the linear one, is not used, due to its longer training time [2] [11]. The color of a vehicle is classified into eight classes: black, white, blue, yellow, green, red, grey, and cyan. They apply their models to two datasets which they built, a vehicle image dataset (15,601 vehicle images with half for training and half for testing) and a vehicle video dataset.

Rachmadi and Purnama [4] convert an input image to different color spaces: RGB, CIE Lab, CIE XYZ, and HSV, and input their dataset into a convolutional neural network (CNN) architecture to classify the vehicle color as one of the 8 classes: black, blue, cyan, grey, green, red, white, and yellow. Their CNN architecture consists of 2 base networks of 8 layers each giving a total of 16 layers including two convolutional layers with ReLU as the activation function. The base networks were followed by normalization and max pooling, two convolutional layers without pooling and normalization process, and one more convolutional layer with only pooling and no normalization. They follow the procedure introduced by Krizhevsky et al. [12] where after a number of iterations, the learning rate is decreased by a factor of 10. They test their approach on the image dataset provided by Chen in [2].

Shaded colors, such as navy blue are often misclassified as black. A concept proposed in [13] has vehicles with different chromatic attributes to be trained separately. Their method is comprised of multiple steps. First, each foreground vehicle is extracted from its background. Then, color correction is performed to reduce lighting effects. More precisely, a mapping function is built to minimize the color distortions; using a background image and two video frame images of the car. Next, window parts of the car image is removed and the vehicle color is classified using the lower parts of the image like bumper and doors, making vehicle pixels of the same color more apparent. Finally, a tree-based classifier labels the vehicle into grey and non-grey subgroups, which is then followed by a detailed grey classifier identifying black, silver or white. In addition, a detailed color classifier for red, green, blue, and yellow, is designed for classifying vehicles into chromatic and nonchromatic classes. They have experimented with the NTOU Vehicle Dataset which contains 3,373 vehicle images as the training set and 16,648 vehicle images as the test set.

In [14], a deep-learning-based algorithm (a CNN architecture proposed by Krizhevsky et al. [12]) is adopted as the feature extractor, that computes a feature vector for each image. A linear SVM instead of a fully connected artificial neural network is employed as a color classifier. The spatial pyramid strategy is combined with the original convolutional neural network architecture to improve recognition accuracy [14]. The features are learnt from training data automatically, instead of adopting manually designed features. To assess the proposed process, Hu et al. use the vehicle color dataset provided by Chen [2] with nine-group splits of training and test. Their results show that the ratio of training data does not influence the final recognition accuracy for their proposed feature.

Kim et al. in [15] use the HSI (Hue Saturation Intensity) space to represent the color feature of the image by a histogram (from the normalized H, S and I component). Using a distance function to compute the similarity between a feature vector and templates, the color of a vehicle is classified into one of seven colors: black, silver, white, red, yellow, green, and blue. They use 700 images for their experiment, where there are 100 images for each color. Half of the images are used for templates, and the remaining 350 images are used for the test.

Chowdhury et al. [16] advocate Image segmentation by splitting the image into groups of homogeneous pixels based on ROI, as a universal step for image recognition. Many researchers favor the use of image segmentation to extract the most significant regions that contain the desired characteristics of the image, such as the dominant color. The threshold and seed-point selections are two parameters that have a major impact on the effectiveness of image segmentation [17].

There are many threshold selection methods such as basic global thresholding, clustering, region growing and entropy based.

In information theory, the entropy of a random variable is the expected amount of "information", or "uncertainty" inherent to the variable's possible outcomes. Given a discrete random variable X, with possible outcomes x1, x2,…, xn which occur with probability P(x1), P(x2),…, P(xn), the entropy of X is defined as:

$$H(x) = -\sum_{i=1}^{n} P(x\,i)\; log\; P(xi)$$

Another useful measure of entropy that assesses the difference between two distributions and works equally well in the discrete and the continuous case is the relative entropy of a distribution, the Kullback–Leibler divergence, which is extensively used in Entropy-based image processing studies [19]. one of the common approaches in this area is entropy based object detection using object color which tries to develop an object classifier by defining the bounding limits of the regions corresponding to a color belonging to the object [18].

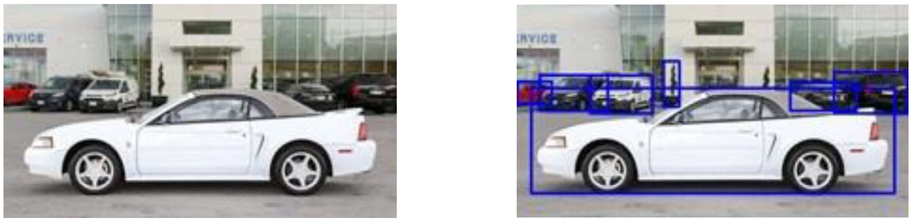
Another approach, that is one of the main approaches in using Entropy of the image, is Entropy-based image segmentation which uses entropy to measure how uniform the pixel characteristic is in order to partition an image into separate regions, which ideally correspond to different real-world objects [18][19].

