# Exploratory Data Analysis (EDA) Report on Cityscapes dataset

Project title: AI-based Autonomous Driving

Prepared by: NITESH KUMAR

Date: 16/10/2024

# Introduction

In this study, the first exploratory data analysis (EDA) of the Cityscapes dataset is presented. Images with pixel-level annotated counter parts and JSON files with object polygon data are included in the collection for applications including semantic segmentation, lane detection, object detection/avoidance, and traffic sign identification. Understanding the properties of the data, evaluating the necessity for preprocessing, and identifying any potential difficulties with deep learning model training are the objectives of this investigation.

# Dataset and Methodology (Exploration)

Data Resolution and Channels

The image's resolution, which is determined by its dimensions (2048 × 1024 pixels), tells us how detailed the image is. More detail is usually available at higher resolutions, which is advantageous for applications like object detection or semantic segmentation that require for fine-grained analysis. Larger images, however, also need more processing power, so it's critical to strike a balance between processing speed and resolution.

The image is in color, especially an RGB image, where each pixel is represented by three values that correspond to the intensities of Red, Green, and Blue. This is indicated by the number of channels (3 in RGB). Choosing the right algorithms for tasks like picture modification or normalization requires an understanding of the number of channels.

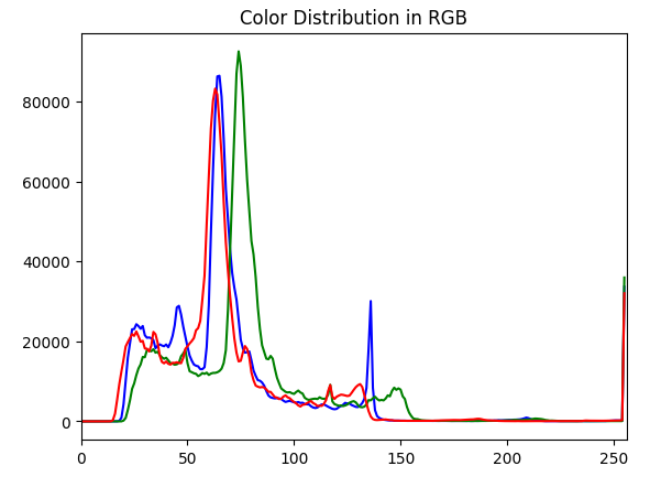
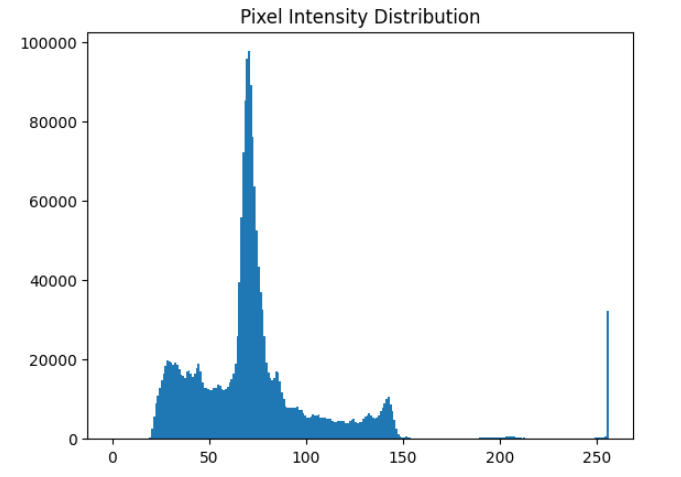
Lastly, the aspect ratio of 2.00 indicates that the image's width is double its height. The aspect ratio is crucial for operations like cropping and scaling, where maintaining the image's natural proportions avoids distortion and preserves the visual content's integrity.

Data Composition

The dataset, which consists of 5000 photos in total, is divided into train, validation, and test sections. Of these, 2975 are for trains, 500 are for validation, and 1525 are for tests; that is, 59.5% of the percentage goes to train images, 10.0% to validation images, and 30.5% to test images.

Pixel and color Intensity distribution

The pixel intensity distribution of a picture is depicted by the histogram, where the y-axis shows the number of pixels that correspond to each intensity level and the x-axis shows pixel intensity values ranging from 0 (black) to 255 (white) for an 8-bit image.



Key observations from the histogram:

1. Pixel Intensity Concentration: The distribution displays a significant spike in the lower intensity range, particularly in the 50–75 range, suggesting that the image has a lot of darker pixels or areas with lower brightness. This could imply that there are a lot of shadows or darker regions in the picture.
2. High and Mid-Range Intensities: With the exception of a sharp peak close to the maximum value (255), there are relatively few pixels in the higher intensity range and fewer in the mid-range (between 100 and 200). This implies that even though the image is mostly dark, there are some white or brilliant spots that might be surfaces or bright objects that are extremely reflective.

