Algotive Challenge Report

Problem: You have a dataset of car images, taken from surveillance cameras, you need to develop a system that generates groups of similar images (you are free to decide which feature of the cars you will use). The system must automatically generate groups, where the elements inside each group share the feature selected.

Introduction:

Color identification in vehicles has multiple practical applications, particularly given the increasing use of surveillance cameras, which can be crucial even in detecting criminal activities. However, a significant challenge is the detection of the Region of Interest (ROI) in a vehicle to focus on its color, avoiding the noise from other elements in the image.

In this challenge, a 'specular-free' algorithm is implemented to differentiate between chromatic and non-chromatic areas, focus to preserve the saturation of color values in all pixels. Additionally, a threshold with horizontal projection is used to refine the object's identification. Finally, the size of a region containing the most relevant information is defined, allowing the extraction of its features through a Residual Network (ResNet 50). This data is then used to train a clustering algorithm that classifies colors into different groups.

Steps:

In the specular-free method, chromatic and non-chromatic areas are distinguished in images, keeping the saturation of pixels constant while retaining their hues. This is achieved by the equation below:

$$I_{r,g,b}^{spec} = I_{r,g,b} - \frac{\sum_{r,g,b} I_i - \frac{\bar{\mathrm{I}}(3\check{\mathrm{C}} - 1)}{\check{\mathrm{C}}(3\bar{\mathrm{I}} - 1)}}{3}$$

where $\bar{\mathrm{I}} = \max(I_r, I_g, I_b)$, $\check{\mathrm{C}} = \frac{\bar{\mathrm{I}}}{I_r + I_g + I_b}$ and $[I_r, I_g, I_b]$ denote the RGB values of a pixel. In this way is possible processes the RGB values of each pixel and applies a pre-set threshold (0.4) to identify chromatic pixels. This approach effectively highlights the colour of vehicles, facilitating better color identification in surveillance and reconnaissance applications (Zhang et al., 2017).

Then it was implemented a three-step process for the efficient extraction of regions of interest (ROI) in images. Initially, we apply a thresholded segmentation, using the horizontal projection and preserving colour. This technique involves calculating the sum of the colour channels per row, setting a threshold (.15) based on the maximum of these sums and creating a mask to identify the relevant rows. The resulting binary image is then analysed to determine the most informative row, based on the largest sum of pixel values, indicating a concentration of important

features. Finally, a horizontal ROI centred on this row is extracted, using a predefined window size, to isolate the section of the image with the most relevant information for further analysis. Below are some images obtained from this processing:

Image 1

Image 2

Image 3

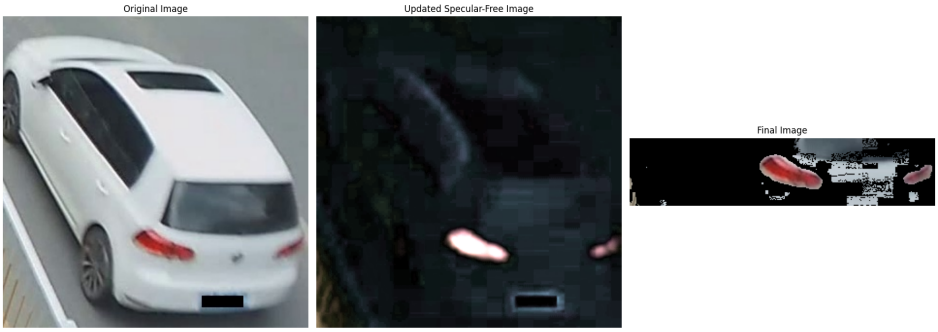

Image 4

As can be seen, the method seems to be robust in most cases, however, it could be observed that it has a lot of difficulty with the white color, this could be due to different cases such as high reflectivity of the white, little difference in saturation, among others, however, it was decided to continue with this method with the understanding that this can add noise.

Then, a deep learning model is used, specifically a variant of the **Convolutional Neural Network ResNet50 (Torch implementation),** known for its effectiveness in computer vision tasks. This network has previously been trained on a vast dataset (Imagenet), allowing it to recognize a wide range of visual features. However, instead of using the network for a standard classification task, we modify its structure to act as a feature extractor. For this, the last two layers average pooling and fully connected were removed. This allows the power of the network to be harnessed to identify and encode the essential visual properties of vehicles in a format that facilitates further analysis.

To efficiently manage and process the data through this model, a data management system is implemented that organizes the images in batches and processes them sequentially. This approach not only optimizes the use of computational resources but also enables the handling of a large volume of data, an essential aspect given the size and complexity of contemporary image datasets.

The result of this process is a set of feature vectors for each image, concisely representing the key visual properties of the vehicles. This will allow us to feed this feature vector for each of the images into a clustering algorithm for unsupervised classification.

For the cluster analysis it was used a Gaussian Mixture Model (GMM) for vehicle color classification is based on its ability to handle variability and overlapping color features. Unlike simpler methods such as K-means, the GMM allows modelling clusters with irregular shapes and probabilistic membership assignments, which is crucial for handling the subtle variations and ambiguities in vehicle colors.

It is important to mention that before applying GMM, it is essential to normalize the extracted features. This normalization standardizes the feature scales, which is crucial for algorithms such as GMM that are sensitive to variation in data scales. Normalization ensures that each feature contributes equally to the analysis, allowing for a more accurate and representative segmentation of color groups in vehicles.

