International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Relative Depth Prediction of Objects in Hazy Images Using Depthmap and Object Detection

Nandini B. M.*[1], Narasimha Kaulgud[2]

Submitted: 22/08/2023 **Revised**: 12/10/2023 **Accepted**: 23/10/2023

Abstract*:* The ability to establish the relative distances of objects in a single image is essential for many computer vision applications, including scene understanding, augmented reality, and robotics. In this study, we present a method that combines object detection and depth maps to provide an estimate of the relative depth of objects within an image. First, we locate and identify objects within the image using a state-of-the-art object identification model, which yields a set of bounding box coordinates, then estimate the monocular depth maps using a deep learning model. The estimated depth map statistics are used to determine the average depth value of each object enclosed within the bounding box. This data is utilized to estimate the relative distance of objects in the scene. The level of closeness is measured by comparing the average depth value of the objects with a hyper-parameter. An object with average depth value higher than the hyper-parameter is closer to the camera, whereas an object with an average depth value lower than the hyper-parameter is farther away from the camera. We have categorized the relative depths of objects into four levels on the basis of the average depth value's correlation to the hyper-parameter. Experimental evaluations on standard and real-time datasets have shown that the proposed strategy is effective and precise, emphasizing its potential applicability in several computer vision areas.

Keywords: *Deep learning, Dehazing, Depth map estimation, Object detection model, Relative depths.*

1. Introduction

The advancement of image processing and computer vision methods have led to significant progress in various applications, including vehicle navigation. To achieve complete autonomy in driving and navigation, a critical hurdle lies in ensuring dependable and precise obstacle detection. Numerous studies have been dedicated to tackling the challenge of identifying obstacles [1] through object detection and distance prediction techniques. When it comes to object detection, a crucial prerequisite for seamless navigation for the autonomous system is to possess comprehensive information about the objects present in its immediate surroundings. To overcome this barrier, researchers have investigated several methods of integrating object identification algorithms with depth mapping techniques to provide predictions about the relative depth of objects in images. The relative depth of objects refers to the correspondence of their positions in relation to one another based on their distance from the observer. In hazy conditions, the presence of airborne particles and scattering phenomena severely reduces visibility and makes it difficult to perceive the depth information accurately. Consequently, to acquire relative depths of objects in hazy images, it becomes indispensable to dehaze these images as it's a critical requirement for various applications, including autonomous navigation, augmented reality, and scene understanding.

1 Department of Information Science and Engineering, The National Institute of Engineering, Mysuru, Visvesvaraya Technological University, Belagavi, India.
ORCID ID : 0000-0001-8758-0949
2 Department of Electronics and Communication Engineering, The National Institute of Engineering, Mysuru, Visvesvaraya Technological University, Belagavi, India.
ORCID ID : 0000-0002-4019-9179
** Corresponding Author Email: nandinibm@nie.ac.in*

In recent years, object-detection algorithms based on deep-learning concepts, such as Convolutional Neural Networks(CNNs) [2] and region-based methods like Faster R-CNN (Region-based CNN) [3] and YOLO(You Only Look Once) [4], are successful in accurately localizing and identifying objects in images. These methods can not only locate objects within an image, but also provide bounding box coordinates that define their spatial extent within the image. By leveraging these object detection outputs, researchers have explored the possibility of estimating the relative depth of objects in dehazed images.

Another important component in estimating relative depth is the utilization of depth maps. Depth maps represent the scene's geometric structure by assigning a depth value to each pixel, indicating the distance of that pixel from the camera. Traditional depth estimation techniques often rely on stereo vision, or active sensing technologies like LiDAR(Light Detection and Ranging) [5]. Unfortunately, LiDAR sensors tend to be expensive, and many depth cameras face limitations in real-world environments, such as synchronization issues between optical and imaging elements [6]. However, recent advancements in deep learning have led to the development of depth estimation models based on monocular images, allowing for depth map generation from a single image. Deep learning-based monocular depth estimation methods can be by self-supervised [7]-[9] or unsupervised [10]-[12] learning techniques. Supervised methods tend to achieve higher accuracy by utilizing depth-maps for training. However, obtaining accurate depth-maps in real-world scenarios can be challenging. Whereas, unsupervised methods do not rely on original depth-maps for training, but this lack of supervision leads to a slight degradation in performance.

We present a novel framework for estimating the relative depth of objects in hazy images by combining a supervised deep-learning object-detection model with a self-supervised depth estimation

model. By leveraging the synergy between object detection and depth estimation, we aim to contribute to the advancement of image enhancement techniques and enable applications that rely on depth perception in hazy conditions. The key phases of this study are (1) Implementation of the deep learning object-detection model based on YOLOv5 architecture. (2) Implementation of the self-supervised deep-learning model for mono-depth estimation. (3) Combining both models into a cohesive framework to determine the relative distances of objects in the image. In Section 2, We provide an overview of the relevant prior works in the field. Section 3 outlines the methodology we have proposed, detailing our approach. Finally, in Section 4, we present the experiments we conducted and describe the outcomes and results obtained.

2. Background Literature

The most sophisticated techniques and approaches for object-detection and depth-map estimation are thoroughly outlined in this section.

2.1. Object detection Model

Object detection is an essential component of computer vision, and it entails finding and pinpointing specific objects in images or videos. By establishing bounding boxes around objects, the objective is to precisely locate them in addition to identifying their vicinity. Autonomous driving, surveillance systems, image interpretation, robotics, and human-computer interaction are just few of many fields where object detection is vital. It provides a way for machines to understand their surroundings and act responsibly. In recent years, deep learning-based approaches have emerged as the dominant paradigm for object detection. Convolutional Neural Networks (CNNs) [2] are at the core of these methods, which learn hierarchical representations of visual features directly from the data. Notable deep-learning models for object detection include Faster R-CNN [3], YOLO(You Only Look Once) [4], SSD(Single Shot MultiBox Detector) [13], and RetinaNet [14].

The object detection mechanism employed in this work is the YOLO model. The YOLO (You Only Look Once) object detection model is a widely recognized and influential approach in the field of computer vision. Unlike traditional object detection methods that involve multiple stages, such as region proposals and subsequent classification, YOLO takes an unified approach. YOLO takes an input image, grids it, and makes class and bounding box predictions directly for each cell. The key advantages of the YOLO model include its speed and efficiency. YOLO's real-time processing is made possible by its single-pass detection, making it ideal for time-sensitive uses. Several iterations of the YOLO model have been developed, with each version introducing enhancements to improve performance. Notably, YOLOv5 [15] is a recent iteration that focuses on both accuracy and simplicity. It utilizes a two-stage architecture with a lightweight backbone network called CSPDarknet, along with a head network for object detection. YOLOv5 has demonstrated state-of-the-art performance on various benchmark datasets.

