# Spatio-temporal Formal Verification of Pedestrian Detection System in Low-light Conditions

Swapnil Mallick (USC ID: 2258452509)
Shuvam Ghosal (USC ID: 1183526904)

## 1. Introduction:

Object detection is the process of localizing objects using bounding boxes and classifying them. It is a principal component of Advanced Driver Assistance Systems (ADAS) that enables cars to perform pedestrian detection to prevent accidents and fatalities. CNN-based models have become widespread in the field of object detection and have been able to achieve state-of-the-art results when tested on benchmark datasets. These models have also been deployed in autonomous vehicles. Although these models have fared well when tested on high-quality images captured in clear weather conditions, they have failed miserably in adverse and low light conditions like foggy conditions or during night. The main reason for this failure is domain shift since these models are usually trained using clear weather images taken during daytime and tested on images captured in adverse weather conditions or during night. This has encouraged us to improve the object detection model that performs well in adverse weather and low-light conditions.

## 2. Problem Definition:

In this project, we have mainly focussed on building and fine tuning a reliable pedestrian detection system for autonomous vehicles. For that, the concept of domain adaptation has been used on top of pre-existing models. We have also tested how the model fares in detecting pedestrians in low-light situations like at night or in foggy conditions using a state-of-the-art spatio-temporal formal verification method called *Timed Quality Temporal Logic (TQTL)* [1]. Most of the traditional perception models test their accuracy by comparing their performance with the ground truth labels. However, we have tried to use TQTL to check the quality of our pedestrian detection model since TQTL is seen as a great alternative metric in the absence of ground truth labels.

## 3. Challenges:

There are major challenges in coming up with an object detection model that performs well in low-light conditions. Most of the pre-existing models are trained on images that have been captured in clear sunny weather during daytime. But our goal is to build and test a model that can detect pedestrians in foggy conditions or during night. So, there arises a domain gap between the source domain and the target domain. Apart from that, finding a reliable dataset is also a demanding task. But since pedestrian detection systems are used in autonomous vehicles, the biggest challenge is to build a system that is reliable and safe. So, formal reasoning is required because we are dealing with systems that are *safety* and *mission-critical,* having huge implications on human health, wellbeing and economy.

## 4. Solution:

The object detection algorithms can be broadly categorized into two groups based on how they operate. One group of algorithms find the regions of interest (RoI's) and then classify the regions by training neural networks. These algorithms are called the region proposal-based methods. The other class of algorithms are called single-stage regression based methods, such as the YOLO series [2]. In this project, we have used YOLOv3 as the backbone architecture and tried to finetune it to improve its performance in low-light conditions.

Significant research has been conducted for general pedestrian detection. However, only a few efforts have been made to detect pedestrians successfully in low light conditions. Previous works have followed a rather straightforward approach where the images have been transformed using traditional dehazing and sharpening methods.

In order to enhance an image, the technique of image enhancement is usually employed. In their works, Polesel, Ramponi, and Mathews 2000 [3]; Yu and Bajaj 2004 [4]; Wang et al. 2021 [5] have tried to calculate the parameters of image transformation adaptively based on the corresponding image features. Apart from that, Zeng et al. (2020) [6] have used CNN to flexibly learn the hyperparameters of image transformation. In their seminal work, CNN has been used to learn the image adaptive 3D LUTs according to the global context such as color, brightness and tones.

Usually, images captured in adverse weather or during low-light conditions have low visibility due to weather related factors. This makes object pedestrian detection difficult in these conditions. An image adaptive detection model IA-YOLO [7] has been used to overcome this problem, which ignores the weather specific information and highlights the latent information.

