Traffic Sign Recognition for Varying Lighting Conditions Using Unsupervised Clustering and YOLOv8

Samson Connelly

Indiana University – Luddy School of Informatics, Computing and Engineering, Bloomington, Indiana, 47406, United States

Traffic Sign Recognition is an extremely important part of autonomous vehicles. If autonomous vehicles cannot comply with laws displayed by traffic signs it causes serious safety concerns. Varying lighting and weather conditions have a negative effect on traffic sign recognition. This problem can be solved by training custom models on different lightning conditions. Images are converted into the HSV color space then converted into a histogram based on the V value. Next Principal Component Analysis is applied to the histograms and Fuzzy C Means clustering is used to classify the images into clusters based on how similar the histograms features are. A YOLOv8 model is then trained on each cluster. This allows the model to be chosen based on lighting conditions. A much larger dataset is required to prove that this pipeline is effective some classes achieved comparable scores to the general model. This solution shows potential in solving the lighting issue, however not enough data was used for this study to be conclusive.

I. Introduction

Autonomous vehicles must be able to identify and comply with traffic signs. The main challenge with traffic sign recognition is changing lighting and weather conditions. Models need to be accurate and reliable in varying conditions because misclassified or missed detections could have serious safety concerns. To solve this problem this paper presents a method that creates specific models that are trained on various lighting conditions. This is done by using unsupervised clustering to group images into categories based on lighting conditions, and training specialized models based on these clusters.

II. Related Works

Many approaches to TSR focus on preprocessing steps to remove glare and haze [1]. Other approaches include attempting to classify the image in order to adapt the model to the current conditions. Previous attempts have used supervised classification for weather and lighting conditions. However many of these studies do not demonstrate an increase in accuracy metrics on unseen data [4]. These results suggest that classes defined by humans do not offer an accurate representation of the data.

Researchers have used unsupervised learning to capture patterns in images [2]. Unsupervised learning groups data into clusters based on the similarity of features of the data. Researchers have found that Gaussian Mixture Models (GMM) allow images to be placed into more nuanced categories, because they assign a probability that the data point belongs to each cluster [2]. Some researchers have also found that applying Principal Component Analysis improves the classification of images when using unsupervised learning [6].

There are two main types of unsupervised learning, hard clustering and soft clustering. Hard clustering assigns each data point to a cluster, while soft clustering functions like GMMs assigning probabilities for each cluster to the data point. More research directly compares K-Means (hard clustering) and Fuzzy C Means (soft clustering), and concludes that Fuzzy C Means (FCM) is better for image classification [3].

Since we are attempting to classify based on lighting we must convert the image from RGB color space to the HSV color space. The V value in HSV represents the brightness of an image, therefore we can convert the image to a histogram of the V value to obtain information about the lighting of the image [5].

III. Methodology

This paper was implemented using the Self Driving Cars Dataset which can be found on roboflow. The data consists of 5609 annotated images of traffic signs and stop lights. Each image was resized to be 416x416 and no image augmentation was applied to the dataset. The dataset was split into 80% train, 10% val, and 10% test data for each cluster and 70% train, 20% val, and 10% test for the general model.

During preprocessing each image was converted from RGB color space to the HSV color space which gives more information on lighting conditions and allows the images to be separated based on lighting. Next the image data was transformed into a histogram based on V(Value). V represents the brightness of an image enabling us to separate the images based on lighting conditions [5].

Next Principle Component Analysis (PCA) was applied to the histogram from 32 components to 16 components. The application of PCA is used to reduce data complexity and noise therefore increasing data quality [6].

Fuzzy C Means (FCM) was chosen as the unsupervised learning model. This choice was made because FCM assigns a probability of the data point belonging to each cluster. This allows for the creation of an ambiguous cluster. The ambiguous cluster is made of the images whose highest probability was less than 0.6. The reasoning for this decision is to increase the uniqueness of each cluster, therefore increasing the uniqueness of each corresponding model.

FCM requires a value for the hyperparameter C, the number of clusters. In order to find the optimal value for C the elbow method was used. In order to use the elbow method I ran FCM where C = 2, 3, 4, … 10. For each clustering of images the Fuzzy Partition Coefficient (FPC) was calculated. Then the number of clusters was graphed against the FPC. The optimal number of clusters is when the FPC starts to flatten out. Based on Figure 1, 4 clusters were chosen.