2.2. Depth-map Estimation model

By applying a depth-map estimation model to an image or scene, we can learn about the scene's spatial structure and the relative positions of objects. Depth estimation models leverage different techniques and data sources to infer depth information. Some common approaches are as follows:

- Monocular Depth Estimation: This approach estimates depth using a single input image. It is challenging since depth information is inherently ambiguous in a 2D image. Monocular depth estimation models often employ deep learning techniques, such as recurrent neural networks (RNNs) [3], convolutional neural networks (CNNs) [2], or encoder-decoder architectures like U-Net [16].
- Stereo Depth Estimation: Stereo vision involves using a pair of stereo images captured from two horizontally displaced cameras to compute depth information. Stereo depth estimation models [17] leverage the disparities between corresponding pixels in the left and right images to infer depth. They often incorporate techniques like disparity mapping, cost aggregation, and disparity refinement algorithms.
- LiDAR-based Depth Estimation: Light Detection and Ranging (LiDAR) sensors emit laser beams to measure the distances to objects in a scene. LiDAR-based depth estimation models [5] use the point cloud data generated by LiDAR sensors to estimate accurate depth information. These models often involve point cloud processing, voxel-based methods, or deep learning techniques combined with LiDAR data.

For depth estimation in this work, the MIDASNet (Monocular Depth Estimation in Arbitrary Scenes Network) [9] a deep-learning model has been chosen. It is possible to generate a high-resolution depth-map from an input image by using a combination of a multi-scale feature extraction network (encoder network) and a feature upsampling network (decoder network). MIDASNet leverages a pre-trained ResNet-50 backbone as the encoder, which has been trained on a large-scale image classification dataset. The decoder network employs a chain of upsample and skip connections to integrate features from multiple scales, allowing the model to collect both global and local context information for depth estimation. A key aspect of MIDASNet is its training strategy. It employs a self-supervised learning approach, where depth supervision is obtained from the input image itself, without requiring ground truth depth annotations. This is achieved by formulating depth estimation as a relative depth regression problem, using a scale-invariant loss function that enforces consistency between predicted depths for different image regions. MIDASNet has demonstrated impressive results in various challenging scenarios and datasets, demonstrating its depth estimation accuracy in extensive scenarios.

3. The proposed Relative Depth Estimation Methodology

In our approach, we present a methodology that combines object-detection and depthmaps to predict the relative depths of objects present in a dehazed image. Here, the input to our method is a hazy image. The dehazing is done using the "wavelet-based Color Attenuation Prior" method as proposed in [18]. To begin, we use a state-of-the-art object identification model called YOLOv5 to find and precisely locate objects in the dehazed image and extract their bounding box coordinates. Parallely, we leverage MIDASNet, a robust deep-learning network optimized for monochromatic depth estimation. We next calculate the average value of pixels for each object in its mono-depth using the object's bounding box coordinates. With this average pixel value, we are able to estimate the relative distances of the objects. Fig. 1 provides an illustration of the comprehensive framework employed in our method. Below, we break out each stage of the prediction process in detail:

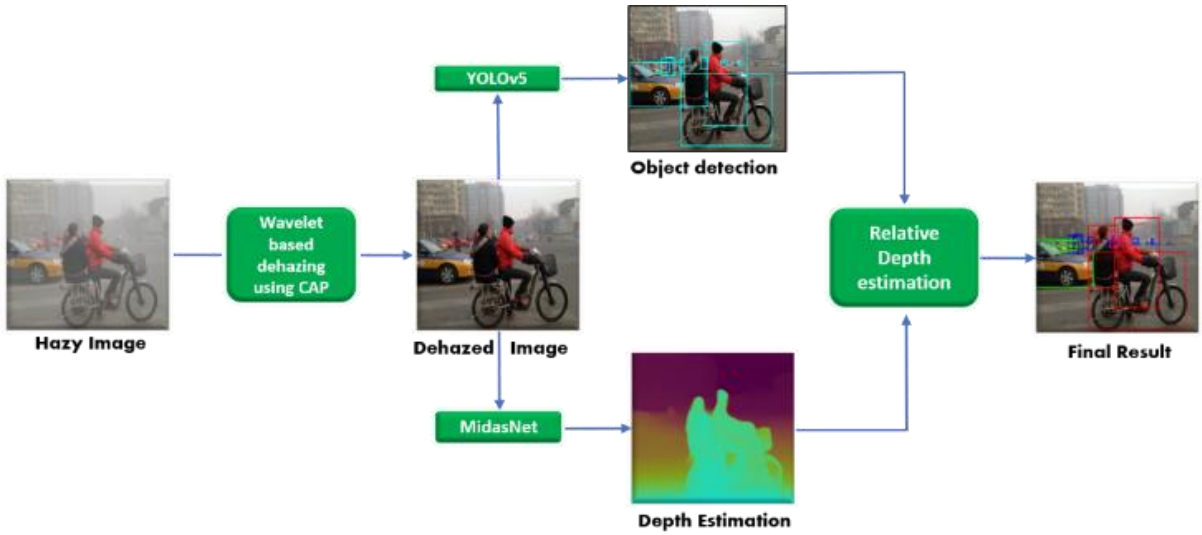


Fig 1. An illustration of the comprehensive framework employed in our method

3.1. Dehazing

Dehazing techniques aim to remove or reduce the effects of haze in an image, restoring its true colors, details, and overall visibility. By enhancing hazy images, dehazing algorithms can improve the interpretability, accuracy, and performance of computer vision systems. They enable better object detection, recognition, tracking, and scene understanding in challenging outdoor environments with haze or fog. An efficient and effective approach for dehazing is through "Wavelet-based dehazing using Color Attenuation Prior(CAP)" [18]. This method involves applying a multilevel discrete wavelet transform [19] to the image, which partitions it into two distinct frequency domains: the low-frequency domain and the high-frequency domain. Following this, dehazing exclusively takes place in the low-frequency domain through the utilization of the Color Attenuation Prior (CAP) [20]. The estimated transmission map is employed to improve texture detail, and a soft-thresholding procedure [21] is implemented in the high-frequency domain to eliminate any leftover noise. Lastly, the two domains are reconstructed to obtain a clear and haze-free image. The flow of dehazing using the former model is illustrated in Fig. 2 [18].

