**Journal Pre-proof**



LiDAR-based estimation of bounding box coordinates using Gaussian process regression and particle swarm optimization

Vinodha K, E.S. Gopi, Tushar Agnibhoj

PII: S2667-3797(23)00054-2

DOI: <https://doi.org/10.1016/j.birob.2023.100140> Reference: BIROB 100140

To appear in: *Biomimetic Intelligence and Robotics*

Received date : 4 August 2023

Revised date : 20 November 2023

Accepted date : 23 November 2023

Please cite this article as: V. K, E.S. Gopi and T. Agnibhoj, LiDAR-based estimation of bounding box coordinates using Gaussian process regression and particle swarm optimization, *Biomimetic Intelligence and Robotics* (2023), doi: <https://doi.org/10.1016/j.birob.2023.100140>.

This is a PDF file of an article that has undergone enhancements after acceptance, such as the addition of a cover page and metadata, and formatting for readability, but it is not yet the definitive version of record. This version will undergo additional copyediting, typesetting and review before it is published in its final form, but we are providing this version to give early visibility of the article. Please note that, during the production process, errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

© 2023 The Author(s). Published by Elsevier B.V. on behalf of Shandong University. This is an open access article under the CC BY-NC-ND license ([http://creativecommons.org/licenses/by-nc-](http://creativecommons.org/licenses/by-nc-nd/4.0/) [nd/4.0/](http://creativecommons.org/licenses/by-nc-nd/4.0/)).

LiDAR-based estimation of bounding box coordinates using Gaussian process regression and particle swarm optimization

VINODHA K a, E. S. GOPI a, TUSHAR AGNIBHOJ a

*aDepartment of Electronics and Communication Engineering National Institute of Technology Tiruchirappalli*

*Tamil Nadu - 620015 India*

*Email addresses -* [*vinodhakamaraj@gmail.com*](mailto:vinodhakamaraj@gmail.com)

## Abstract

Camera-based object tracking systems in a given closed environment lack pri- vacy and confidentiality. In this study, light detection and ranging (LiDAR) was applied to track objects similar to the camera tracking in a closed en- vironment, guaranteeing privacy and confidentiality. The primary objective was to demonstrate the efficacy of the proposed technique through carefully designed experiments conducted using two scenarios. In Scenario I, the study illustrates the capability of the proposed technique to detect the locations of multiple objects positioned on a flat surface, achieved by analyzing LiDAR data collected from several locations within the closed environment. Scenario II demonstrates the effectiveness of the proposed technique in detecting mul- tiple objects using LiDAR data obtained from a single, fixed location. Real- time experiments are conducted with human subjects navigating predefined paths. Three individuals move within an environment, while LiDAR, fixed at the center, dynamically tracks and identifies their locations at multiple instances. Results demonstrate that a single, strategically positioned LiDAR

*Preprint submitted to Biomimetic Intelligence and Robotics November 25, 2023*

can adeptly detect objects in motion around it. Furthermore, this study pro- vides a comparison of various regression techniques for predicting bounding box coordinates. Gaussian process regression (GPR), combined with parti- cle swarm optimization (PSO) for prediction, achieves the lowest prediction mean square error of all the regression techniques examined at 0.01. Hyper- parameter tuning of GPR using PSO significantly minimizes the regression error. Results of the experiment pave the way for its extension to various real-time applications such as crowd management in malls, surveillance sys- tems, and various Internet of Things scenarios.

*Keywords:* LiDAR, data acquisition, bounding box, gaussian process regression, particle swarm optimization (PSO)

## Introduction

In the computer vision application landscapes, the foundational task of object detection has long relied upon cameras as the primary sensing modal- ity (Lvwen et al., 2017; Drazen et al., 2013). However, persistent challenges,

5 such as sensitivity to lighting variations and privacy concerns, have fueled the exploration of alternative methodologies. This study supports a transforma- tive shift from the traditional use of cameras by proposing the integration of two-dimensional (2D) light detection and ranging (LiDAR) sensors for object detection, a shift underscored by notable advancements in sensor technology

10 and the proliferation of deep learning methodologies (Alejandra et al., 2016; MahmudulHasan et al., 2022; Wu et al., 2021; Demetri et al., 2019; Benedek et al., 2021).