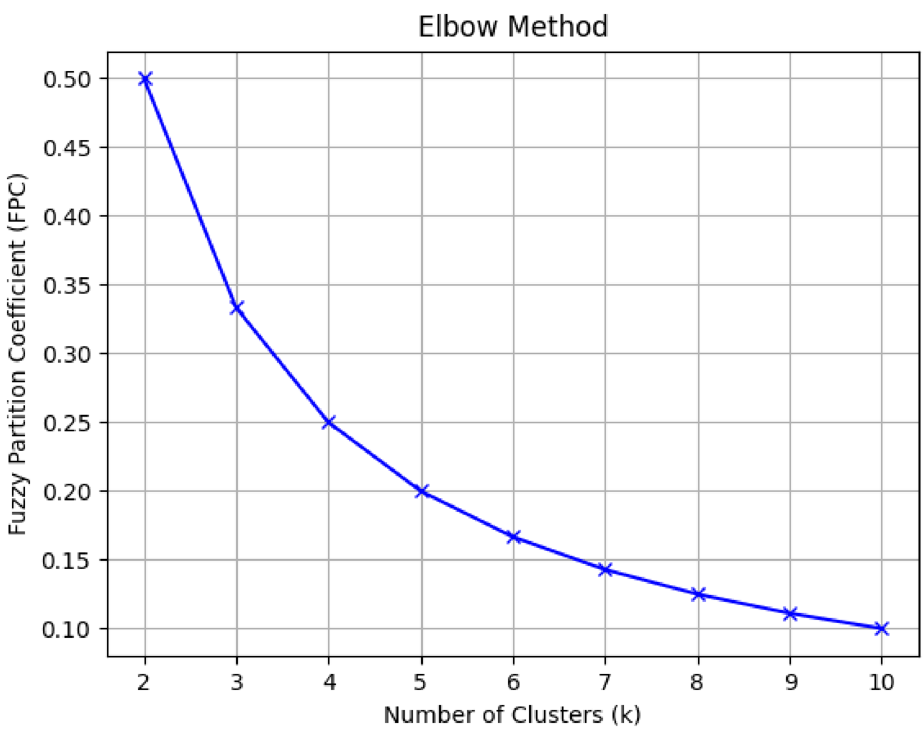

Figure 1. Elbow Method

Figure 2 displays a graph of the 4 clusters, and an ambiguous cluster category for data with less than a 60% Probability