In [17], different entropy measures have been applied on both greyscale and color images. The entropy of an image is computed with a four-step algorithm in a plot of entropy versus grey level. This plot gives the minima points and the lowest minima can be employed as a threshold value. To reflect the contextual information between pixels, a thresholding method is proposed that constructs a two-dimensional histogram of a pixel's brightness and local relative entropy of its neighboring region [20]. The local relative entropy (LRE) measures the brightness difference between a pixel and its neighboring pixels. In [16], a multilevel thresholding image segmentation method is proposed based on the minimization of bi-entropy function. Then both the Shannon's entropy and the proposed method are employed in image segmentation. Their method considers both the spatial information and the grey information to reduce computation complexity. Pauzi in [21] introduces a real-time instrumentation system for visually impaired people to help them recognize color. Their color classification system uses an entropy algorithm based on the HSV and RGB color spaces, and labels a color into one of the ten colors: black, brown, cyan, red, orange, yellow, green, blue, magenta, grey and white. The results show that the HSV model classifies more accurately than the RGB one.

Chapter 3: VINN's Dataset

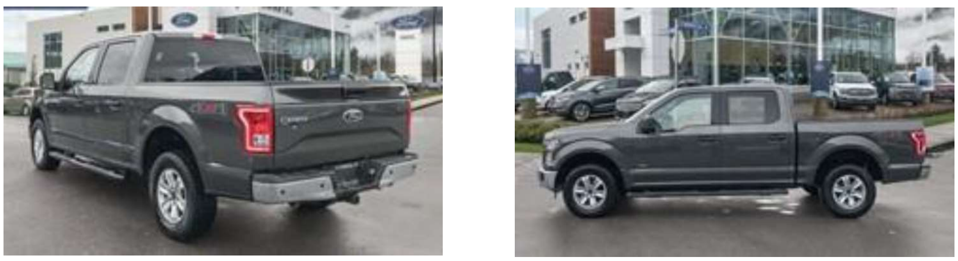
VINN's dataset has more than 111,250 vehicle images. There are 35 features associated with each vehicle. For each vehicle there are between 1 to 30 unlabeled images of various parts such as interior, exterior, engine, wheels, lights, etc., and in different perspectives including front, back, rear, left-rear, and right-rear. As the goal of this project is exterior color recognition, there is a need to label the images as exterior for the training, validation, and test sets. Labelling has been done manually using a GUI (python application with Flask hosted on AWS) which displays an image and labels it based on user's key press to exterior, interior, not a car, engine, open trunk, etc. Finally, 2,530 images labelled as exterior are used as samples. To ensure a valid dataset, all these images have been checked manually for consistency between their color in the image and the corresponding 'exterior color' field in the dataset. The 14 color labels are Red, Black, Blue, Silver, White, Grey, Purple, Brown, Yellow, Orange, Green, Pink, Beige and Gold.

There exist other challenges in the dataset. The majority of the images are captured by smartphones in various resolutions at a close range mostly in a parking lot area. Environmental conditions often affect the quality of the images. Some images are impacted by sun light, shadows, and reflections of other objects, such as trees and people, in the area. Furthermore, in most of the images there are many other parked vehicles which necessitates a pre-processing step to differentiate and identify the target vehicle in the image as Figure 1 illustrates.



(a) Sample Image (b) Detected vehicles with their bounding boxes

Figure 1. A sample image without (a) and with (b) vehicles detected



(a) Rear-left view (b) Side view

(c) Rear view

Figure 2. Sample images from different views

Another big challenge is that for each vehicle, there are several exterior shots from different views as shown in Figure 2. Hence, a post-processing step is required to conclude the exact color of each vehicle based on the recognized colors, which may be different from all its images. Moreover, there are seven body types in the dataset including SUVs, trucks, sedans, hatchbacks, convertibles, coupes and vans, which may complicate the color recognition process. Obviously, similar to most other data-related projects, the lack of labelled data instances, in our case, color labelled images, is a major challenge in developing effective and efficient models. We overcome this lack of sufficient samples by increasing the number of images, by applying image augmentation technique to our image dataset.

Chapter 4: Proposed Color Recognition Method

This section contains the details of the process carried out and the models applied to our image set for color recognition. The results of alternative solutions used in each step are compared. A flowchart of the proposed method is illustrated in Figure 3.