* 1. *Research background*

Research builds upon an evolving environment, highlighting the signifi-

15 cance of LiDAR usage and the fusion of deep learning techniques in object and human detection across diverse environments. LiDAR-based inspection, exemplified by its efficacy in identifying flaws in bridge construction, en- hances operational efficiency and mitigates the risks associated with the presence of humans in high-risk locations (Liu et al., 2017). Techniques

20 involving beam-steering and scene flashing provide valuable insights into op- tical modulation, LiDAR designs, and detection methods, broadening the LiDAR application spectrum (Behroozpour et al., 2017).

Regarding mobile robot navigation, prior attempts utilizing support vec- tor machine classification demonstrated commendable accuracy in indoor ob-

25 ject detection (Alejandra et al., 2016; Yao et al., 2023). However, challenges persist in the precise location of objects within complex scenes, fueling the need for advancements in segmentation methods (Wang et al., 2021). De- spite strides in these methodologies, acquiring pinpoint accuracy in object location remains an ongoing challenge. Comparative studies of 2D and 3D

30 LiDAR data for pedestrian detection demonstrate the comparable or superior results obtained with the 2D LiDAR data (Bu et al., 2020; Sohn et al., 2021). Combining 2D LiDAR with camera data to estimate the indoor environment layouts, leveraging motion cues, and optimization algorithms demonstrates the versatility of this sensing modality (Li and Stevenson., 2021). Moreover,

35 machine-learning algorithms applied to 2D LiDAR data analysis proved ef- fective in fall detection and activities of daily living monitoring, emphasizing the adaptability of this technology (Miawarni et al., 2020).

* 1. *Research significance*

Despite these advancements, challenges persist in the existing LiDAR

40 data processing methods related to computational complexity and privacy. In this study, this gap is addressed by introducing an innovative approach centered on estimating object coordinates. This approach involves using camera target values derived from LiDAR data while focusing on error min- imization and reducing reliance on visible light. The potential impact of

45 this research extends beyond computer vision, with the potential to trans- form surveillance, crowd management, and various IoT applications. The proposed methodology could disrupt the surveillance landscape, providing a compelling alternative to traditional CCTV cameras. Meanwhile, it en- sures privacy in object detection applications and enhances accuracy under

50 challenging lighting conditions, positioning this research as a motivator in reshaping surveillance technologies.

The indoor LiDAR-based object detection determines the target points by tracking specific features or objects located in the LiDAR point cloud data. One unexplored application is the estimation of object coordinates

55 using the camera-derived target values, suggesting a new research direction and facilitating the replacement of CCTV cameras with LiDAR for surveil- lance purposes. An experimental dataset including LiDAR data and camera- derived coordinates is provided for public use. During training, the system uses camera data to establish target points; however, during testing, the sys-

60 tem operates independently of cameras, relying solely on LiDAR data. This methodology is unique and significant in advancing LiDAR-based surveil- lance technologies.

* 1. *Overview of the proposed work*

To our knowledge, no attempts have been made to estimate object posi-

65 tions using the 2D LiDAR by estimating camera target values for determin- ing object coordinates. This study demonstrated that Gaussian process re- gression (GPR) with hyperparameters tuned by particle swarm optimization (PSO) gives better results than state-of-the-art techniques, such as kernel smoothing regression and convolution neural networks (CNN), in lidar ap-

70 plications. In GPR, hyperparameters can be tuned using various methods, including manual tuning, grid search, random search, Bayesian optimiza- tion, maximum likelihood estimation (Pintea et al., 2016), and PSO. In this study, PSO is adopted to tune hyperparameters in LiDAR-based applications and minimize object detection errors in closed environments. In this work,

75 demonstrating 2D LiDAR, data are collected with 360° coverage and the object position at various locations is identified using regression techniques