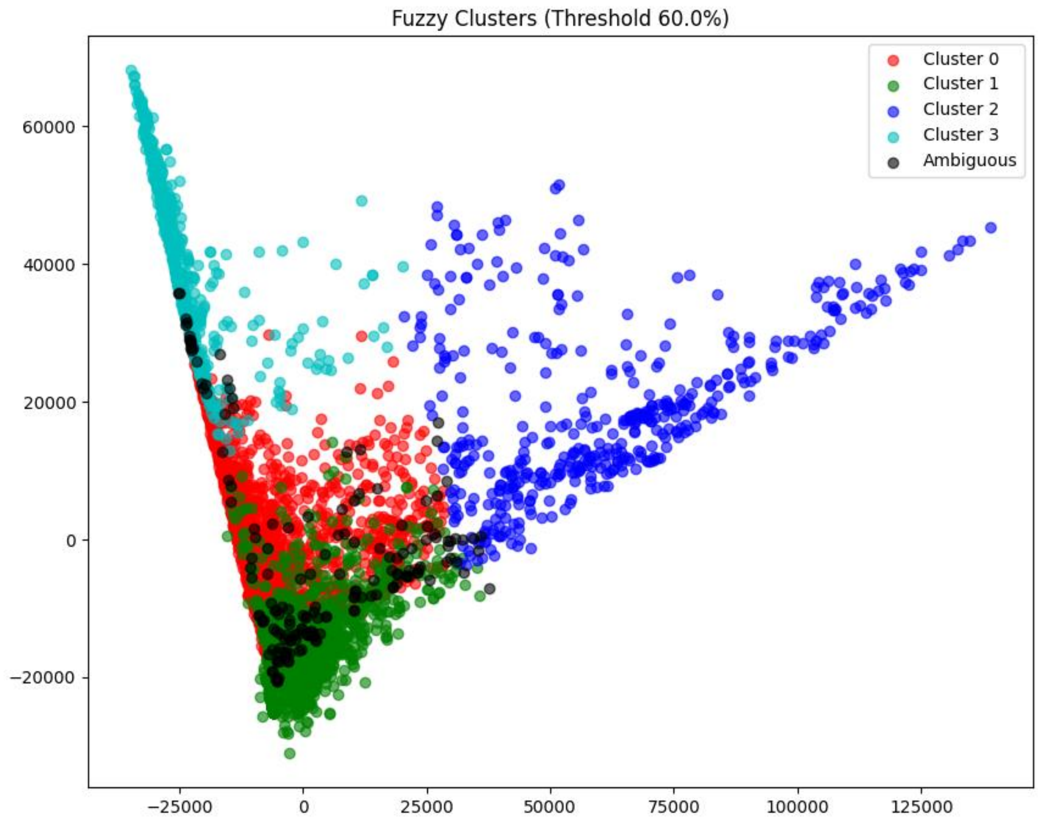

Figure 2. Fuzzy C Means Clusters

Table 1 displays the number of images in each cluster and their train, test, val split.

Cluster	Train	Test	Validation	Total
Cluster 0	1,609	202	200	2,011
Cluster 1	490	62	60	612
Cluster 2	364	46	44	454
Cluster 3	1,377	173	171	1,721
Total	3,840	483	475	4,798

Table 1. Dataset Split Based on Cluster

Figures 3 – 6 display the number of instances of each class in the specified cluster. Figure 7 displays the number of instances of each class in the entire dataset.