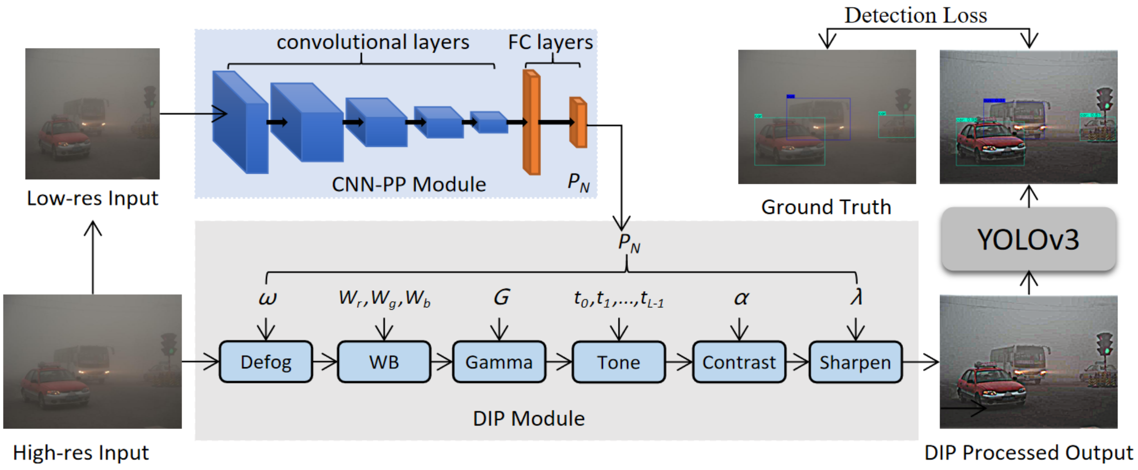

**Fig 1. The training pipeline of the IA-YOLO framework**

The pipeline consists of a CNN based parameter predictor, a differentiable image processing module (DIP) and a detection network. At first, the input image is resized to the dimension of 256 x 256 and fed to the CNN parameter predictor. The CNN parameter predictor then tries to predict the parameters of DIP. Following that, the image filtered by the DIP module is fed as input to the YOLOv3 detector for the pedestrian detection task.

The input images have been downsampled to the dimensions of 256 x 256 and the filter parameters have been learnt from these downsampled images since training CNN takes a lot of computing resources. Therefore , the image filters in the DIP module should be independent of image resolution.

**i) CNN-based Parameter Predictor:**

As shown in Fig. 1, the CNN parameter predictor module consists of five convolutional blocks followed by two fully-connected layers. Each convolutional block contains a 3 x 3 convolutional layer with stride 2 and a leaky ReLU activation layer. The module tries to understand the global content of the image, such as brightness, color and tone and the degree of fog in order to predict the parameters required by the DIP module. In order to save computation cost, the input images are downsampled to a lower resolution of 256 x 256 using bilinear interpolation. The output channels of the five convolutional layers are 16, 32, 32, 32 and 32, respectively. The output of this module is fed into the DIP module.

**ii) DIP module:**

The DIP module consists of six differentiable filters with adjustable hyperparameters, namely *Defog, White Balance (WB), Gamma, Contrast, Tone and Sharpen*. According to Hu et al. 2018 [8], the standard color and tone operators, such as *White Balance, Gamma, Contrast* and *Tone*, can be expressed as pixel-wise filters. Therefore, the filters can be classified into three categories namely, *Pixel-wise*, *Defog* and *Sharpen Filters*. The *Defog filter* has been designed for foggy scenes only.

**Pixel-wise filters:** In pixel-wise filters, an input pixel value $P_i = (r_i, g_i, b_i)$ is mapped into an output pixel value $P_o = (r_o, g_o, b_o)$, where $(r, g, b)$ represent the values of the red, green and blue color channels, respectively. The mapping functions of the pixel-wise filters have been shown in Table 1.