3. Sharp Peak at 255: A large number of pure white pixels are present near 255, which may suggest regions of high brightness like light sources, traffic signs, lane markings, or other bright objects. This noticeable rise could indicate areas of the image that are saturated.

Blurriness score on Laplacian scale

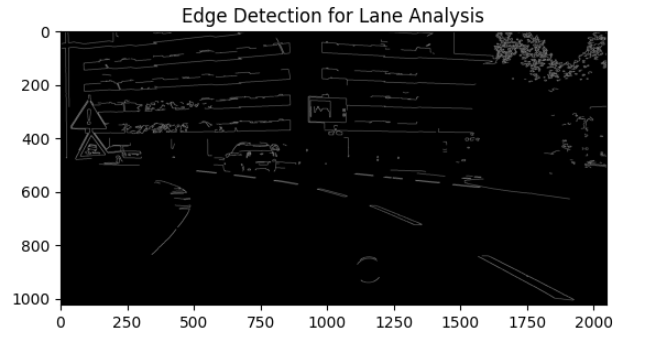
The Blurriness score 48.25 indicates a moderately blur image, on this scale higher is more sharp, considering 100 as sharp, 48.25 on our dataset wouldn’t affect the model much for training, but if exposed to other images for analysis may become a point of concern.

Edge Detection Annotation Using Canny on Grayscale

By identifying abrupt changes in color or intensity, edge detection—a basic image processing technique—highlights the boundaries inside an image. Because of its accuracy and precision, the Canny Edge Detection algorithm is a widely used technique for edge detection.

First, the image is converted to grayscale (if it isn’t already) to simplify the analysis. In grayscale, each pixel represents brightness intensity, making it easier to identify intensity changes.

To cut down on noise, a Gaussian filter is used to smooth the image. By calculating the image's intensity gradient, areas with a sharp shift in pixel values are identified. To lessen the possibility of identifying erroneous edges, thin the edges so that only the most important edges remain. The algorithm tracks the edges across the image and applies two thresholds to identify strong and weak edges.



Resizing

Images are frequently resized as part of the preprocessing process, particularly when working with machine learning datasets. To facilitate training models by guaranteeing constant input dimensions, the goal is to make all photos the same size.   
Due to the homogeneous resolution of all the photos in our dataset, resizing is not required; nevertheless, it can be done by the aspect ratio to improve training and save computing resources.

Normalizing

The process of normalization involves scaling an image's pixel values to a predetermined range, usually between 0 and 1 or -1 and 1. In machine learning models, this is particularly crucial because it aids in stability, Because normalized data prevents the model from being stuck because of outlier pixel values, it speeds up convergence during training.   
Uniformity: Normalization ensures consistency throughout the collection, even though individual photos may have varied pixel intensity ranges.

Data Integrity

When dealing with huge datasets, maintaining data integrity is essential, particularly for projects that involve picture datasets and the annotations that go with them (stored in JSON format and PNG masks). It is necessary to clean the dataset by eliminating corrupted values and filling in missing values. Our dataset was found to be clean upon inspection because there were no corrupted images.

# Results

# Although it doesn't interfere with training, the dataset's pixel intensities, which are clustered around 75 throughout all three channels, may have an impact on generalization to different lighting situations because the images were captured in dim daylight.

# Resizing the images isn't strictly necessary but to save up on some computational resources can be resized in accordance with aspect ratio. The images were normalized between [0 1].

# Since the dataset lacks bounding box annotations, the training must rely on polygon-based annotations to form bounding boxes and masks to be used for object detection.

Due to the small amount of annotated samples and excessive blurriness (with a score of 48.25), the 40–50 samples that were manually annotated for YOLOv8 training were unable to generalize to unseen images. The model's performance was constrained by the small sample size, and blurriness affected edge recognition and feature extraction. A bigger, more varied dataset with higher-quality photos would be required to enhance the results.

The Annotation part for lane detection is done using Canny algorithm, since there is no annotations present in gtfine set for it. This will help in detecting the edges and to proficiently map the lane markings

# Conclusion

Annotation processing is required since certain modules, such as object and lane detection, lack annotation and labeling, however the Cityscapes dataset of images is already partially preprocessed and may not require much preparation.

The model trained with this dataset can well perform on the test images of same dataset, but may cause some discrepancy on the outside data.

# Future Objectives (for the next two weeks)

Data Loading Functions: These features make it easier to handle and get image and annotation files from the dataset, guaranteeing seamless access for further processing stages.

Image Preprocessing: To improve image quality and get it ready for analysis, image preprocessing entails steps like scaling, normalization, and edge detection. This ensures that the data is appropriate for model training.

Annotation Preprocessing: This step ensures compatibility with the selected object identification framework and increases the precision of model predictions by transforming polygon-based annotations into segmentation masks or bounding boxes.

Dataset Preparation & Splitting: Since our dataset has previously been separated into train, validation, and test, this step may not be required. However, we can alter the percentage of the already divided dataset to obtain more accurate results.

# References

# https://link.springer.com/article