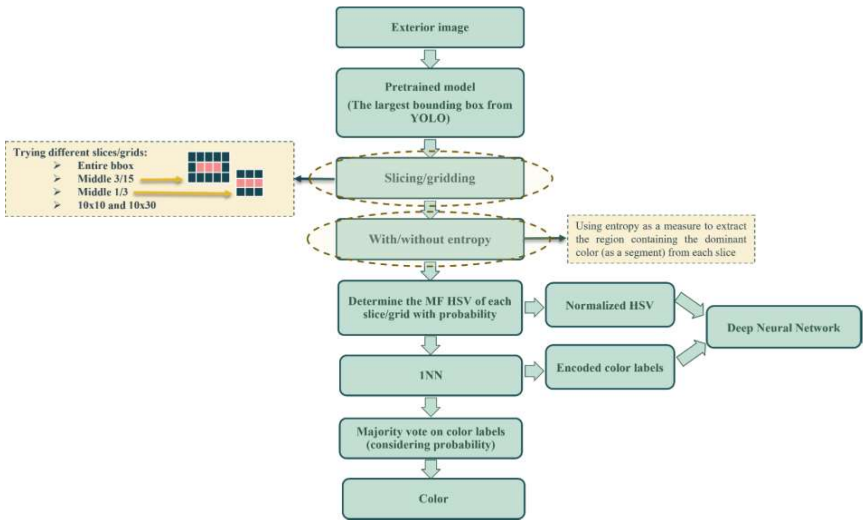


Figure 3. Flowchart of the proposed method

4.1 Pre-processing

The majority of the images includes more than one vehicle, therefore we need to identify the main vehicle of interest in each image. We used YOLO v4 [22] as a pre-trained model [23] to localize all the vehicles in an image. After obtaining their coordinates, we computed the area of each detected vehicle in the image. We then selected the vehicle with the largest area as the main vehicle as shown in Figure 1(b). Also, to ensure that the pre-trained model was working effectively in finding and localizing the main vehicles, and more importantly to set its appropriate confidence level, we checked all the output images visually. Since the RGB color space is not robust enough when images are captured in complex natural scenes, as artifacts of low illumination, strong lighting, and camera color bias, etc., a further pre-processing step is needed to convert an image to HSV color space for color recognition.

4.2 ROI selection

We tested our method on different regions of an image. We first removed the top and bottom third of the bounding box of the vehicle as these regions contain windows and wheels, respectively. Then we sliced the remaining region into 5 horizontal subregions and selected the middle three slices for dominant color extraction. Another experiment worked with the entire middle third of the bounding box without any slicing. Subsequent experiments used the entire bounding box without removing the top and bottom subregions, while using different configurations to grid this region. These include 10x10 and 10x30 grids. We also tested the 10x10 and 10x30 grids on the middle-third of the bounding box to see how the model performed. Furthermore, some of these experiments have been applied on the regions with and without entropy-based image segmentation. Thus, for each region, the grey level local relative entropy (LRE) method proposed in [20] is utilized to calculate the entropy of each pixel by using the brightness value of all the pixels and the mean grey level value of the pixels in its N×N neighborhood. Let $I(x, y)(x=1, 2, \ldots, M; y=1, 2, \ldots, N)$ be the brightness of a pixel located at (x, y) in the image I. $I(x, y) \in \{0, 1, \ldots, L-1\}$. As shown in equation (1), local relative entropy (LRE) of each pixel (x, y) in its N×N neighborhood is computed as:

$$LRE = \sum_{i=-(n-1)/2}^{(n-1)/2} \sum_{j=-(n-1)/2}^{(n-1)/2} I(x+i, y+j) \times \left| log \frac{I(x+i,y+j)}{\tilde{I}(x,y)} \right| \qquad (1)$$

where $\tilde{I}(x,y)$,), which is the mean gray level value as explained in the [20], is calculated from equation (2):

$$\tilde{I}(x,y) = \frac{1}{n^2} \sum_{i=-(n-1)/2}^{(n-1)/2} \sum_{j=-(n-1)/2}^{(n-1)/2} I(x+i, y+j) \qquad (2)$$

The LRE is then normalized to a number between 0 and L-1 (L is the highest brightness value in that neighborhood).

We tested both N=10 and N=20 as the number of adjacent pixels. A sample image with its output based on this approach is shown in Figure 4.