**TABLE I**

MAPPING FUNCTIONS OF PIXEL WISE FILTERS

| Filter | Parameters | Mapping Function |
|---|---|---|
| Gamma | G: gamma value | $P_o = P_i^G$ |
| WB | $W_r$, $W_g$, $W_b$: factors | $P_o = (W_r r_i, W_g g_i, W_b b_i)$ |
| Tone | $t_i$ : tone params | $P_o = (L_{tr}(r_i), L_{tg}(g_i), L_{tb}(b_i))$ |
| Contrast | α: contrast value | $P_o = \alpha En(P_i) + (1 - \alpha)P_i$ |

**Defog Filter:** The defog filter has been designed following the dark channel prior method He, Sun, and Tang 2009 [9]. the formation of a hazy image can be formulated as follows:

$$I(x) = J(x)t(x) + A(1 - t(x))$$

where $I(x)$ is the foggy image, $J(x)$ represents the scene radiance (clean image), $A$ is the global atmospheric light, and $t(x)$ is the medium transmission map. The atmospheric light A and the transmission map $t(x)$ need to be obtained to recover the clean image $J(x)$. At first, the dark channel map of the haze image $I(x)$ has been computed and the top 1000 brightest pixels have been picked. Then, the average of those 1000 pixels of the corresponding position of the haze image $I(x)$ has been taken to estimate the value of $A$.

**Sharpen Filter:** The sharpen filter has been used to enhance the image details. For sharpening the images, the following equation describes the process:

$$F(x, \lambda) = I(x) + \lambda(I(x) - Gau(I(x)))$$

where $I(x)$ is the input image, $Gau(I(x))$ denotes Gaussian filter, and $\lambda$ is a positive scaling factor.

**iii) Detection Module:** The single-stage detector model YOLOv3 has been used as the detection network. The same network architecture and loss function has been used as the original YOLOv3 [10]. YOLOv3 contains darknet-53 which has successive 3 x 3 and 1 x 1 convolutional layers based on the idea of Resnet.

**iv) TQTL:**

TQTL (Timed Quality Temporal Logic) is a spatio-temporal verification approach that attempts to check the robustness of real-time object detection tasks in videos or sequences of frames. This method gives us an idea of how well the quality, i.e., the spatio-temporal aspects of an object are maintained across the multiple frames of a video. It is an extension of the Timed Propositional Temporal Logic (TPTL) [11], [12]. In case of object detection, there can be multiple objects in a particular frame and this number can be dynamic. TQTL helps us to check if the system satisfies the required specification in such scenarios. A TQTL formula $\varphi$ over a finite set of predicates $P$, a finite set of frame number variables ( $V_t$), and a finite set of object ID variables ($V_{id}$) can be defined according to the following grammar:-

$$\begin{aligned}\varphi ::= \top \mid \mu \mid t.\varphi \mid \exists id@t, \varphi \mid \forall id@t, \varphi \mid \\ t \leq u + n \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \varphi_1 \, \mathrm{U} \, \varphi_2\end{aligned}$$

The time constraint is $t \leq u + n$, which implies the timespan starting from $u$ and spanning across $n$ consecutive frames. The semantics of TQTL deals with a data stream $D$, which is a sequence of video frames containing multiple candidate objects in each frame, a frame number $T$ and a valuation function $V$ to a real-valued entity. A valuation function in this context is a function that assigns some values to frames and objects present in the corresponding frames. TQTL mainly deals with the task that if a particular object is tracked across multiple frames of a video, the probability of detecting it does not fall below a certain threshold across a certain number of consecutive frames.

In our case, we focus on detecting only pedestrian type objects in foggy and night conditions. We aim to validate the following - If a person is detected with a confidence score of 0.5 or more in a particular frame, then in the next 4 frames, the probability of detecting the same person should never drop below 0.4. The TQTL monitor gets triggered as soon as the probability of detecting a person is equal to or above 0.5. After that, whenever the score becomes less than 0.4, it displays “Formula violated” and “Formula Satisfied” otherwise. The TQTL formula in our case can be represented in TQTL semantics in the following manner:-

$$\begin{aligned}\varphi = \Box(x.\forall id_1@x, (C(x, id_1) = Pedestrian \wedge P(x, id_1) \geq 0.5) \\ \rightarrow \Box(y.((x \leq y \wedge y \leq x + 4) \\ \rightarrow C(y, id_1) = Pedestrian \wedge P(y, id_1) > 0.4))\end{aligned}$$

We have chosen the confidence scores to be 0.5 and 0.4 respectively to account for the comparatively poorer performance of the IA-YOLO as compared to the Vanilla YOLO which is used to detect objects in much clearer images. We created a custom dataset containing night and foggy driving videos and also monitored the robustness values of the model against the above TQTL formula as shown in Table II.