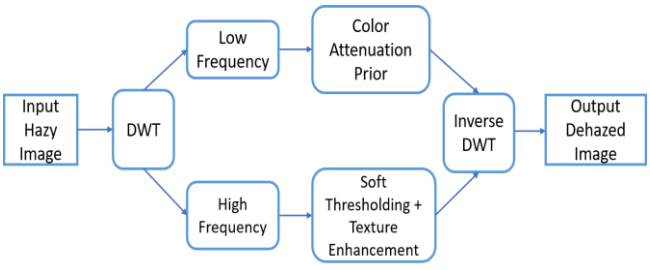


Fig. 2. Wavelet-based dehazing using Color Attenuation Prior

3.2. The Object-Detection process

Object-detection is a sub-field of computer vision concerned with the recognition and localization of certain visual elements. You Only Look Once (YOLO) is an advanced algorithm for detecting objects in real-time. YOLO is popular for object-detection because (1) It is extremely fast and can process images at 45 fps (Frames Per Second) (2) YOLO is accurate with very few background errors (3) provides a better generalization for new domains, hence it is suitable for applications relying on fast and robust object detection (4) Making YOLO open-source has spurred continuous improvement of the model. YOLO architecture [4] is illustrated in Fig. 3, it has overall 24 convolutional layers, 4 max-pooling layers, and 2 fully-connected layers. The following are some of the ways the architecture functions:

- The input image is resized to a dimension of 448x448 prior to being processed by the convolutional network.
- A 7x7 convolution is first used to minimize the channels

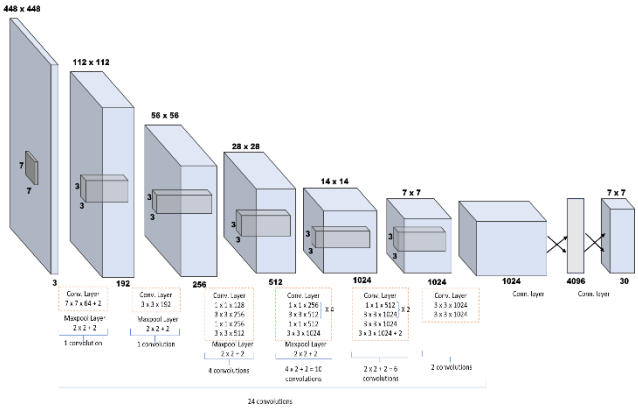


Fig. 3: YOLO Architecture [4]

followed by 3x3 convolution to yield cuboidal output.

- The activation function utilized for all layers is Rectified Linear Units (ReLU) [22], but for the final layer, a linear activation function is employed.
- Additional techniques, such as batch normalization and dropout [23], are employed to regularize the model and prevent overfitting.

In our work, we have used YOLOv5 [15] for object detection. The YOLOv5 architecture shown in Fig. 4 is composed of three primary components:

- Backbone: YOLOv5 uses a feature extraction backbone network called CSPDarknet53 [24] a modified version of the Darknet architecture. The Darknet architecture is built up of

convolutional layers, which take in and extract features of an image.

- Neck: YOLOv5 employs a neck module known as PANet (Path Aggregation Network) [25]. PANet combines data from many scales to improve the model's ability to recognize objects of various sizes. It enhances the network's receptive field and feature representation.
- Head: The head of YOLOv5 consists of linear and convolutional layers that process the information gleaned from the neck. It is responsible for generating bounding-box estimations and class probabilities.

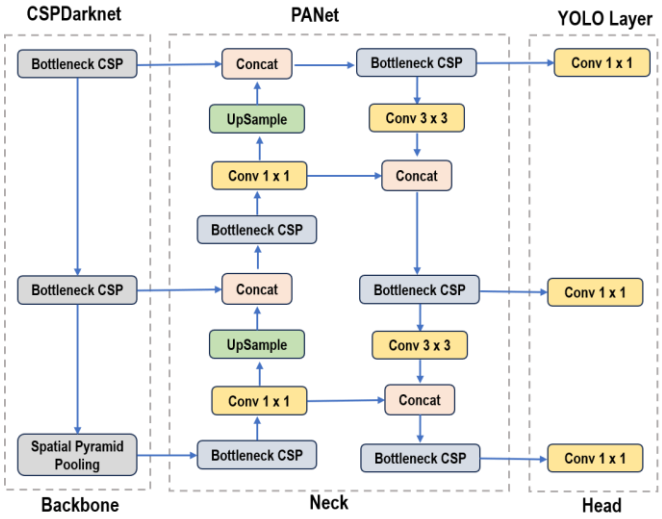


Fig. 4: Overview of YOLOv5 Architecture [15]

YOLOv5 utilizes an anchor-based approach for object detection. It predefines anchor boxes at various scales and ratios, which act as references for predicting object locations and sizes. During training, the model uses these offsets to readjust these anchor boxes. Training YOLOv5 requires only bounding box labels annotated on massive datasets. During inference, the model processes input images by dividing them into a grid and predicts objects within each grid cell. Non-maximum suppression (NMS) [26] is then applied to filter redundant and overlapping detections, retaining only the most confident ones.

3.3. Depth-map Estimation

The technique of determining an object's depth or distance in a scene is known as depth estimation. It entails figuring out how far away the camera or observer is from each pixel or point in a picture. There are different methods to estimate the depthmap, including Time-of-Flight (ToF) depth estimation, stereo depth estimation, and monocular depth estimation. In our study, we have employed the MidasNet [9] a deep learning model to perform monocular depth estimation. A summary of MidasNet architecture for depth-map estimation is illustrated in Fig. 5:

- Encoder: The first step of MidasNet is an encoder module that reads an image as input and extracts features from it. The encoder typically consists of multiple convolutional layers that progressively capture high-level representations.
- Decoder: A decoder module receives the encoded features and upsamples the image to restore it to its native resolution. The decoder utilizes skip connections that connect corresponding feature maps from the encoder to the decoder, aiding in capturing fine-grained details of varied scales.
- Fusion: MidasNet includes a fusion module to merge features from multiple layers together. This fusion helps to incorporate both low-level and high-level information, enabling accurate depth estimation across various spatial scales.
- Monodepth Estimation: For every pixel in the input, MidasNet calculates an estimated depth value to produce a depth-map as the final output. The depthmap provides a per-pixel understanding of the scene's geometric structure and the relative distances of objects from the camera.