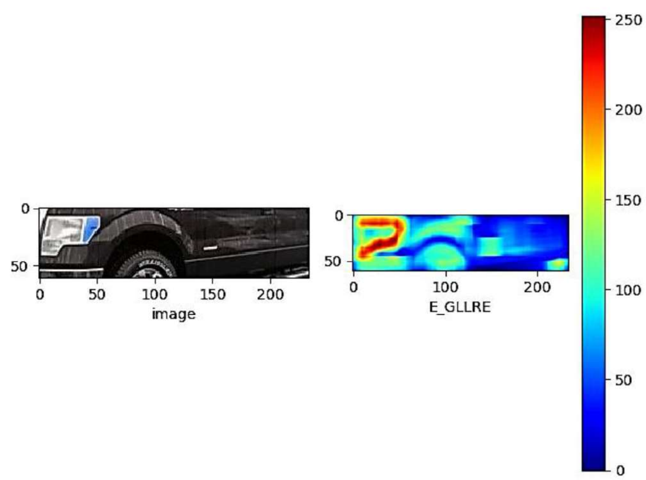


Figure 4. Sample output of the grey level local relative entropy

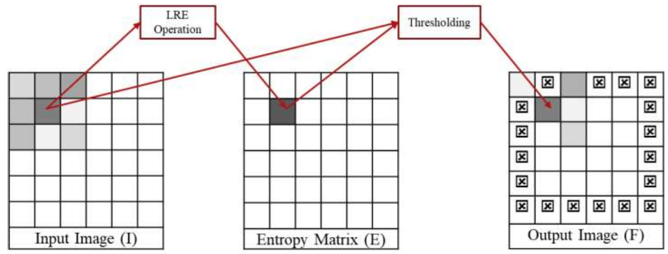


Figure 5. The principle of LRE and thresholding process. To calculate a pixel in the output image, a pixel from the input image and its neighbors are processed.

Next, a thresholding step was performed on the pixel-wise entropy matrix (E) to generate a binary mask. Threshold selection techniques can be roughly classified as global, local and adaptive according to the nature of the algorithms [24]. Local thresholding algorithms select the threshold based on local properties in the histogram function such as the existence of maxima and minima. Global thresholding algorithms attempt to measure some global statistics of the histogram as the criteria for the selection. And adaptive thresholding calculates a threshold value for each pixel in the image. We considered the mean entropy as a global threshold value [25], to distinguish the color of different parts of the vehicle. We care more about the pixels with entropy lower than mean entropy, because these pixels and their neighbor pixels highly probably belong to the same class. We constructed a binarized matrix with the same shape of one channel of the image as the output that is presented in equation 3:

$$Mask(x,y)=\begin{cases}0 \; if \; E(x,y) \geq t \\ 1 \; if \; E(x,y) < t\end{cases} \quad (3)$$

Where value is 1, the pixel in the coordinate of (x,y) in all three channels (in the original color image) is considered in the region for color recognition. Figure 5 shows this process where cross marks show that the corresponding pixel will not be contributed to the next step.

Any output region from the previous step has its most frequent HSV component extracted with probability calculated as (4):

$$Probability = \frac{number\ of\ pixels\ of\ MF\ HSV}{number\ of\ pixels\ in\ region} \qquad (4)$$

This probability was used as a measure in the majority voting on slice colors and then on vehicle images, for the cases when there are equal votes on slice or image color labels. We employed three models in this step to recognize color. The first model used normalized HSV vectors of all the slices with probabilities as a feature vector, to feed into a deep neural network. As an example, for a 10x30 grid image, we have 300 normalized H, S and V numbers and 300 probability numbers, resulting in a total of 1200 features. The second and third models used the assigned color label from each slice's MF HSV. This color label assignment is described below.

4.3 Color Label Assignment

For color label assignment, we conducted a multi-stage approach in mapping the MF HSV to the color classes. First, we did a thresholding on the value component to differentiate black, and then another thresholding on saturation to distinguish white, grey and silver from other colors. To differentiate these three colors, we checked the hue to determine what color it was. At the next stage, we used a 1NN classifier with centroids referring to the other eight colors. To define the centroids and ranges of H, S and V components for each color, we utilized an HSV color thresholder script to determine the lower and upper bounds of each color using sliders. We also used a color wheel named Martian color [26] to fine tune the ranges and centroids. We combined the red and brown as a single color (red-brown) in 1NN as they have some overlap in hue. Similarly, we combined orange and brown as (orange-brown) in 1NN. If these two new colors were selected from the 1NN, then we checked their value to decide if it was brown, red or orange. As the hue is circular, any two pixels with hue 175 and 5 should have the same distance as from a pixel with hue 0[1]. As a result, we had to change the distance function in 1NN to (5).

$$Hue\_Distance = 90 - ||Hue - Centre| - 90| \tag{5}$$

The output of this step was used as the input to the two other models. The first model performed a majority vote on the slice colors of each image and used the probability in case of equal votes.

The second model encoded the color labels to 14 bits using a one-hot encoder, while slice probabilities were concatenated as well to construct the feature vector. For example, for a 10x10 grid image, a feature vector of size 1500 is constructed. Then, this feature vector was used as the input to a deep neural network for color classification. After recognizing the color of each image, a majority vote using the image color labels and their probabilities from the images of the same vehicle, was carried out to determine the vehicle's color.

---

[1] Hue is scaled in the range of 0 to 179 for all the images. Even if the hue is in the range of 0 to 359, two pixels with hue 350 and 10 should have the same distance from a pixel with hue 0.