## 5. Results :

**TABLE II**

ROBUSTNESS VALUES ACHIEVED ON CUSTOM DATASET USING IA-YOLO MODEL AGAINST $\varphi$

| Frame Sequence | Robustness |
|---|---|
| Foggy_Sequence1_Pos | 0.131 |
| Foggy_Sequence1_Neg | - 0.4 |
| Foggy_Sequence2_Pos | 0.254 |
| Foggy_Sequence2_Neg | - 0.096 |
| Night_Sequence1_Neg | - 0.4 |
| Night_Sequence1_Pos | 0.122 |
| Night_Sequence2_Pos | 0.158 |
| Night_Sequence2_Neg | - 0.088 |

### I. Object Detection in foggy condition:

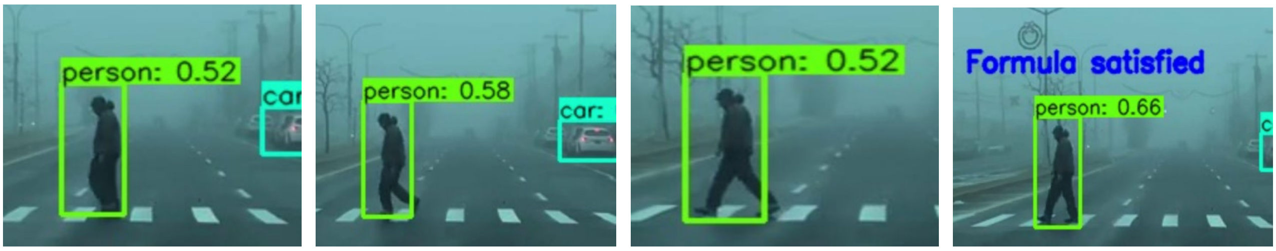

Fig 2. Pedestrian detected and TQTL formula satisfied

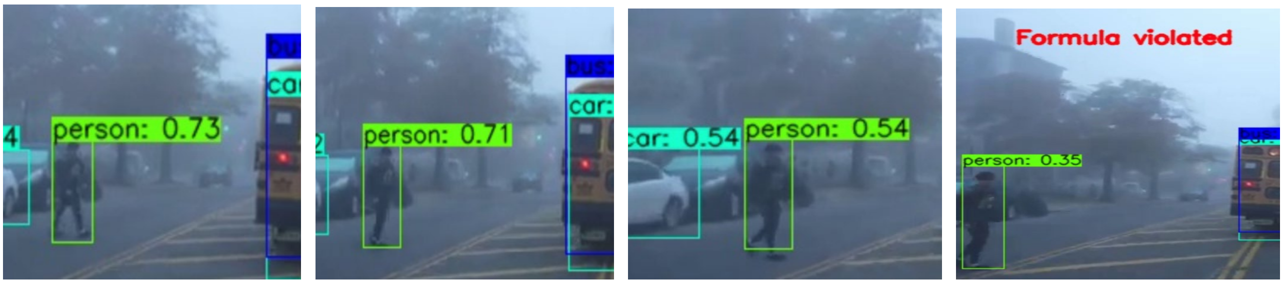

Fig 3. Pedestrian detected but TQTL formula violated

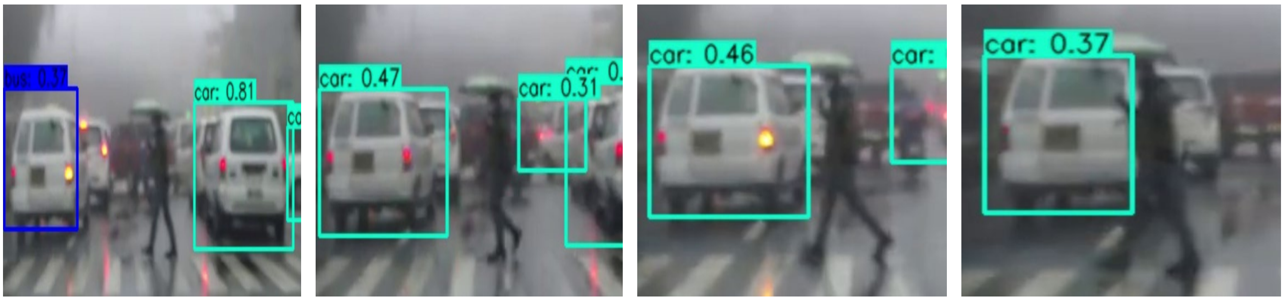

Fig 4. No pedestrian detected

### II. Object Detection in night condition:

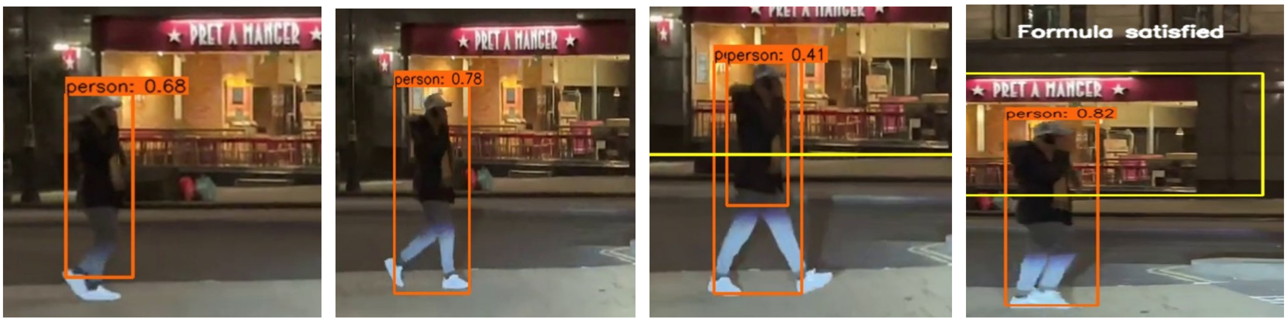

Fig 5. Pedestrian detected and TQTL formula satisfied

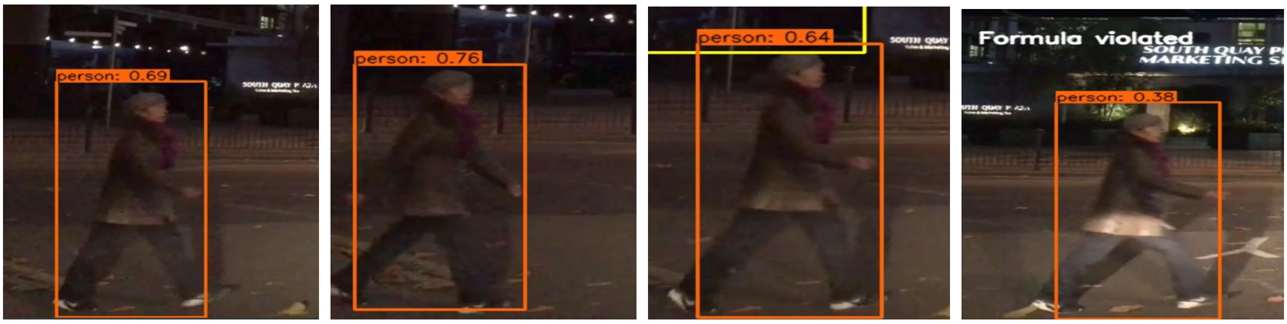

Fig 6. Pedestrian detected but TQTL formula violated

## 6. Conclusion:

The IA-YOLO model has been able to generate satisfactory results on image sequences in foggy and night conditions having appreciably low inference time. This is attributed to the several transformation techniques performed using a set of learned hyperparameters in the DIP Module which aids in capturing the content in the image necessary for the vanilla YOLO model. However, we found some cases where it fails to detect pedestrians in one of the consecutive frames with the desired level of confidence. The robustness of this model has been estimated using TQTL which has been able to correctly verify