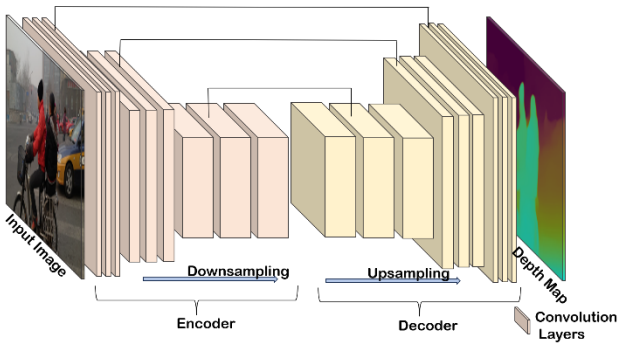


Fig. 5: Overview of MidasNet Architecture

3.4. Relative depth prediction

We employ the bounding box derived from object-detection in conjunction with depth information derived from mono-depth estimate to arrive at a relative depth prediction. The Algorithm 1 outlines the process for determining relative depth through the suggested approach. The relative depth prediction involves:

- Using the bounding box details obtained from object detection, we extract the corresponding depth information from the mono-depth map of the image. Bounding-Box coordinates are given as *(x, y, w, h)*, and *(x, y)* denote the top left corner of the box, or the pixel location from which the box is formed. The horizontal extent of the bounding-box is shown by the value of *w*, while the vertical extent is indicated by the value of *h*.
- By comparing the average depth values of the objects within the bounding boxes, to a fixed reference point, we can determine the relative depth of objects.

$$Mean\_Depth = (1/n) * \sum Depth\_i \quad (1)$$

where $Mean\_Depth$ represents the mean depth value within the bounding-box. n denotes the number of depth values within the bounding-box. $\sum Depth\_i$ represents the sum of all depth values within the bounding-box. The average of $Mean\_Depth$ values of the objects encompassed within the bounding-boxes serves as the reference point.

$$Mean = (1/N) * \sum Mean\_Depth\_i \quad (2)$$

where $Mean$ represents the average of *Mean depths* of detected objects. N is the number of objects detected. $\sum Mean\_Depth\_i$ represents the sum of *Mean depths* of the objects.

- Subsequently, we classify the relative depths of the objects captured in the image into four distinct levels: L1 = VERY CLOSE, L2 = CLOSE, L3 = SAFE DISTANCE, and L4 = FAR AWAY. This categorization is determined by the proximity or distance of the objects with respect to the reference point.

Algorithm 1: Predict the relative depth

Input : The bounding box BB = (x, y, w, h)
N- The number of objects
D- depth map of the input image

Output: Categorize the relative distance.

n = the total number of depth values within the bounding box BB of each object

$Depth\_i$ = the sum of all depth values of i^{th} object within the bounding box BB in the depth map D

for $i \leftarrow 0$ **to** $N-1$ **do**
  $Mean\_Depth\_i = (1/n) * \sum Depth\_i$
end

$Mean = (1/N) * \sum_{i=1}^{N} Mean\_Depth_i$

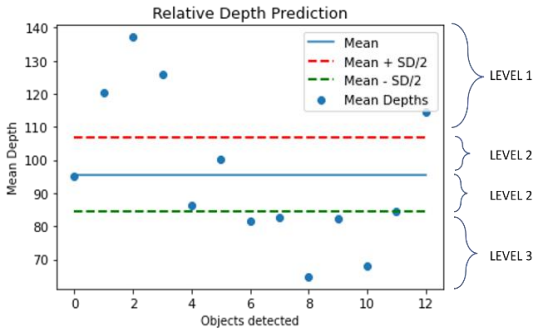
$\sigma = \sqrt{\frac{\sum(Mean\_Depth\_i - (Mean)^2)}{N}}$

for $i \leftarrow 0$ **to** $N-1$ **do**
  $Mean\_Depth\_i = (1/n) * \sum Depth\_i$
  if $Mean\_Depth\_i > Mean + \sigma/2$ **then**
    The object is at Level L1
  end
  if $Mean\_Depth\_i > Mean$ ***AND*** $Mean\_Depth\_i <= Mean + \sigma/2$ **then**
    The object is at Level L2
  end
  if $Mean\_Depth\_i < Mean$ ***AND*** $Mean\_Depth\_i > Mean - \sigma/2$ **then**
    The object is at Level L3
  end
  if $Mean\_Depth\_i < Mean - \sigma/2$ **then**
    The object is at Level L4
  end
end

3.5. Hyper-parameter selection for Relative Depth Prediction

The method is evaluated by estimating the relative depth of objects found in the input image using the RESIDE Dataset [27]. Fig. 6 depicts an example of a prediction. The scatter plot in Fig.6(a) illustrates the mean depths of objects detected in a selected sample image from the RESIDE dataset [27]. The mean depths of the objects change depending on whether they are close to or far from the camera, as seen by the plot. After examining the scatter plots for each image in the RESIDE dataset, we have classified the object's distances into four levels. A level 1 object is one that is relatively close to the camera and is located above the red line. Objects at Level 2 are close enough to the camera, with mean depths between the blue and red lines. Objects with mean depths between the green and blue lines are classified as Level 3. Level 4 objects are the farthest away and are located past the green line. The range for these levels is obtained by comparing the mean depth of each object with the average of the mean depths of all objects and considering the standard deviation [28] of the *mean_depths*. Fig.6(b) displays the statistics of the twelve objects detected in the sample image using the YOLOv5 model, with their bounding-box values utilized to calculate the mean depths in their respective depth maps. Fig.6(c) demonstrates the final outcome of the relative depth prediction, where objects with red-colored boundary boxes correspond to level L1, purple-colored boundary boxes correspond to level L2, green-colored boundary boxes correspond to level L3, and blue-colored boundary boxes correspond to level L4.

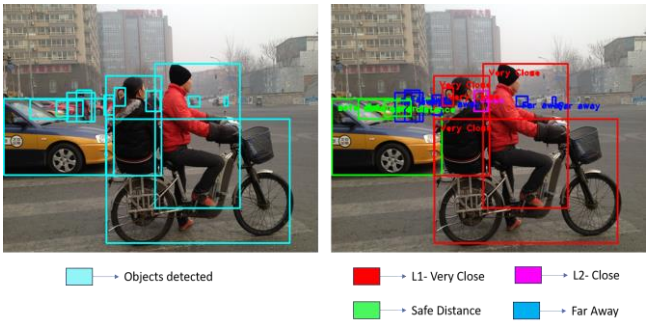
The variability metric, standard deviation [29], is utilized to ascertain the range for each level. This makes it conceivable for us to measure how dispersed a dataset is. The normal distribution or Gaussian distribution [30] with the mean*(μ)* and standard deviation *(σ)* shown in Fig. 7 provides a benchmark for comparing different



(a) Relative depth prediction plot of the Sample figure.