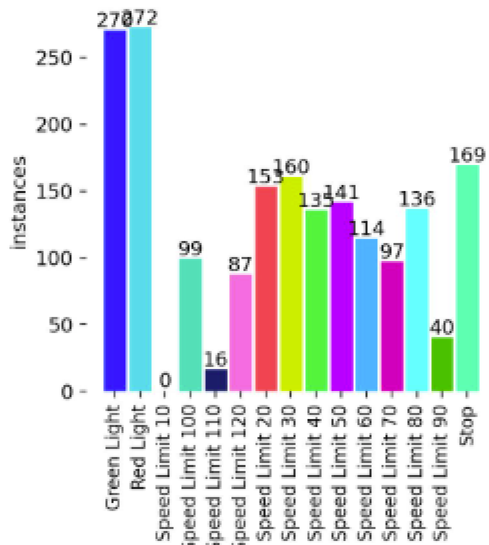

Figure 3. Cluster 0 Instances

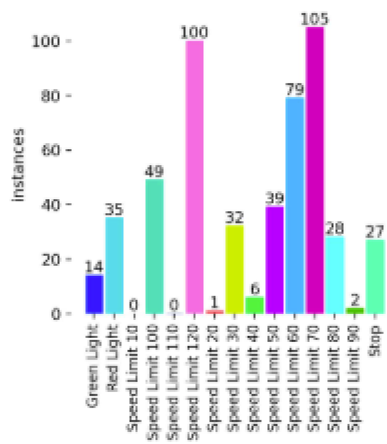

Figure 4. Cluster 1 Instances

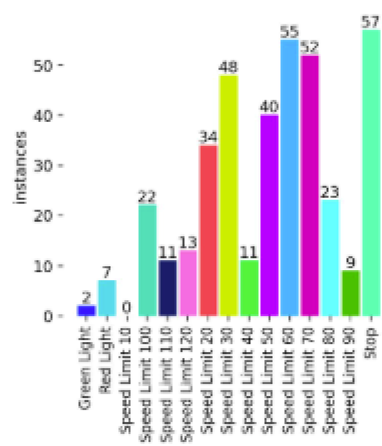

Figure 5. Cluster 2 Instances

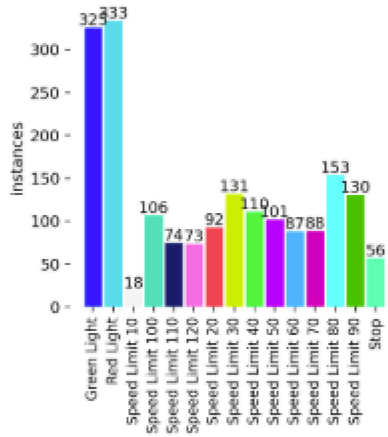

Figure 6. Cluster 3 Instances

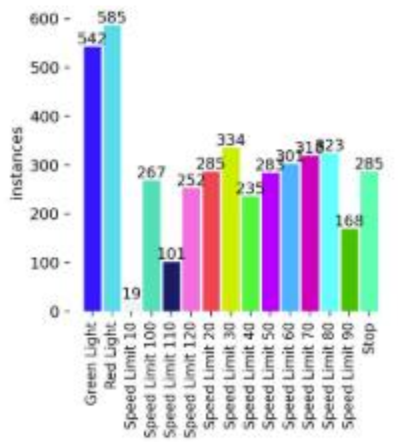

Figure 7. General Dataset Instances

Then a YOLOv8 model was trained on each cluster using these parameters, epochs = 100, patience = 5, mixup = 0.1. For comparison a YOLOv8n model was trained on the entire dataset as a baseline.

When making a prediction on an image the same pipeline is followed. First the same HSV histogram is calculated using the same algorithm. Next the PCA model (calculated earlier) is loaded and applied to the test image. The FCM centers are loaded and used to calculate the cluster probabilities of the image using the calculation in Equation 1 [7].

$$u_{ij} = \left[\sum_{t=1}^{c} \left(\frac{||x_j - v_i||_A}{||x_j - v_t||_A} \right)^{\frac{2}{m-1}} \right]^{-1} \tag{1}$$

Substituting in known parameters c=4 and m=1.1. It can also be assumed that A=I, and since script is only predicting a single image the calculation can be simplified to Equation 2.

$$u_i = \left[\sum_{t=1}^{4} \left(\frac{||x - v_i||}{||x - v_t||} \right)^{20} \right]^{-1} \tag{2}$$

Which can be implemented in python by code in Figure 8.

```python
#Load the cluster centers
centers = np.load("fcm_centers.npy")
#Compute ||x - v_i|| for each cluster
distances = np.linalg.norm(centers - x_pca, axis=1)
#Set constants
m = 1.1
power = 2/(m-1)
#Initialize array of probabilities
probs = np.zeros(len(centers))
#Complete the summation and raise each output to the -1st power
for i in range(len(centers)):
    probs[i] = np.sum((distances / distances[i]) ** power)
    probs[i] = 1/probs[i]
#find the maximum probability and assign the cluster ID
max_prob = np.max(probs)
cluster_id = np.argmax(probs)
```

Figure 8. Code For Probability Calculation

Then the YOLOv8 model chosen based on the cluster ID of the image, and runs inference on the image. If the highest probability of the image is less than 0.6, then the general model is selected. A diagram of the entire system architecture can be found in Figure 9.