the required specification in circumstances where multiple objects can be present in a particular frame. This quality metric will help in debugging or improving the existing model and lead to better detection results in foggy and night conditions in a safety-critical context.

## 7. Future Work:

We plan to develop a more robust YOLO model for pedestrian detection in low-light conditions by modifying the CNN Parameter Prediction component so that it learns more optimal parameters required by the DIP module to capture the main content in the foggy and night images. Apart from this, we will try to come up with a few more relevant image processing filters for the DIP module to increase the model efficiency. Moreover, we also aim to augment the current TQTL logic formula by incorporating the distance of a pedestrian from the car camera factor. We can enforce restriction on the distance values in consecutive frames such that the distance does not change by more than some calculated epsilon value, which again can be expressed as a function of the car dynamics corresponding to the previous and current frames.

## 8. References:

[1] A. Balakrishnan, A. G. Puranic, X. Qin, A. Dokhanchi, J. V. Deshmukh, H. B. Amor, G. Fainekos, *Specifying and Evaluating Quality Metrics for Vision-based Perception Systems*, DATE 2019.

[2] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 779-788, doi: 10.1109/CVPR.2016.91.

[3] Polesel, A.; Ramponi, G.; and Mathews, V. J. 2000. Image enhancement via adaptive unsharp masking. IEEE Transactions on Image Processing, 9(3): 505–510.