	xmin	ymin	width	height	confidence	class	name	mean
0	1.618149	157.960236	192.157990	127.630707	0.912627	2	car	95.109963
1	265.904175	99.315956	148.834106	240.874268	0.867179	0	person	120.210980
2	180.519394	191.040649	320.696167	206.270294	0.828052	1	bicycle	137.132100
3	179.101089	119.787315	97.698593	166.306244	0.794600	0	person	125.719786
4	47.756557	158.413437	66.488327	33.166794	0.704224	2	car	86.443067
5	249.042938	144.898956	25.691986	34.742065	0.651138	0	person	100.403529
6	176.454666	151.628876	17.147888	33.443756	0.456456	0	person	81.600713
7	128.216003	143.100067	26.020691	52.367828	0.422376	0	person	82.661489
8	386.674835	154.635712	6.334747	15.691727	0.387949	0	person	64.744444
9	111.491768	147.977554	49.229576	48.302963	0.367016	2	car	82.223498
10	323.757477	153.426697	20.719208	17.613754	0.338033	2	car	67.845098
11	93.795670	160.018341	46.920639	26.555435	0.284851	2	car	84.647715
12	194.596573	139.116089	19.649109	30.465637	0.262563	0	person	114.557310

(b) Statistics from the object- detection model YOLOv5



(c) Relative depth prediction using boundary box of object detected

Fig. 6: Sample prediction of relative depths in an image

levels of distribution. Under the bell curve, the area between $\mu \pm \sigma$ is about 68% of the entire area. This area is quite wide to set the relative depth levels. To narrow it down we have chosen $\mu \pm \sigma/2$ with the area reduced to 34%. But narrowing it further at $\mu \pm \sigma/3$

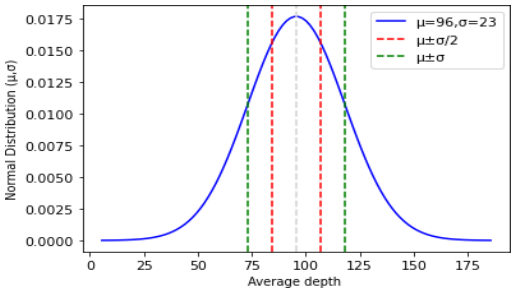
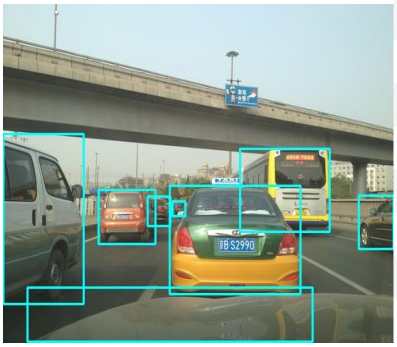


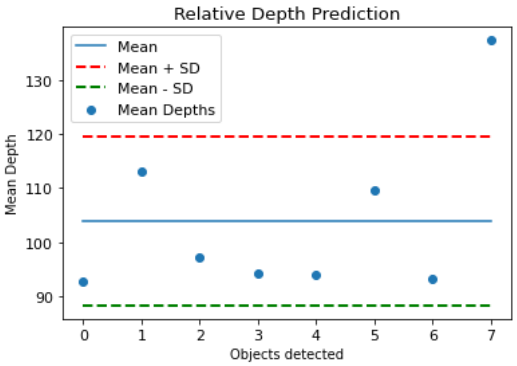
Fig. 7. The normal distribution with μ and σ



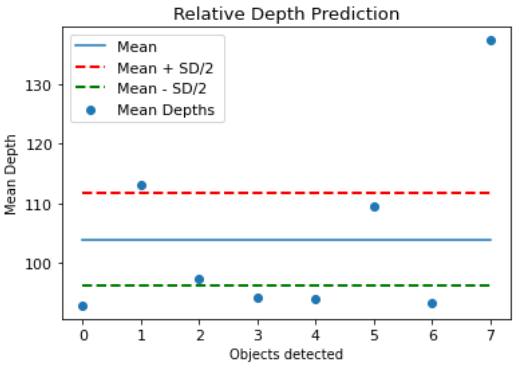
	xmin	ymin	width	height	confidence	class	name	mean
0	333.753296	176.048660	126.124908	101.163864	0.927590	5	bus	92.877207
1	234.615387	221.654236	184.509735	132.552368	0.921663	2	car	113.131272
2	134.955505	226.116013	80.993561	66.388199	0.917044	2	car	97.290215
3	500.614441	233.188721	49.021667	65.227722	0.876619	2	car	94.210989
4	203.835785	234.104691	32.945221	36.657303	0.779325	2	car	93.904803
5	0.604938	158.014877	113.010620	207.795090	0.749210	2	car	109.563009
6	238.301666	240.755600	18.775513	18.247360	0.444463	2	car	93.240741
7	34.952560	345.688965	401.074341	66.548096	0.302840	2	car	137.334202

(a) Object Detection

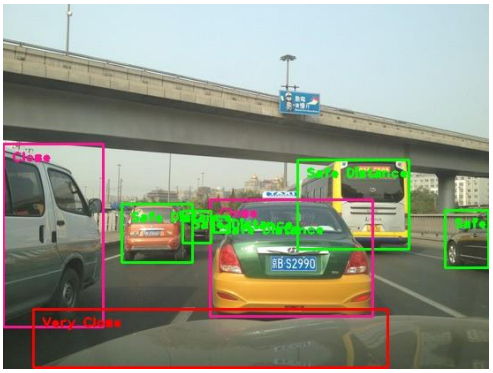
(b) Statistics from YOLOv5 model for Object Detection



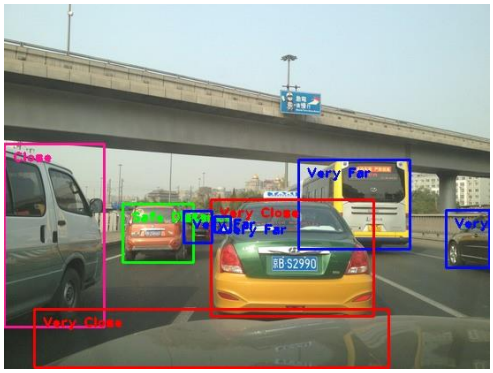
(c) Levels of distribution for σ



(d) levels of distribution for $\sigma/2$



(e) Relative distance for σ



(f) Relative distance for $\sigma/2$

Fig. 8. The relative depth prediction for σ and $\sigma/2$

or $\mu \pm \sigma/4$ will reduce the area under the bell curve drastically to less than 20%, which will be inappropriate to set different levels of relative depths. The relative depth prediction for images from RESIDE dataset was tested with σ and $\sigma/2$ chosen as benchmarks for comparing different levels of distribution. Fig. 8 depicts an example of the obtained data. According to the results, the relative depth prediction is more accurate for the $\sigma/2$ metric than for the metric σ itself. With the σ parameter the objects that are actually closer to the camera are wrongly categorized under L2 (pink boxes) instead of L1 (red boxes) and some objects are categorized under L3 (green boxes) instead of L4 (blue boxes). With $\sigma/2$ parameter the objects are correctly categorized at appropriate levels.