4.4 Feature-representation-transfer learning

To compare the results of this method, we applied transfer learning as one of the other deep learning network based techniques. Transfer learning is a machine learning method relying on transferring or reusing whole or partial of the knowledge gained from a developed model in a new and different but related problem [27]. Computation power and time are two main resources that are required for training neural networks. This is the reason for the popularity of transfer learning approach, it enables the researchers to apply pre-trained models on their datasets in order to speed up training and improve the performance of their deep learning model. One of the different approaches to transfer learning, based on what to be transferred, is called Feature-representation-transfer that implies on encoding the knowledge into learned feature representation to reduce difference between the source and the target domain and the error of classification and regression models [28]. Using this method, we extracted the features from a pre-trained model, VGG19, as the input to our DNN as this is a common type of transfer learning in the field of deep learning. VGG models are powerful and useful in both image classification problems and as the basis for new models that use image inputs as they generalize well to other datasets [29]. They are trained on ImageNet dataset that contains 1000 categories and 1.2 million images. After localization of the main vehicle, we fed it as an input to VGG model, keeping whole network architecture and weights and removing the last three fully connected layers. The extracted feature vector then inputted into the DNN used in proposed method. This means that the feature extraction step in our proposed method is replaced by the knowledge gained from the pre-trained model which is VGG19 here to compare the performance of the models.

Chapter 5: Experimental Results

The performance of the methods was evaluated by the precision of all color categories, with the micro average precision of all categories as the average precision (AP). In the feature learning stage, we constructed different feature vectors based on the number of grids selected from the images. To train the deep neural network, we used five dense layers with ReLU as an activation function and one last fully connected layer with a SoftMax activation function. The number of neurons in the last layer was equal to the number of colors, 14 in our case. The prediction output was a 14-length vector which we used to decode and mapped to the 14 color labels. To avoid overfitting, we applied dropout and l2 regularization as suggested in [12] since this is the most commonly used method to reduce the effect of overfitting in training deep learning networks. We used 80% of our images for training, 10% for validation and 10% for testing. Each class consisted of different number of instances from 12 to 435. To tackle the imbalanced dataset issue, we applied the Synthetic Minority Oversampling Technique after our feature training step. The proposed techniques were implemented in Python on a Windows 10x64-based system. The experiments were carried out on a desktop computer with an Intel(R) Xeon(R) E-2176M CPU 2.71 GHz with 16-GB memory.

We summarized the results of our approaches in Table 1[1], showing each method with the configuration of the slice, the number of grids, and how they performed. As can be seen in this table, performance increases with the slicing of the vehicle

---

[1] For the last two rows of experiments, the entire vehicle bounding box was gridded into 10x10 and 10x30.

image and the gridding of more subregions. Results using entropy show an improvement in comparison to non-entropy based experiments on the same region and configuration in the voting model. It may be helpful to experiment with additional configurations of the models, while considering entropy before the MF HSV extraction and color label assignment step. The only issue with entropy is the relatively long computation time which is a critical factor in many real-time applications. This issue could be resolved if the models were running on appropriate hardware configuration as confirmed in [4]. The precisions of each color category from the best results obtained (in our proposed method), the corresponding confusion matrix and learning curve are presented in Table 2, Figures 6 and 7, respectively. Furthermore, the same results achieved from transfer learning model are provided in Table 3, Figures 8 and 9. Comparing the achieved results with VGG19 feature representation based results show that average accuracy are close together. Although for some colors the VGG based model outperform up to 21 percent and for some others it fails to receive the same results comparing to our model and there is 20 percent decrease in the precision metric for those colors. It can be concluded that our method which relies on feature extraction using entropy approach works well enough in comparison to VGG19 feature vector representation.

Table 1. Average recognition performance of our various models

		Method					
		Voting on color labels		DNN on normalized HSV		DNN on encoded color labels	
		Without entropy	With entropy	Without entropy	With entropy	Without entropy	With entropy
Configu	Entire vehicle bounding box	31%	39%	39%	62%	29%	64%

	Middle 3/15	53%	62%	57%	66%	55%	67%
	Middle 1/3	46%	54%	49%	58%	53%	65%
	10 × 10 grid	55%	61%	59%	65%	56%	72%
	10 × 30 grid	42%	63%	60%	68%	57%	71%

We also investigated the misclassified images to identify the deficiency of the proposed model. More than 15% of the grey images which are misclassified as black, have a very dark shade close to black. Therefore, the color assignment labeler may have incorrectly classified their slices as black. The same case appears in some other class samples which are misclassified as blue. Reviewing the samples show that these images have more bluish tones in their regions. This could be due to the reflection of the sky on the vehicle bodies. A more accurate and flexible ROI selection process would be helpful in solving this issue. Another important issue is that we have some misclassified images from various colors to silver. This may have happened because of the very bright sunlight reflection on metallic paint or the color is too light so it becomes very close to another color. This indicates the need for a color correction step. As previously mentioned, one challenge is having different images of the same vehicle from different viewpoints, which affects the model performance; however, performance seems to be better with a cleaner image set and an orientation model which differentiates the viewpoints.