To determine the optimal number of clusters in our GMM model, we use the elbow method. This approach allows us to identify a point where increasing the number of clusters no longer provides significant benefits in explaining the variability of the data, thus providing a balance between accuracy and complexity of the model. Result of this method is in the image below.

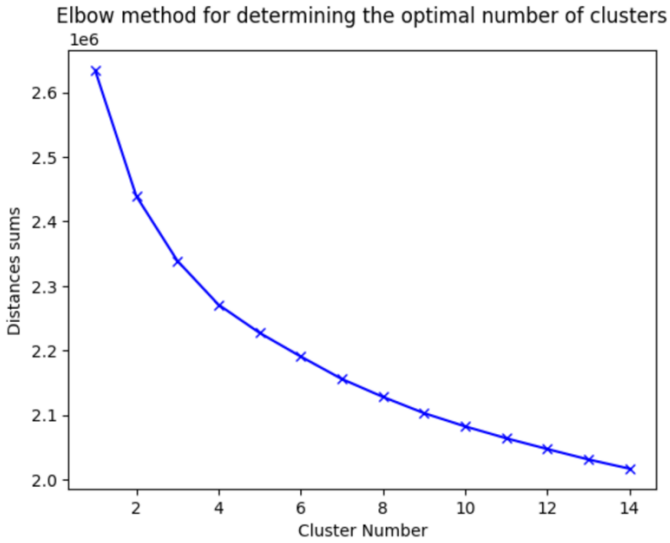

Image 5

Due to the limited time we had for this challenge, we did not use more techniques such as dimension reduction to observe the behavior of the clusters, the validation used was groups of images for each of the clusters, below are some images where you can see that it does seek to group them in a certain way, but sometimes the classification is by color and in some other clusters may also be taking into consideration the type of vehicle.

Image 6

Image 7

Image 8

Image 9

It can be observed that the shape of the vehicle influences the clustering, suggesting an area for improvement in our current approach. A potential strategy to address this issue would be the implementation of an advanced segmenter, such as the Measurement Attention Segmenter (SAM), focused on exclusively isolating the vehicle in the image. This would allow a more accurate concentration on the color of the vehicle, minimizing the influence of other visual factors.

In addition, scanning analyses in different color spaces, such as HSV or LAB, could provide clearer color discrimination, which could be beneficial for improved classification. These modifications have the potential to refine the accuracy of our model in color identification and represent valuable directions for future research and continued model development.

It is important to mention that for each of the models obtained from the steps described above, the models (and scaler) were saved for further performance testing.

Finally, attention was focused on the evaluation and optimization of the model's performance in terms of latency and throughput. This stage is crucial to ensure the viability of the model in real-time applications, where speed and processing efficiency are essential.

Initially, we conducted rigorous tests to measure the latency and throughput of the model in its original form. Latency, which refers to the time it takes for the model to process a single instance of data, and throughput, which indicates the number of instances the model can process per unit time, are critical indicators of model performance in practical scenarios.

After these measurements, we proceeded to convert our model to an ONNX (Open Neural Network Exchange) format. The main objective of this conversion is to improve the efficiency of the model, taking advantage of the interoperability and optimization offered by the ONNX

format. This format is known to improve the portability and performance of deep learning models, allowing their use on a variety of platforms and devices with different hardware capabilities.

Once the model was converted, we performed a new series of tests to measure latency and throughput. The results obtained were compared with previously recorded metrics to assess performance improvements. The results are shown in the table below:

Model Format	Latency Batch 1	Latency Batch 4	Latency Batch 8	Throughput
Torch	0.0819	0.2183	0.4112	10 images/second
ONNX	0.0617	0.1692	0.3242	30 images/second

The substantial increase in performance with ONNX is particularly noteworthy, as it suggests that ONNX optimizations allow the model to use system resources more efficiently, thus processing more images in the same amount of time. This capability is crucial for real-time applications, where large data streams often need to be processed quickly.

In conclusion, the conversion to ONNX from Torch provides considerable improvements in processing efficiency, as evidenced by reduced latencies and increased throughput. These improvements are critical for deployment in scenarios where performance and speed are essential, such as real-time monitoring and autonomous vehicle systems, in addition, if we want to migrate to another format, for example TensorRT, the process is facilitated by having the model already converted to ONNX format.

Although deployment in a production environment was not within the project timeframe, the strategic choice would have been to use NVIDIA Triton Inference Server with TensorRT optimizations. This decision is based on Triton's ability to efficiently handle optimized models on production servers, providing high performance and efficient resource management.

References:

1. Zhang, Q., Li, J., Zhuo, L., Zhang, H., & Li, X. (2017). Vehicle color recognition with vehicle-color saliency detection and dual-orientational dimensionality reduction of CNN deep features. Sensing and Imaging, 18(20). https://doi.org/10.1007/s11220-017-0173-8

2. Gao, Y., & Lee, H. J. (2016). Local tiled deep networks for recognition of vehicle make and model. Sensors, 16(2), 226.

3. Chen, P., Bai, X., & Liu, W. (2014). Vehicle color recognition on urban road by feature context. IEEE Transactions on Intelligent Transportation Systems, 15(5), 2340–2346.