I. Selective Lossy Image Compression for Autonomous Systems

Shreyan Sood
B.Tech, Department of Mathematics and Computing
Delhi Technological University
New Delhi, India
shreyansood_2k18mc110@dtu.ac.in

Yatharth Ahuja
B.Tech, Department of Electrical Engineering
Delhi Technological University
New Delhi, India
yatharthahuja@2k18ee249@dtu.ac.in

***Abstract*— The main objective of this paper was to effectively interface object detection based on Convolution Neural Networks (CNNs) with selective lossy image compression techniques to improve the efficiency of subsequent image operations and reduce the memory requirement for storing the images in autonomous applications of self-driving vehicles. Object detection and localization was performed using 2 state-of-the-art CNN based models from the Tensorflow 2.0 Object Detection API - Faster R-CNN ResNet152 V1 1024x1024 and CenterNet HourGlass104 1024x1024. Lossy Image Compression centred around the most prominent detected object (which is preserved) is done through 3 techniques – K-Means Clustering (KM), Genetic Algorithm (GA), Discrete Cosine Transform (DCT). The compressed and preserved parts were recombined to produce the final image. Analysis of the results obtained from different models and compression techniques was carried out. It was found that DCT produced the best results on both the models.**

Keywords—CNN, Object Detection, Lossy Image Compression, Autonomous Systems

II. Introduction

Object detection is a computer vision technique used to identify and locate objects in an image or video implemented by drawing bounding boxes around the detected objects. Object Detection finds many applications, including image classification, human behaviour analysis, face recognition, astronomical photography and autonomous systems such as self-driving vehicles. The implementation process of object detection is a combination of object localization and object classification. Hence the process pipeline of traditional object detection models can be mainly divided into three stages: informative region selection, feature extraction and classification and localization. Most state-of-the-art object detection models are based on backbones of deep CNNs followed by either a single-stage (CenterNet) or a double-stage (Faster R-CNN) detection architecture.

In most autonomous applications, there is a profound impact and interest in perceiving visual data and its subsequent processing and storage. Very often, such a visual data flux pipeline causes a great stress on processing units installed on respective autonomous systems, due to their limited capacity and functionality suited to remote operations. After a region of interest (ROI) has been identified in preceding image analysis steps, it makes the further development convenient by compressing the portions of image outside of ROI. This enables the processing unit to get rid of unimportant information and provide better image analysis performance. Since, in this application of compression techniques we are dealing with visual information outside ROI, it is feasible and profitable to exploit the tendency to lose such redundant information. Hence, we inflicted lossy compression on the non-ROI areas of processed image via the techniques of K-Means Clustering, Genetic Algorithm and Discrete Cosine Transform.

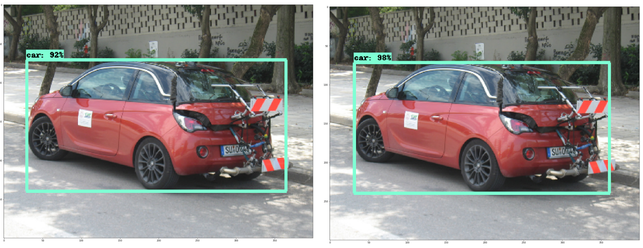
III. Methods

A. Object Detection

Faster R-CNN is a 2-stage object detector. In the first stage the image is scanned using a sliding-window approach and ROI proposals are generated. In the next stage the ROIs are classified and corresponding bounding boxes for detected objects are generated. The Faster R-CNN architecture consists of 3 important modules: 1. Backbone CNN and FPN: This is a standard CNN (ResNet 152) that serves as a feature extractor. FPN refers to the Feature Pyramid Network which helps in localizing and concatenating features across different layers by taking the high-level features from the first pyramid and passing them down to lower layers.
2. Region Proposal Network (RPN): The RPN is a lightweight neural network that scans the image through a sliding-window algorithm and finds areas that might contain detectable objects using the concept of anchor boxes.
3. ROI Classifier and Bounding Box Regressor: This module is basically a combination of ROI pooling operation and the classification SoftMax layer which classifies the detected object and generates the coordinates of the bounding boxes.

CenterNet is a 1-stage object detector based on an extended version of CornerNet that uses keypoint triplets. A convolutional backbone network (HourGlass104) applies cascade corner pooling and center pooling to output two corner heatmaps and a center keypoint heatmap, respectively. Like CornerNet, a pair of detected corners and the similar embeddings are used to detect a potential bounding box. Then the detected center keypoints are used to determine the final bounding boxes. The CornerNet architecture detected the 2 corner points of the bounding boxes Making predictions based on only these 2 corner keypoints had a major flaw since even though the algorithm was sensitive to the boundary of objects it was not aware of which pairs of keypoints should be grouped into objects. Consequently, it often generated some incorrect bounding boxes. Hence, this network was extended to add a third central keypoint which provided the ability to perceive visual patterns in the center of the detected object and identify the correctness of the bounding boxes by itself.

Both chosen models - Faster R-CNN ResNet152 V1 1024x1024 and CenterNet HourGlass104 1024x1024 were implemented using TensorFlow 2.0 Object Detection API on Google Colaboratory. Since these models were pre-trained on the COCO 2017 dataset which already included relevant classes such as “Car”, “Truck”, “Person”, “Bicycle”, “Bus”, “Traffic Light”, “Motorcycle”, “Stop Sign” etc., no additional training of the heads layer of the models was required. Inferencing was done on a test dataset containing 20 images of cars on roads. The coordinates of the bounding box of the object detected with the highest confidence score were extracted.



1. CenterNet   2. Faster R-CNN

B. Image Compression

Apart from the pixels inside the coordinates of the extracted bounding box, rest of the images were compressed using the following 3 techniques.