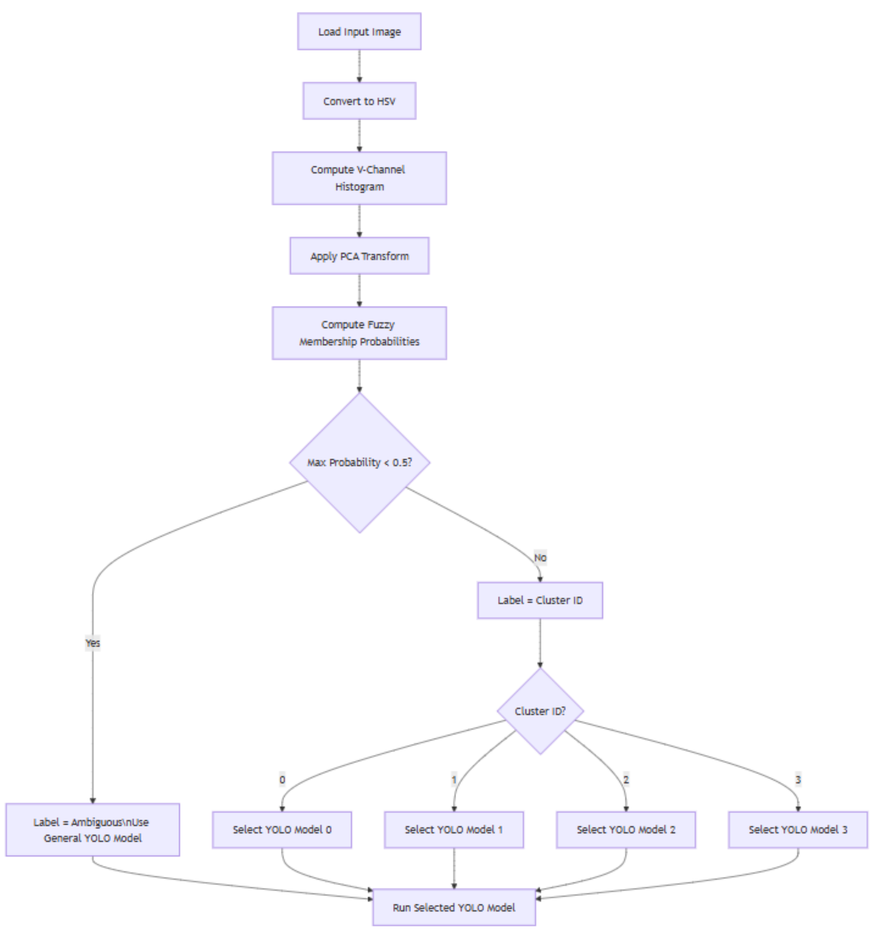

Figure 9. System Architecture

IV. Results

Each cluster has a distinct lighting pattern, demonstrating the methodology used for unsupervised classification is reliable. Cluster 0 is slightly dark, cluster 1 is very dark, cluster 2 is very light, and cluster 3 is slightly light.

Image 1. Cluster 0

Image 2. Cluster 1

Image 3. Cluster 2

Image 4. Cluster 3

When evaluating an object detection model the mAP score is frequently used [8]. The precision recall curve and the mAP score can be viewed in each figure below. The more the line follows the top right corner of the graph the better the model performs. AP scores are the area under the precision recall curve, and the mAP of the model is the mean value of the AP scores for each class [8]. The higher the mAP score the better the model performs. For our case the mAP scores can be found in Table 2.

Category	mAP	Stop Sign mAP
General	0.904	0.995
Cluster 0	0.604	0.992
Cluster 1	0.413	N/A
Cluster 2	0.085	N/A
Cluster 3	0.464	0.995

Table 2. mAP scores

The reason that cluster 1 and 2 do not have a Stop Sign mAP is because there are no stop sign instances in the validation set. From the figure you can see that the mAP for the clusters are significantly lower than the general model. However, the stop sign mAP score for cluster 0 is equal to the general stop sign mAP score and the stop sign mAP for cluster 3 is only slightly less than the general stop sign mAP. This Phenomenon can also be observed in the precision recall graphs. The precision recall curves for each model can be observed in Figure 10-14.