Chapter 6: Conclusion and Future Work

In this report, we present a color recognition technique for vehicle exterior images. We classify the vehicle color into 14 classes. Our proposal is based on the HSV version of the images. After some pre-processing to detect the target vehicle in the image and its RoI using mean local relative entropy of pixels as threshold value combining with the slicing and gridding of the RoI, followed by extracting the most frequent HSV component. Three models are tested with various feature vectors as input. We assign a color label using 1NN to each slice of the MF HSV component in the first model, and determine the color based on a majority vote. The second and third models use the encoded color label and normalized HSV component, respectively, concatenated to the color probability of each grid as feature vectors to a deep neural network. Although experimental results show 72% of successful classification on average, the precision is low for some color classes. We also extracted feature vector from VGG19 to compare our model results. The causes of misclassification are discussed, with enhancements as future work presented.

Table 2. Precision metrics of each color category from our best results.

	Precision	Recall	F1-score	Support
Red	0.73	0.83	0.78	42
Black	0.82	0.80	0.81	79
Yellow	0.63	0.50	0.56	24
Grey	0.67	0.79	0.72	58
White	0.83	0.90	0.86	78
Silver	0.70	0.66	0.68	29
Blue	0.72	0.84	0.77	49
Gold	0.50	0.58	0.54	19

Brown	0.59	0.48	0.53	21
Orange	0.67	0.78	0.72	23
Purple	0.60	0.33	0.43	27
Green	0.79	0.60	0.68	25
Beige	0.56	0.38	0.45	13
Pink	0.75	0.63	0.69	19
Total				506
Accuracy			0.72	