1. K-Means Clustering

   This technique performed quantization of colors present in the image by selecting K different colors as centroids representative of similar colors. Similar colors were then clustered together and hence the number of color combinations used to represent the image got significantly reduced resulting in lossy compression.

2. Genetic Algorithm

   The decimal value of each pixel was converted to binary. A specific number of the least significant bits of the binary value were discarded according to the given number of generations. These reduced binary values were converted back to decimal, resulting in the required lossy compression.

3. Discrete Cosine Transform

   DCT is a technique for converting a signal into elementary frequency components. DCT-based image compression relies on two techniques to reduce the data required to represent the image:

   i. Quantization: It is the process of reducing the number of possible values of a quantity, thereby reducing the number of bits needed to represent it.

   ii. Entropy coding: A technique for representing the quantized data as compactly as possible.

   For analysis of two-dimensional (2D) signals such as images, we use a 2D version of the DCT. It is deployed in the Blocked DCT form.

Following the 3 techniques of lossy image compression, the retained part of the image was recombined with the compressed part of the image to get the final compressed image.

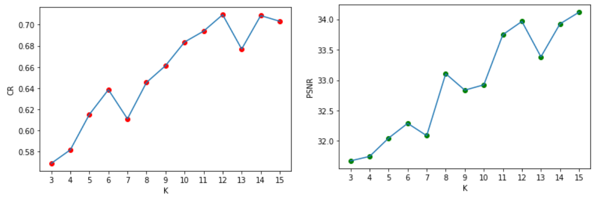
IV. Results

The following metrics were used for evaluation of the final compressed image:

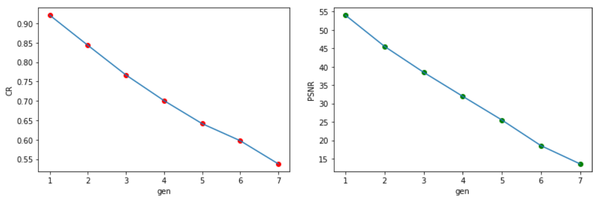
i. Peak Signal to Noise Ratio (PSNR): The ratio between the maximum possible power of a signal and the power of corrupting noise that affects the fidelity of its representation.

ii. Compression Ratio (CR): Compression ratio, also known as compression power, is a measurement of the relative reduction in size of data representation produced by a data compression algorithm. It is typically expressed as the division of uncompressed size by compressed size.

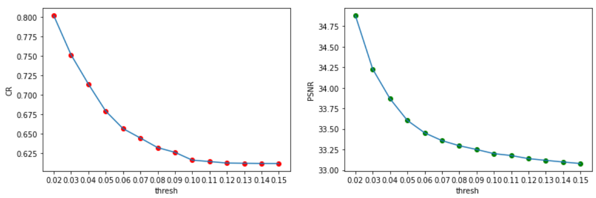
Hyperparameter Tuning was carried out after plotting graphs of the metrics with respect to the tunable parameter. It was found that all test images gave the same graph pattern and hence the optimal parameter for each technique was obtained.



Optimal K = 7 for K-Means

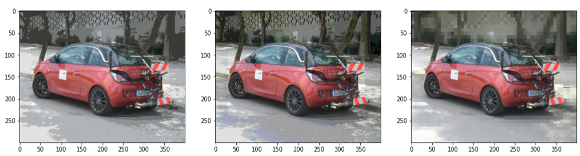


Optimal number of generations = 5 for GA



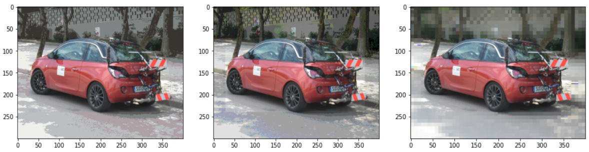
Optimal threshold value = 0.10 for DCT

Using these optimal hyperparameters, compression inference was run on the test dataset for both models.



K-Means   GA   DCT

Faster R-CNN



K-Means GA DCT

CenterNet

Means of the metrics on the 20 test images were calculated on both models and recorded in the table below.

TABLE I. Calculated Mean Metrics

S.No.	Compression Technique	Faster R-CNN		CenterNet	
		Mean PSNR	***Mean CR***	***Mean PSNR***	***Mean CR***
1	K-Means	32.85	0.63	32.88	0.64
2	GA	25.40	0.59	25.38	0.59
3	DCT	33.18	0.55	33.17	0.56

V. Conclusions

- Both Faster R-CNN and CenterNet perform equally well for all 3 compression techniques with considerably negligible differences.
- DCT gives the best CR as well as PSNR in both models. It outperforms K-Means marginally in both metrics.
- GA performs significantly worse as compared to the other 2 techniques.
- In terms of speed of execution, DCT is the fastest followed by K-Means and then GA.

References

[1] Kaiwen Duan, Song Bai, Lingxi Xie, Honggang Qi, Qingming Huang, Qi Tian, “CenterNet: Keypoint Triplets for Object Detection”, in Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), 2019, pp. 6569-6578

[2] Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks”, in IEEE Transactions on Pattern Analysis and Machine Intelligence (Volume: 39, Issue: 6, June 1 2017)

[3] Fan, Bohan, "Selective Compression of Medical Images via Intelligent Segmentation and 3D-SPIHT Coding" (2018). Theses and Dissertations. 1795. https://dc.uwm.edu/etd/1795

[4] Wang, Cheng; Han, Yifei; Wang, Weidong. 2019. "An End-to-End Deep Learning Image Compression Framework Based on Semantic Analysis" Appl. Sci. 9, no. 17: 3580. https://doi.org/10.3390/app9173580