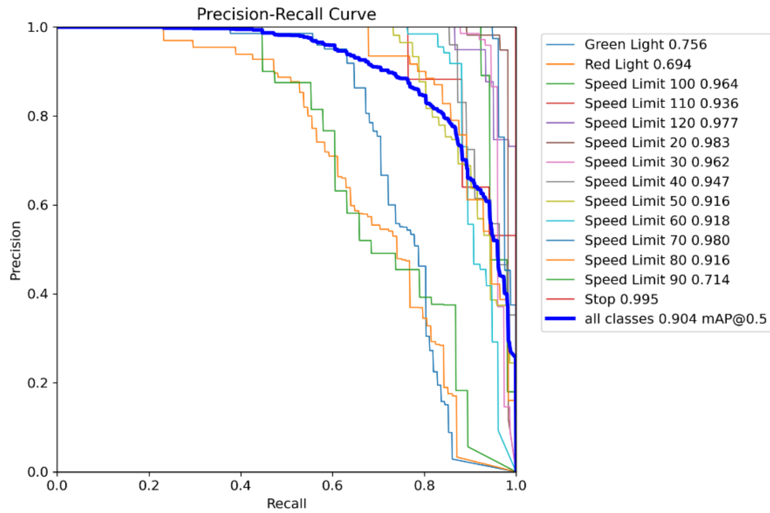

Figure 10. PR Curve for General Model

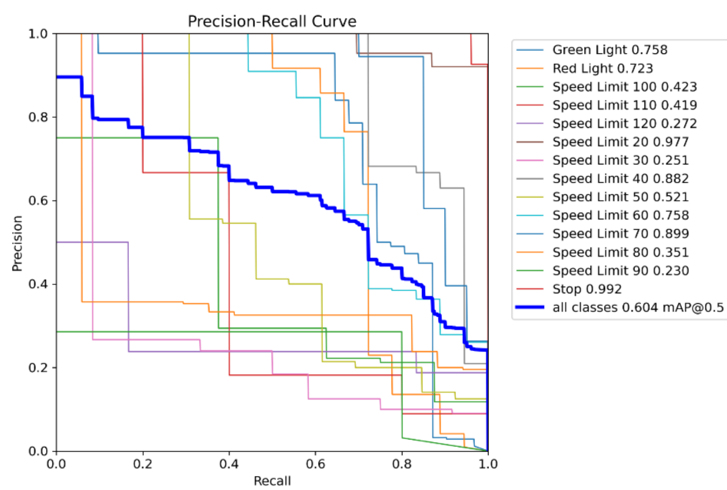

Figure 11. PR Curve for Cluster 0

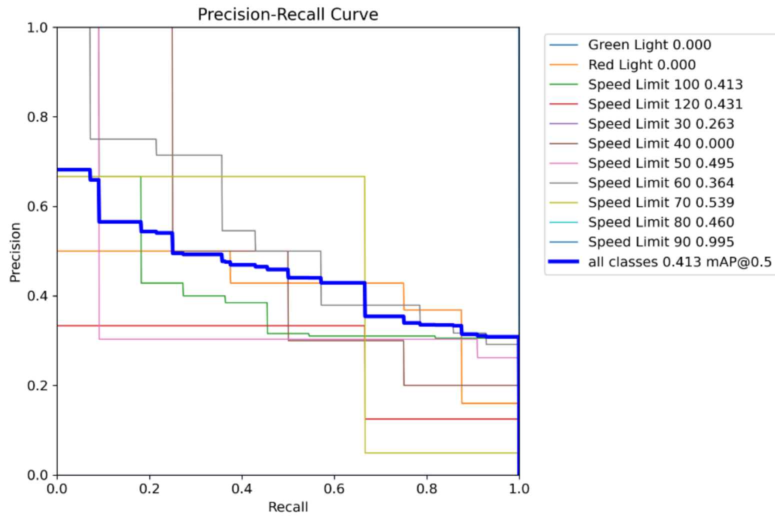

Figure 12. PR Curve for Cluster 1

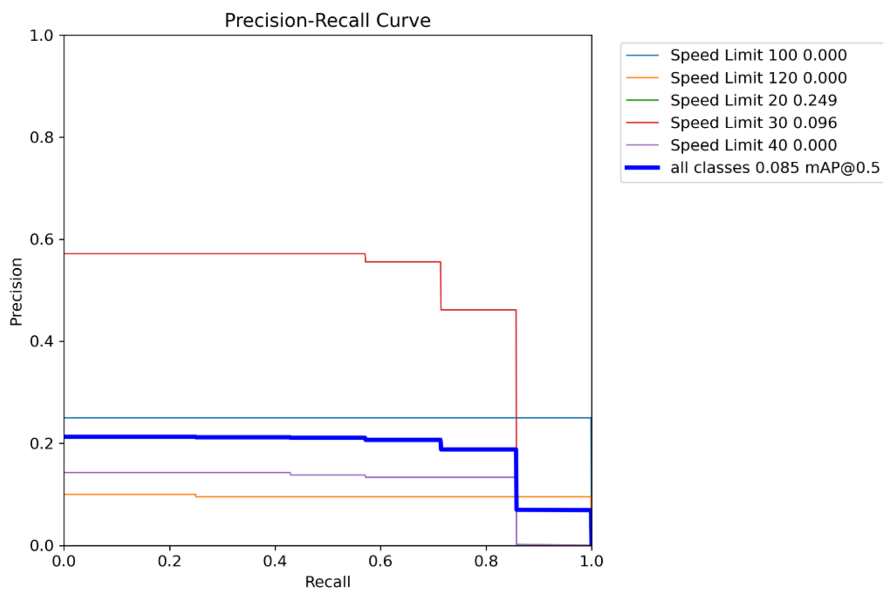

Figure 13. PR Curve for Cluster 2

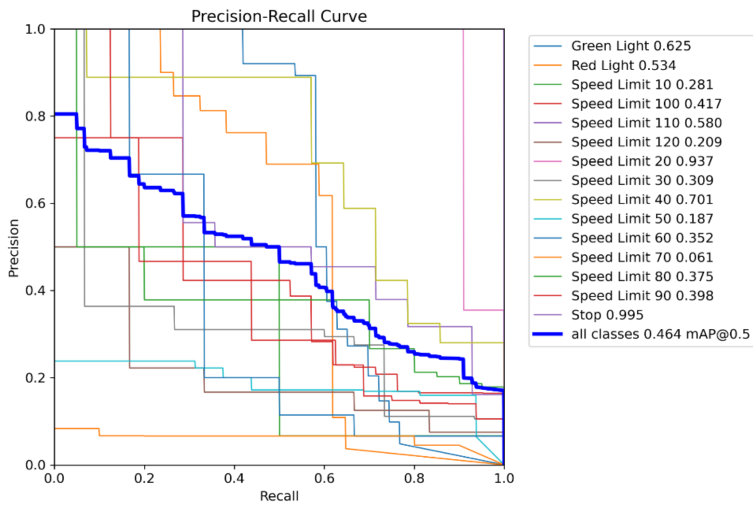

Figure 14. PR Curve for Cluster 3

Due to a lack of training data in cluster 1 and 2 slight modifications were made to the prediction algorithm discussed in methodology. If