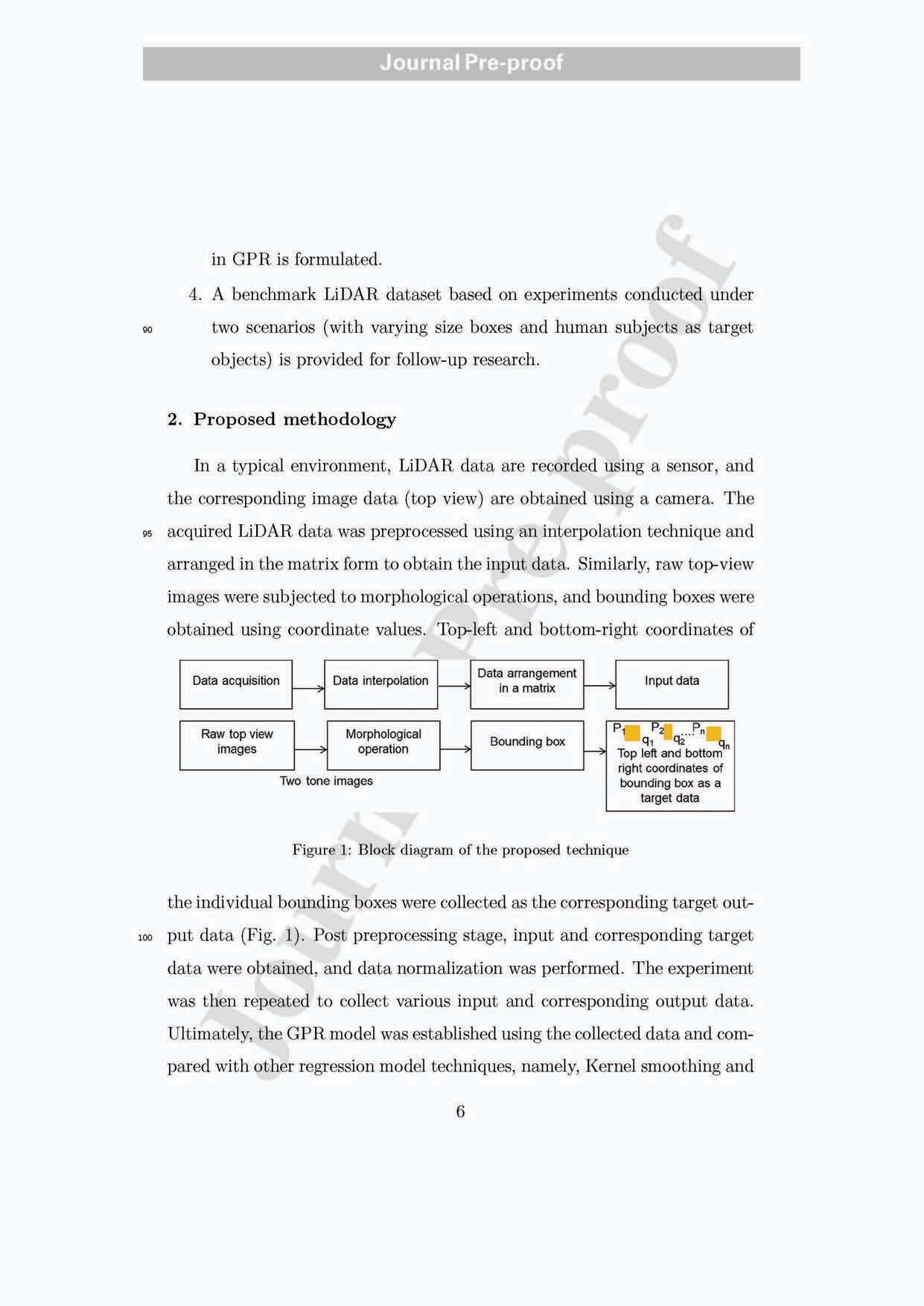
with an optimization algorithm. Data preprocessing is conducted, applying interpolation of the LiDAR data and morphological operations on the binary image data.

80 The main contributions of this study are as follows:

* + 1. The usage of 2D LiDAR data to estimate the coordinates of the bound- ing boxes of multiple objects obtained from a digital camera in closed environment scenarios is demonstrated as a proof of concept.
    2. A methodology for predicting the bounding box coordinates using GPR

85 techniques is established and compared with other regression tech- niques.

* + 1. An objective function applying PSO to tune the hyperparameters used



# Journal Pre-proof

105 CNN.

*2.1. LiDAR sensor description*

110

115

120

125

LiDAR is a remote sensing technology that uses lasers to measure the distance of objects in a scene. LiDAR sensors emit laser pulses and measure the time required for the pulses to return after bouncing off objects in the environment. In this study, a YDLIDAR X4 sensor was used to collect object data at different locations. It rotated 360° to scan and range the surrounding