[4] Yu, Z.; and Bajaj, C. 2004. A fast and adaptive method for image contrast enhancement. In 2004 International Conference on Image Processing, 2004. ICIP'04., volume 2, 1001–1004. IEEE.

[5] Wang, W.; Chen, Z.; Yuan, X.; and Guan, F. 2021. An adaptive weak light image enhancement method. In Twelfth International Conference on Signal Processing Systems, volume 11719, 1171902. International Society for Optics and Photonics.

[6] Zeng, H.; Cai, J.; Li, L.; Cao, Z.; and Zhang, L. 2020. Learning image-adaptive 3D lookup tables for high performance photo enhancement in real-time. IEEE Transactions on Pattern Analysis and Machine Intelligence.

[7] Wenyu Liu, Gaofeng Ren, Runsheng Yu, Shi Guo, Jianke Zhu, Lei Zhang:Image-Adaptive YOLO for Object Detection in Adverse Weather Conditions. AAAI 2022: 1792-1800

[8] Hu, Y.; He, H.; Xu, C.; Wang, B.; and Lin, S. 2018. Exposure: A White-Box Photo Post-Processing Framework. ACM Transactions on Graphics (TOG), 37(2): 26.

[9] He, K.; Sun, J.; and Tang, X. 2009. Single image haze removal using dark channel prior. In Proceedings of IEEE/CVF Conference Computer Vision Pattern Recognition (CVPR).

[10] Redmon, J.; and Farhadi, A. 2018. Yolov3: An incremental improvement. arXiv:1804.02767.

[11] R. Alur and T. A. Henzinger, "A really temporal logic," J. ACM, vol. 41, no. 1, pp. 181–204, 1994.

[12] A. Dokhanchi, B. Hoxha, C. E. Tuncali, and G. Fainekos, "An efficient algorithm for monitoring practical TPTL specifications," in The ACM/IEEE International Conference on Formal Methods and Models for System Design, (MEMOCODE), 2016, pp. 184–193.