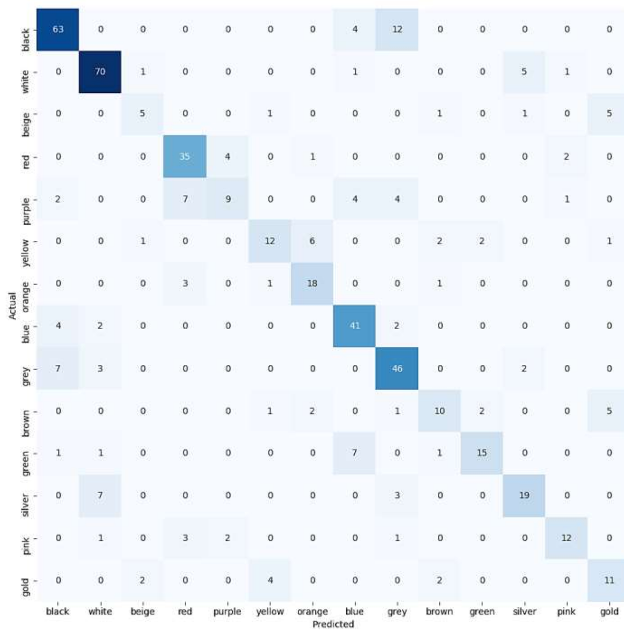


Figure 6. Confusion matrix from our best results

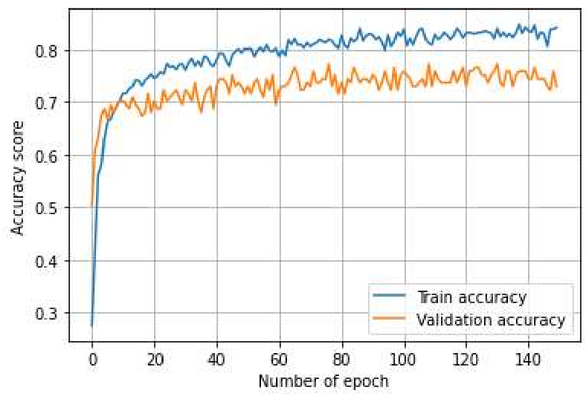


Figure 7. Learning curve from our best result

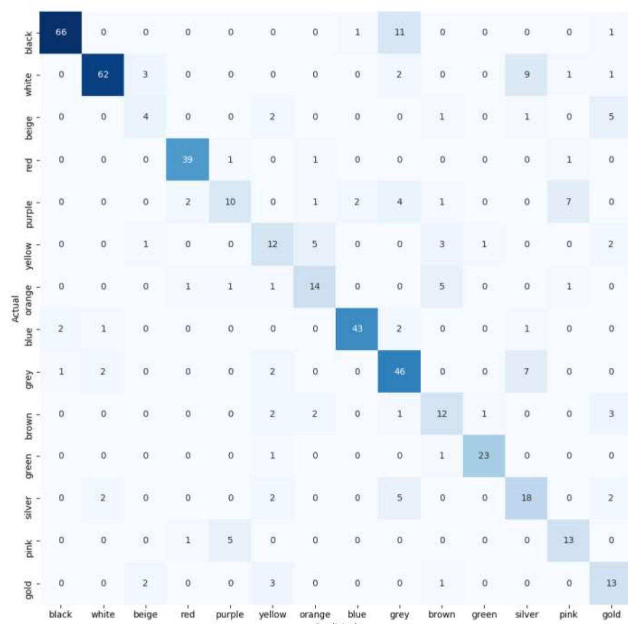


Figure 8. Confusion matrix from transfer learning model

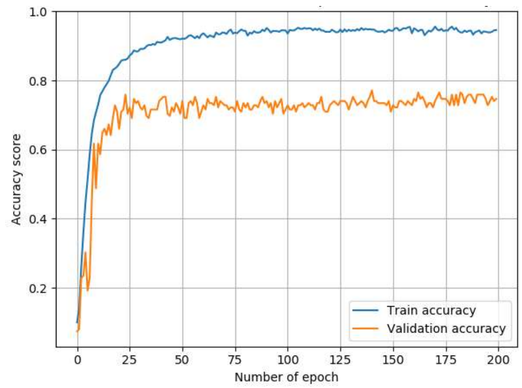


Figure 9. Learning curve from transfer learning model

Table 3. Precision metrics of each color category from transfer learning model

	Precision	Recall	F1-score	Support
Red	0.91	0.93	0.92	42
Black	0.96	0.84	0.89	79
Yellow	0.48	0.50	0.49	24
Grey	0.65	0.79	0.71	58
White	0.93	0.79	0.86	78
Silver	0.50	0.62	0.55	29
Blue	0.93	0.88	0.91	49
Gold	0.48	0.68	0.57	19
Brown	0.50	0.57	0.53	21
Orange	0.61	0.61	0.61	23
Purple	0.59	0.37	0.45	27
Green	0.92	0.92	0.92	25
Beige	0.40	0.31	0.35	13
Pink	0.57	0.68	0.62	19
Total				506
Accuracy			0.74	

References

1. K. S. Aarathi and A. Abraham, “Vehicle color recognition using deep learning for hazy images,” Proceedings of the International Conference on Inventive Communication and Computational Technologies, ICICCT 2017, no. Icicct, pp. 335–339, 2017, doi: 10.1109/ICICCT.2017.7975215.

2. P. Chen, X. Bai, and W. Liu, “Vehicle color recognition on urban road by feature context,” IEEE Transactions on Intelligent Transportation Systems, vol. 15, no. 5, pp. 2340–2346, 2014, doi: 10.1109/TITS.2014.2308897.

3. X. Li, G. Zhang, J. Fang, J. Wu, and Z. Cui, “Vehicle color recognition using vector matching of template,” 3rd International Symposium on Electronic Commerce and Security, ISECS 2010, no. c, pp. 189–193, 2010, doi: 10.1109/ISECS.2010.50.

4. R. F. Rachmadi and I. K. E. Purnama, “Vehicle Color Recognition using Convolutional Neural Network,” pp. 2–6, 2015, [Online]. Available: http://arxiv.org/abs/1510.07391

5. S. R. B. R, “Classification of Vehicles using Image Processing Techniques,” International Journal of Recent Trends in Engineering and Research, vol. 4, no. 3, pp. 393–401, 2018, doi: 10.23883/ijrter.2018.4144.qcmco.

6. D. S. B. Tilakaratna, U. Watchareeruetai, S. Siddhichai, and N. Natcharapinchai, “Image analysis algorithms for vehicle color recognition,” 2017 International Electrical Engineering Congress, iEECON 2017, pp. 4–7, 2017, doi: 10.1109/IEECON.2017.807588.

7. L. W. Tsai, J. W. Hsieh, and K. C. Fan, “Vehicle detection using normalized color and edge map,” IEEE Transactions on Image Processing, vol. 16, no. 3, pp. 850–864, 2007, doi: 10.1109/TIP.2007.891147.

8. E. Dule, M. Gökmen, and M. S. Beratoğlu, “A convenient feature vector construction for vehicle color recognition,” Proc. of the 11th WSEAS Int. Conf. on Neural Networks, NN ’10,

Proceedings of the 11th WSEAS Int. Conf. on Evolutionary Computing, EC '10, Proc. of the 11th WSEAS Int. Conf. on Fuzzy Systems, FS '10, pp. 250–255, 2010.

9. H. Kaiming, S. Jian, and T. Xiaoou, "Single image haze removal using dark channel prior. Single image haze removal using dark channel prior.," Cvpr, vol. 33, no. 12, pp. 2341–2353, 2009, [Online]. Available: http://kaiminghe.com/publications/cvpr09.pdf

10. R. C. Gonzalez et al., "Digital Image Processing Using MATLAB ® Second Edition Tata McGraw Hill Education Private Limited NEW DELHI," 2010.