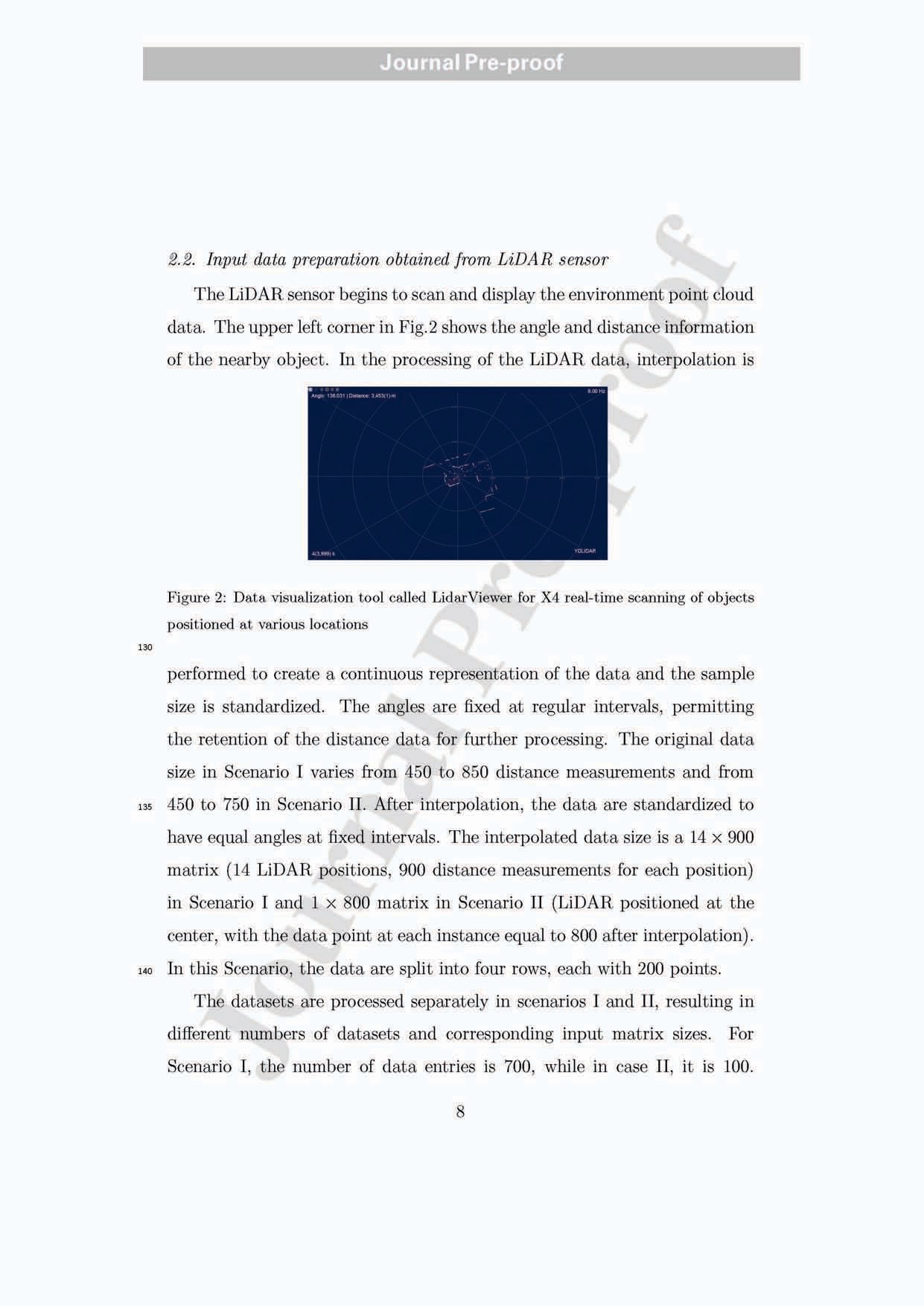
environment, producing a map outline. The YDLIDAR X4 frequency range

was 5*,* 000*Hz* with 6-12*Hz* scan frequency, and the *ϕ* size was 65*.*6 *×* 58*.*39 *×* 101*.*7*mm*. The resolution angle and distance range were 0*.*43–0*.*860 and 0*.*12– 10*m*, respectively. YDLIDAR provides a point cloud viewer (LidarViewer V0.1.9), a visualization program for X4 real-time scanning (2).

According to the product description for the YDLIDAR X4 sensor, the ideal working environment is indoors, and light in the indoor environment does not affect the performance of the device, confirmed by experiments con-

ducted with the test data in real-time with various light intensities (LUX *→*

luminous), such as 312, 161, and 26 LUX. Furthermore, it was demonstrated to detect objects effectively, even in a dark room. Thus, YDLIDAR X4 can function regardless of ambient light conditions; it does not rely on the pres- ence of visible light to operate, making it suitable for a wide range of lighting conditions. To validate the proposed technique, the research experiment is performed under two different scenarios in an indoor environment.