the image is in cluster 1 then the model for cluster 0 is selected, and if the image is in cluster 2 then the model for cluster 3 is selected. In Figure 15 is the updated system architecture.

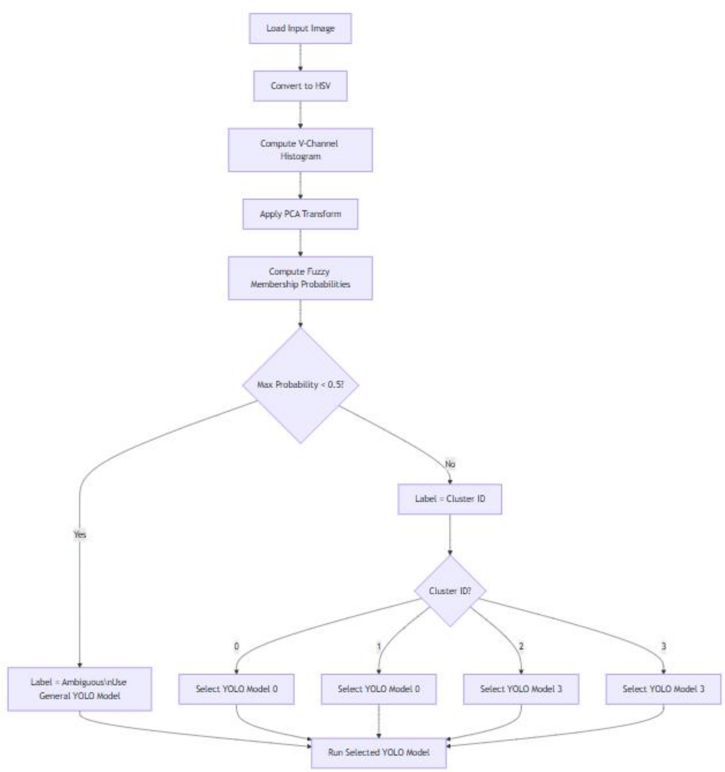

Figure 15. Updated System Architecture

In order to test model results Image 5 is ran through our script.

Image 5. Test Image

The image is classified as cluster 2 and the output is Image 6.

Image 6. Unsupervised Classify Results

And the output of the general model is Image 7.