11. R. E. Fan, K. W. Chang, C. J. Hsieh, X. R. Wang, and C. J. Lin, "LIBLINEAR: A library for large linear classification," Journal of Machine Learning Research, vol. 9, no. 2008, pp. 1871–1874, 2008, doi: 10.1145/1390681.1442794.

12. A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," Communications of the ACM, vol. 60, pp. 84–90, 2012.

13. J. W. Hsieh, L. C. Chen, S. Y. Chen, D. Y. Chen, S. Alghyaline, and H. F. Chiang, "Vehicle color classification under different lighting conditions through color correction," *IEEE Sensors Journal*, vol. 15, no. 2. pp. 971–983, 2015. doi: 10.1109/JSEN.2014.2358079.

14. C. Hu, X. Bai, L. Qi, P. Chen, G. Xue, and L. Mei, "Vehicle Color Recognition With Spatial Pyramid Deep Learning," IEEE Transactions on Intelligent Transportation Systems, vol. 16, no. 5, pp. 2925–2934, 2015, doi: 10.1109/TITS.2015.2430892.

15. K. J. Kim, S. M. Park, and Y. J. Choi, "Deciding the number of color histogram bins for vehicle color recognition," Proceedings of the 3rd IEEE Asia-Pacific Services Computing Conference, APSCC 2008, pp. 134–138, 2008, doi: 10.1109/APSCC.2008.207.

16. K. Chowdhury, D. Chaudhuri, and A. K. Pal, "A new image segmentation technique using bi-entropy function minimization," Multimedia Tools and Applications, vol. 77, no. 16, pp. 20889–20915, 2018, doi: 10.1007/s11042-017-5429-8.

17. H. Sen and A. Agarwal, “A comparative analysis of entropy based segmentation with Otsu method for gray and color images,” Proceedings of the International Conference on Electronics, Communication and Aerospace Technology, ICECA 2017, vol. 2017-Janua, pp. 113–118, 2017, doi: 10.1109/ICECA.2017.8203655.

18. F. A. Pujol, M. Pujol, A. Jimeno-Morenilla, and M. J. Pujol, “Face detection based on skin color segmentation using fuzzy entropy,” Entropy, vol. 19, no. 1, pp. 1–22, 2017, doi: 10.3390/e19010026.

19. H. Zhang, J. E. Fritts, and S. A. Goldman, “An Entropy-based Objective Evaluation Method for Image Segmentation,” no. 1.

20. W. Yang, L. Cai, and F. Wu, “Image segmentation based on gray level and local relative entropy two dimensional histogram,” PLoS ONE, vol. 15, no. 3, pp. 1–9, 2020, doi: 10.1371/journal.pone.0229651.

21. G. A. Pauzi, “Colour Classification Using Entropy Algorithm in Real Time Colour Recognition System for Blindness People,” KnE Engineering, vol. 1, no. 1, pp. 0–5, 2016, doi: 10.18502/keg.v0i0.485.

22. A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, “YOLOv4: Optimal Speed and Accuracy of Object Detection,” CoRR, vol. abs/2004.1, 2020, [Online]. Available: https://arxiv.org/abs/2004.10934

23. A. Ponnusamy, “cvlib-high level Computer Vision library for Python.” 2018.

24. N. Padmasini, R. Umamaheswari, and M. Y. Sikkandar, “Chapter 10 - State-of-the-Art of Level-Set Methods in Segmentation and Registration of Spectral Domain Optical Coherence Tomographic Retinal Images,” in *Soft Computing Based Medical Image Analysis*, N. Dey, A. S. Ashour, F. Shi, and V. E. Balas, Eds. Academic Press, 2018, pp. 163–181. doi: https://doi.org/10.1016/B978-0-12-813087-2.00009-9.

25. D. H. AlSaeed, A. El-Zaart, and A. Bouridane, “Minimum cross entropy thresholding using entropy-Li based on Log-normal distribution for skin cancer images,” Proceedings - 7th

International Conference on Signal Image Technology and Internet-Based Systems, SITIS 2011, pp. 426–430, 2011, doi: 10.1109/SITIS.2011.86.

26. "{Martian Colour Wheel.} {Why do we need another colour wheel?}."

27. J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?," Advances in Neural Information Processing Systems, vol. 4, no. January, pp. 3320–3328, 2014.

28. S. Panigrahi, A. Nanda, and T. Swarnkar, "A Survey on Transfer Learning," Smart Innovation, Systems and Technologies, vol. 194, pp. 781–789, 2021, doi: 10.1007/978-981-15-5971-6_83.

29. K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings, pp. 1–14, 2015.