4. Experiment and Results

The Relative Depth Prediction model is implemented in Python. We have implemented MiDasNet to capture the mono-depth of images chosen from RESIDE dataset and from our own dataset. Simultaneously, we have implemented YOLOv5 to capture the boundary box parameter of the objects detected in the test images. To train and evaluate these models, we use the following data sets.

4.1. Dataset

Popular in the field of computer vision, the KITTI dataset [31] is used for tasks like object-detection and depth estimation. The collection has 7481 training images with 3D bounding boxes added to them. In this datadet, 423 for validation, 711 for testing and 6347 images are for training. All input images are 320 pixels wide and 1024 pixels tall.

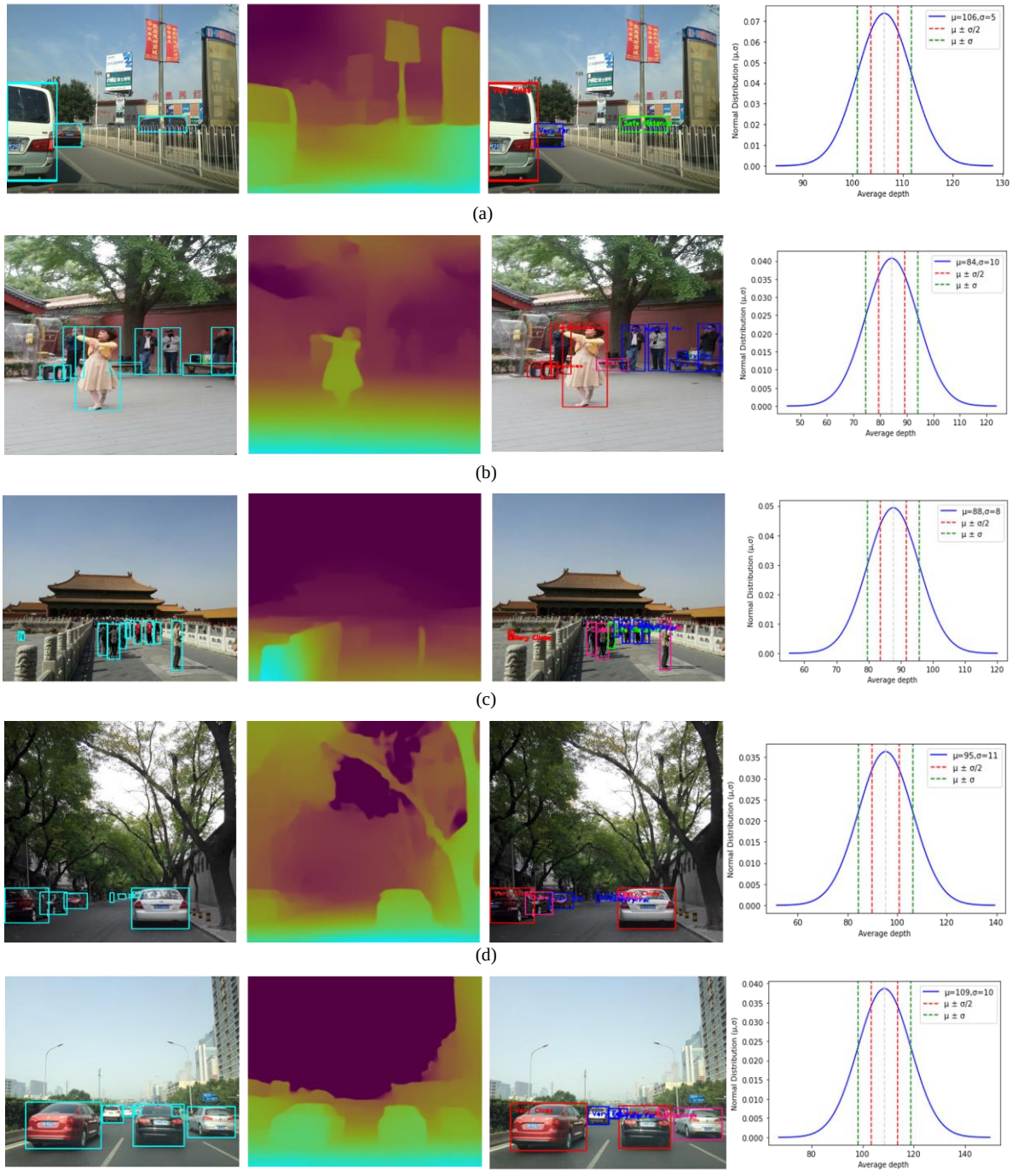
The NYU-Depth V2 dataset [32] comprises video sequences captured within various indoor scenes, using both the RGB and Depth cameras of the Microsoft Kinect device. It has 1449 densely labeled pairs of aligned RGB with depth-map of images and more than four million unlabelled frames. For testing, 10% of the labeled pairs are used, 10% for validation and the remaining for training.

The CocoDataset [33], short for Common Objects in Context Dataset, is a widely recognized dataset for object detection because of its versatility, large-scale annotations, and diverse imagery.The dataset has an incredible 2.5 million annotated instances across 328,000 images, and it contains a diverse collection of 80 different object classes. Using the original partitioning of the dataset, we conducted our research, setting aside 81,434 photos for testing and reserving 81,482 for training.

The REalistic Single Image DEhazing(RESIDE) [34] dataset is an entirely novel, standard dataset comprised of images of both synthetic and actual haze. The Synthetic Objective Testing Set (SOTS) [27] is a part of the RESIDE-Standard dataset consisting of indoor and outdoor subsets of clear and hazy images. There are around 500 clear and hazy image pairs in this subset. We have tested our method on all these images for both dehazing and depth estimation. We have built our own propriety dataset, which consists of 75 images (4032 X 3024 pixels in size) captured in hazy weather conditions at different locations.