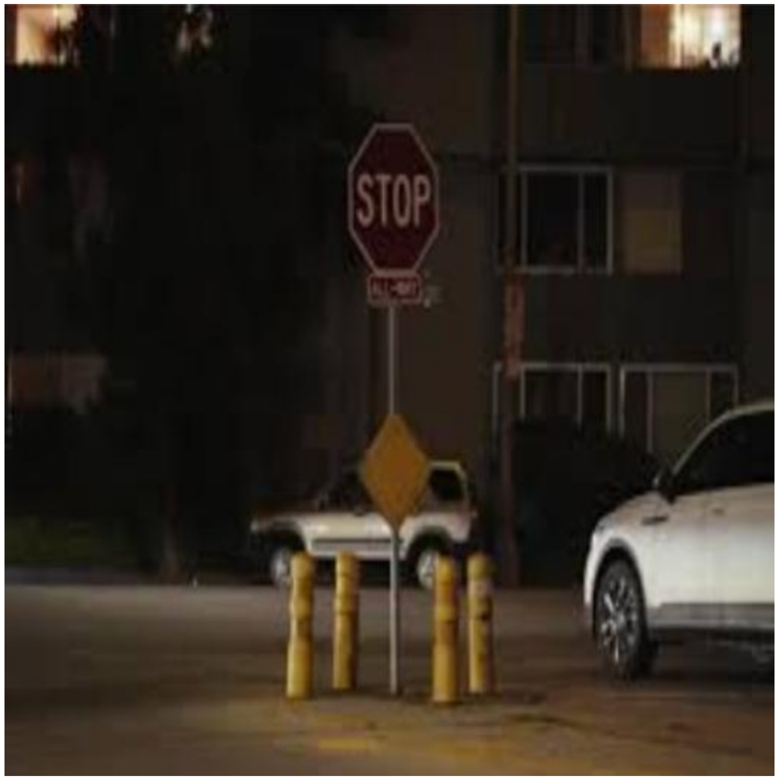

Image 7. General Model Output

When using the model based on the unsupervised classification, the stop sign is correctly predicted at a confidence of 0.45 and a red light is incorrectly predicted at a confidence of 0.26. When using the general model, no predictions are made.

V. Conclusion

This paper explored using unsupervised clustering to customize object detection models based on the lighting conditions. Using Fuzzy C Means on HSV histograms separated the data into 4 separate categories. The results indicate that this methodology can improve accuracy however not enough data was used. Cluster 0 and cluster 3 both achieved mAP scores for stop signs that were similar to the general model. The main constraint of this study is the dataset that was used. To get a clearer picture on if this is a viable pipeline for TSR, this paper must be recreated with a larger, more diverse dataset.

References

[1] A. Radha Rani, Y. Anusha, S.K. Cherishama, S. Vijaya Laxmi, Traffic sign detection and recognition using deep learning-based approach with haze removal for autonomous vehicle navigation, e-Prime - Advances in Electrical Engineering, Electronics and Energy, Volume 7, 2024, 100442, ISSN 2772-6711,

[2] Olaode, Abass & Naghdy, Golshah & Todd, Catherine. (2014). Unsupervised Classification of Images: A Review. International Journal of Image Processing. 8. 2014-325.

[3] V. K. Dehariya, S. K. Shrivastava and R. C. Jain, "Clustering of Image Data Set Using K-Means and Fuzzy K-Means Algorithms," 2010 International Conference on Computational Intelligence and Communication Networks, Bhopal, India, 2010, pp. 386-391, doi: 10.1109/CICN.2010.80.

[4] Mahesh, Dhananjaya & Ravi Kumar, Varun & Yogamani, Senthil. (2021). Weather and Light Level Classification for Autonomous Driving: Dataset, Baseline and Active Learning. 10.48550/arXiv.2104.14042.

[5] Jarvis, D., "[CV2] HSV vs RGB: Understanding and Leveraging HSV for Image Processing," *DEV Community* [online], 29 Nov. 2024, URL: https://dev.to/jarvissan22/cv2-hsv-vs-rgb-understanding-and-leveraging-hsv-for-image-processing-49ac

[6] Das, A., "Deep Dive into Unsupervised Classification of Satellite Images: K-means, PCA, and Spectral Indices for Urban and Vegetation Mapping," *Medium* [online], 24 May 2020, URL: https://medium.com/@apratimonline/deep-dive-into-unsupervised-classification-of-satellite-images-k-means-pca-and-spectral-indices-442a5dc23219

[7] Nayak, Janmenjoy & Naik, Bighnaraj & Behera, Prof. Dr. H.. (2015). Fuzzy C-Means (FCM) Clustering Algorithm: A Decade Review from 2000 to 2014. 10.1007/978-81-322-2208-8_14.

[8] [1] Hui, J., "mAP (mean Average Precision) for Object Detection," *Medium* [online], 6 Mar. 2018, URL: https://jonathan-hui.medium.com/map-mean-average-precision-for-object-detection-45c121a31173