4.2. Evaluation

We tested our model on the samples chosen from RESIDE dataset and the private dataset. The objects detected in the dehazed images are classified to the appropriate levels of relative depths. A few sample test results are shown in Fig. 9. In the figure, from left to right we can see the process of the suggested method, starting from object detection, mono-depth estimation, relative depth estimation, and corresponding bell curve of the normal distribution of mean



(a)

(b)

(c)

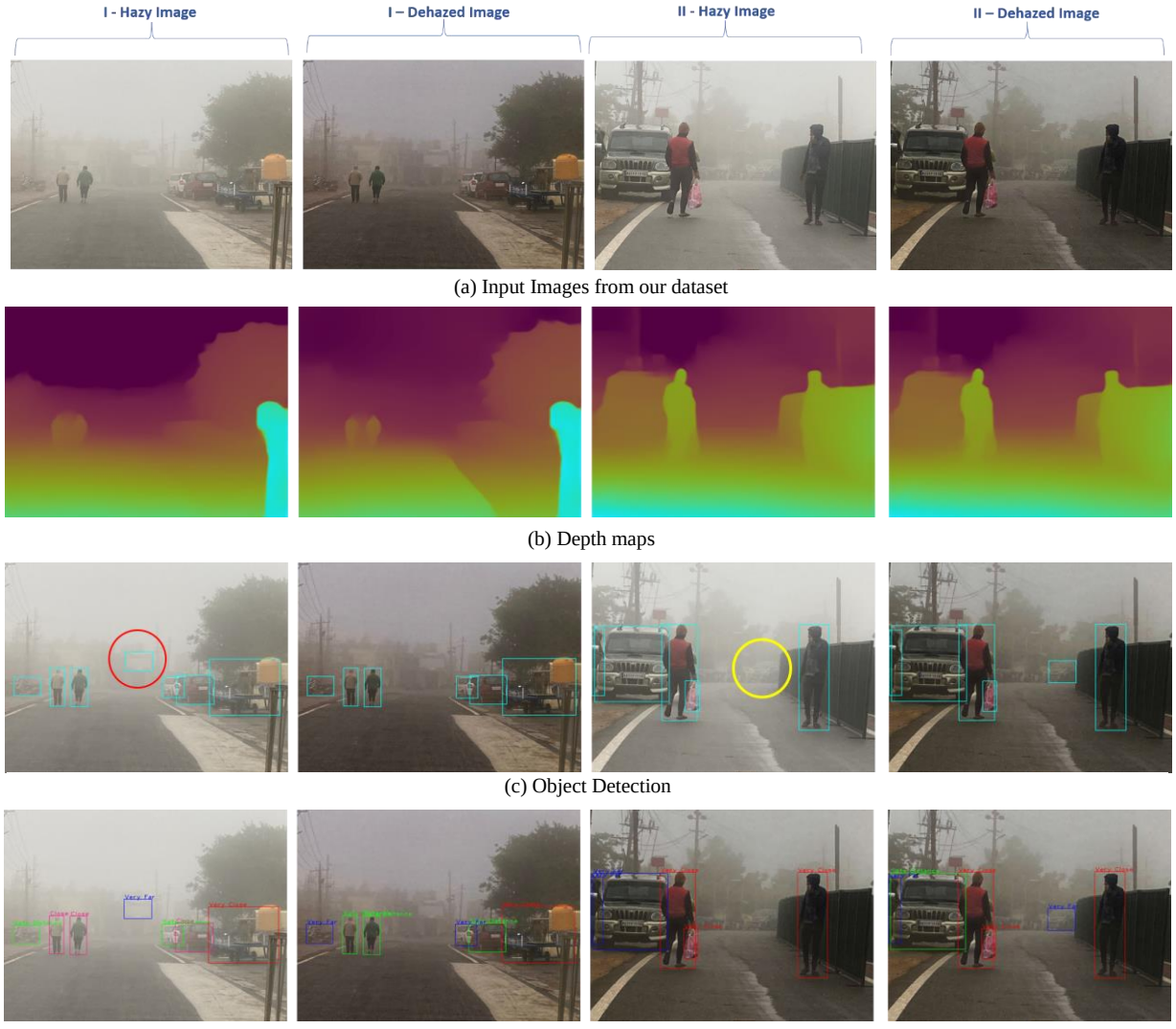
(d)

(e)

Fig. 9: Bell Curve for different images

with the standard distribution. The accuracy of the relative depths predicted in these images can be justified by plotting the shape of the Bell Curve for normal distribution [35] (with the mean and standard deviation) for every image. From the result of experiments conducted on these images, we find that the shape of the Bell Curve for normal distribution for every image is the same, as shown in the last column of Fig. 9. This leads us to infer that the standard deviations or variances of the images are comparable. It can be concluded that the degree of variability or dispersion around the mean is similar for all images. Hence, the categories of levels of relative depths hold good for every image.

Sample test result on hazy and dehazed image pairs chosen from our private dataset is illustrated in Fig. 10. The figure demonstrates that the details of depth in the depthmaps of dehazed images is more detailed than that in the depth-maps of hazy images (as seen in Fig.10(b)). Hence, object detection is more precise in dehazed images than in hazy images. In Fig.10(c), it is evident that in the first sample input hazy image an object is wrongly predicted (circled red), and in the second sample input hazy image the object detection model fails to detect an object (circled yellow), whereas in respective dehazed images, the objects are detected correctly.



(a) Input Images from our dataset

(b) Depth maps

(c) Object Detection

(d) Relative Depth Prediction

Fig. 10. Comparison of Relative Depth prediction on hazy and dehazed images

Also, relative depth prediction in dehazed images is more accurate than that in hazy images. In Fig.10(d), it can be seen that some objects that are at SAFE DISTANCE are wrongly classified as CLOSE and those which are FAR AWAY are classified as SAFE DISTANCE. The accurate prediction of the dehazed images is shown in the 2nd and 3rd columns of Fig.10(d).

5. Conclusion

This study introduces a method that combines object detection and depth maps to predict the relative depth relationships between objects in a single image. By accurately detecting and localizing objects using an advanced object detection model, bounding box coordinates are obtained. Depth information is acquired through depth-maps generated either by monocular depth estimation techniques or by dehazing the im age. These depth maps provide per-pixel depth values, allowing for the prediction of relative depth between objects. By comparing the depth values within the bounding boxes, the method infers the relative distances between the objects. The proposed method demonstrates its effectiveness and accuracy through evaluations of benchmark datasets. It provides a practical approach for estimating relative depth relationships in computer vision applications such as scene understanding, augmented reality, and robotics. The integration of object identification models and depth-maps could be useful in many different areas of computer vision.

Acknowledgements

This research was supported by the research center at Electronics and Communication Engineering department, The National Institute of Engineering, Mysuru, India. We thank our colleagues from the departments of Electronics & Communication Engineering and Information Science & Engineering, who provided insight and expertise that greatly assisted the research. We thank Doctoral Committee members for their valuable comments that greatly improved the quality of the research work.

Author contributions

Both the authors have contributed equally for the research work.

Conflicts of interest

The authors declare no conflicts of interest.

References

[1] P. Buddi, C. V. K. N. S. N. Moorthy, N. Venkateswarulu, and M. Bala, "Exploration of issues, challenges and latest developments in autonomous cars," Journal of Big Data, vol. 10, 05 2023.

[2] K. O'Shea and R. Nash, "An introduction to convolutional neural networks," CoRR, vol. abs/1511.08458, 2015. [Online]. Available: http://arxiv.org/abs/1511.08458.

[3] S. Ren, K. He, R. B. Girshick, and J. Sun, "Faster R- CNN: towards real-time object detection with region proposal networks," CoRR, vol. abs/1506.01497, 2015. [Online]. Available: http://arxiv.org/abs/1506.01497.

[4] J. Redmon, S. K. Divvala, R. B. Girshick, and Farhadi, "You only look once: Unified, real-time object detection," CoRR, vol. abs/1506.02640, 2015. [Online]. Available: http://arxiv.org/abs/1506.02640.

[5] N. Mehendale and S. Neoge, "Review on lidar technology, " SSRN Electronic Journal, 01 2020.

[6] H. G. Olanrewaju and W. O. Popoola, "Effect of synchronization error on optical spatial modulation," IEEE Transactions on Communications, vol. 65, no. 12, pp. 5362–5374, 2017.

[7] S. Pillai, R. Ambrus¸, and A. Gaidon, "Superdepth: Self-supervised, super-resolved monocular depth estimation," in 2019 International Con- ference on Robotics and Automation (ICRA), 2019, pp. 9250–9256.

[8] A. Kumar, S. Bhandarkar, and M. Prasad, "Depthnet: A recurrent neural network architecture for monocular depth prediction," 06 2018, pp. 396– 3968.

[9] R. Ranftl, K. Lasinger, D. Hafner, K. Schindler, and V. Koltun, "Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 44, no. 3, pp. 1623–1637, 2022.

[10] C. Godard, O. Aodha, and G. Brostow, "Unsupervised monocular depth estimation with left-right consistency," 07 2017.

[11] M. Poggi, F. Aleotti, F. Tosi, and S. Mattoccia, "Towards real-time unsupervised monocular depth estimation on cpu," 06 2018.

[12] V. Repala and S. R. Dubey, "Dual cnn models for unsupervised monoc- ular depth estimation," 04 2018.

[13] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in Computer Vision – ECCV 2016, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Cham: Springer International Publishing, 2016, pp. 21–37.

[14] T. Lin, P. Goyal, R. B. Girshick, K. He, and P. Dolla´r, "Focal loss for dense object detection," CoRR, vol. abs/1708.02002, 2017. [Online]. Available: http://arxiv.org/abs/1708.02002

[15] G. Jocher, "Yolov5 by ultralytics," 2020. [Online]. Available: https://github.com/ultralytics/yolov5

[16] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," CoRR, vol. abs/1505.04597, 2015. [Online]. Available: http://arxiv.org/abs/1505.04597

[17] H. Laga, L. V. Jospin, F. Boussa¨ıd, and M. Bennamoun, "A survey on deep learning techniques for stereo-based depth estimation," CoRR, vol. abs/2006.02535, 2020. [Online]. Available: https://arxiv.org/abs/2006.02535

[18] N. B. M and N. Kaulgud, "Wavelet-based method for enhancing the visibility of hazy images using color attenuation prior," in 2023 Interna- tional Conference on Recent Trends in Electronics and Communication (ICRTEC), 2023, pp. 1–6.

[19] Y. Chen, "An introduction to wavelet analysis with applications to image and jpeg 2000," in 2022 4th International Conference on Intelligent Medicine and Image Processing, ser. IMIP 2022. New York, NY, USA: Association for Computing Machinery, 2022, p. 49–57. [Online]. Available: https://doi.org/10.1145/3524086.3524094.

[20] Q. Zhu, J. Mai, and L. Shao, "A fast single image haze removal algorithm using color attenuation prior," IEEE Transactions on Image Processing, vol. 24, no. 11, pp. 3522–3533, 2015.

[21] Y. Zhang, W. Ding, Z. Pan, and J. Qin, "Improved wavelet threshold for image de-noising," Frontiers in Neuroscience, vol.13, 2019. [Online]. Available: https://www.frontiersin.org/articles/10.3389/fnin

s.2019.0009

[22] A. F. Agarap, "Deep learning using rectified linear units (relu)," CoRR, vol. abs/1803.08375, 2018. [Online]. Available: http://arxiv.org/abs/1803.08375.

[23] C. Garbin, X. Zhu, and O. Marques, "Dropout vs. batch normalization: an empirical study of their impact to deep learning," Multimedia Tools and Applications, vol. 79, pp. 1–39, 05 2020.

[24] C. Wang, H. M. Liao, I. Yeh, Y. Wu, P. Chen, and J. Hsieh, "Cspnet: A new backbone that can enhance learning capability of CNN," CoRR, vol. abs/1911.11929, 2019. [Online]. Available: http://arxiv.org/abs/1911.11929

[25] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," CoRR, vol. abs/1803.01534, 2018. [Online]. Available: http://arxiv.org/abs/1803.01534

[26] Y. Song, Q.-K. Pan, L. Gao, and B. Zhang, "Improved non-maximum suppression for object detection using harmony search algorithm," Applied Soft Computing, vol. 81, p. 105478, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1568494619302480

[27] B. Li, W. Ren, D. Fu, D. Tao, D. Feng, W. Zeng, and Z. Wang, "Benchmarking single-image dehazing and beyond," IEEE Transactions on Image Processing, vol. 28, no. 1, pp. 492–505, 2019.

[28] S. H. K. M., Standard Deviation. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 1378–1379. [Online]. Available: https://doi.org/10.1007/978-3-642-04898-2 535.

[29] C. Andrade, "Understanding the difference between standard deviation and standard error of the mean, and knowing when to use which," Indian Journal of Psychological Medicine, vol. 42, no. 4, pp. 409–410, 2020, pMID: 33402813. [Online]. Available: https://doi.org/10.1177/0253717620933419

[30] J. Shkak and H. Hassan, "Characteristics of normal distribution," 12 2020.

[31] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," International Journal of Robotics Research (IJRR), 2013.

[32] P. K. Nathan Silberman, Derek Hoiem and R. Fergus, "Indoor segmen- tation and support inference from rgbd images," in ECCV, 2012.

[33] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dolla´r, and C. L. Zitnick, "Microsoft coco: Common objects in con- text," in Computer Vision – ECCV 2014, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Cham: Springer International Publishing, 2014, pp. 740–755.

[34] B. Li, W. Ren, D. Fu, D. Tao, D. Feng, W. Zeng, and Z. Wang, "Reside: A benchmark for single image dehazing," 12 2017.

[35] C. Tsokos and R. Wooten, "Chapter 7 - normal probability," in The Joy of Finite Mathematics, C. Tsokos and R. Wooten, Eds. Boston: Academic Press, 2016, pp. 231–263. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780128029671000073