# Journal Pre-proof

145

150

These standardized and processed data matrices, each representing a single LiDAR dataset with distance measurements at fixed-angle intervals, can now be used for further analysis and modeling. Depending on the specific goals, they can be applied to various statistical or machine-learning techniques to gain insights or make predictions. The uniformity in the data size and interpolation step could facilitate the use of the data and enable consistent analysis across different datasets.

*2.3. Target data preparation obtained from mobile camera*

The coordinate points of the bounding box (Top-left and bottom-right corners]) corresponding to the individual objects in the captured image data are obtained (as described in Fig. 3). The RGB image format is converted to

155 HSV, and a matrix (with 0*.*12 *< H <* 0*.*18) is obtained for further operation.

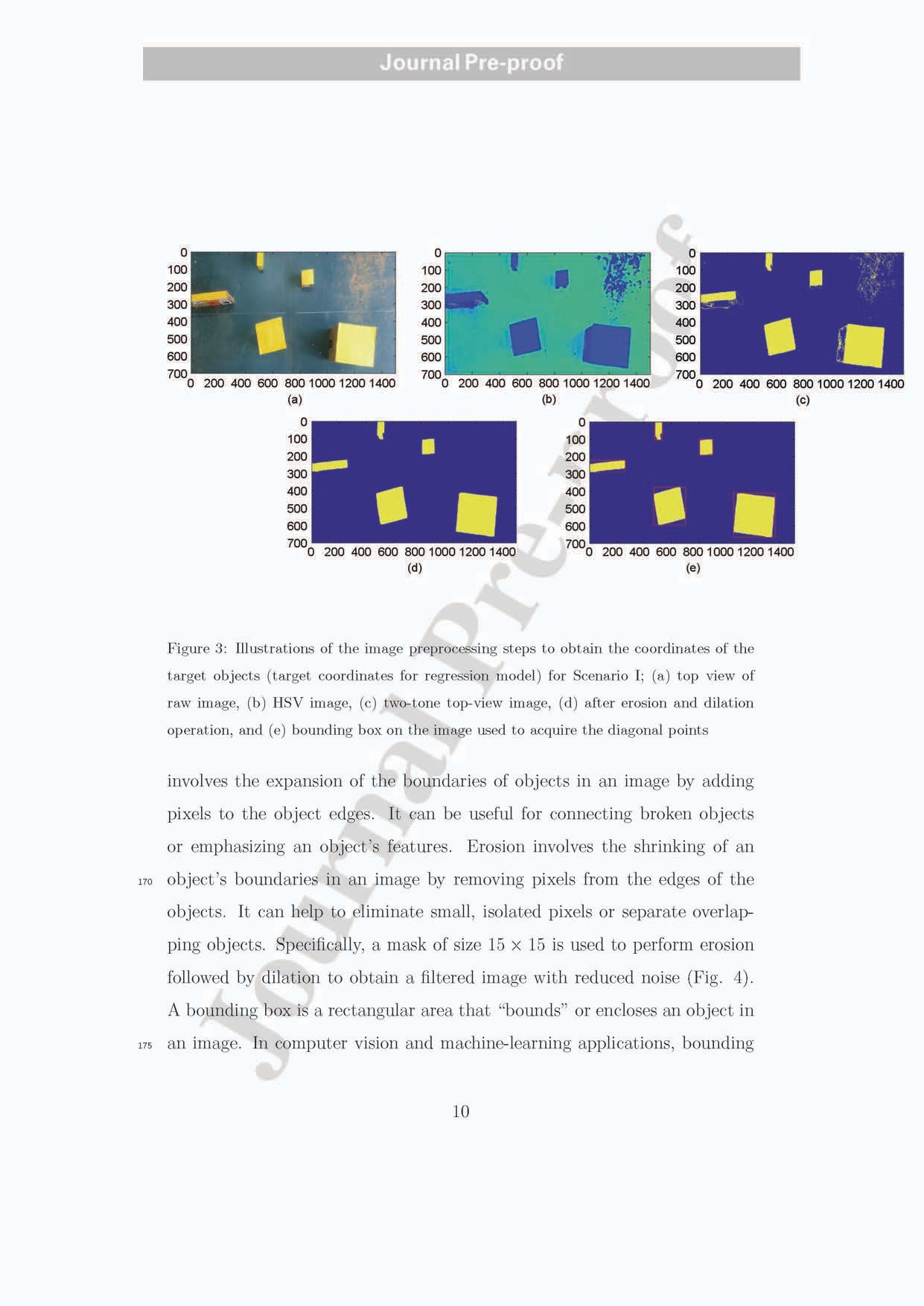
The combinations for Scenario II are as follows:

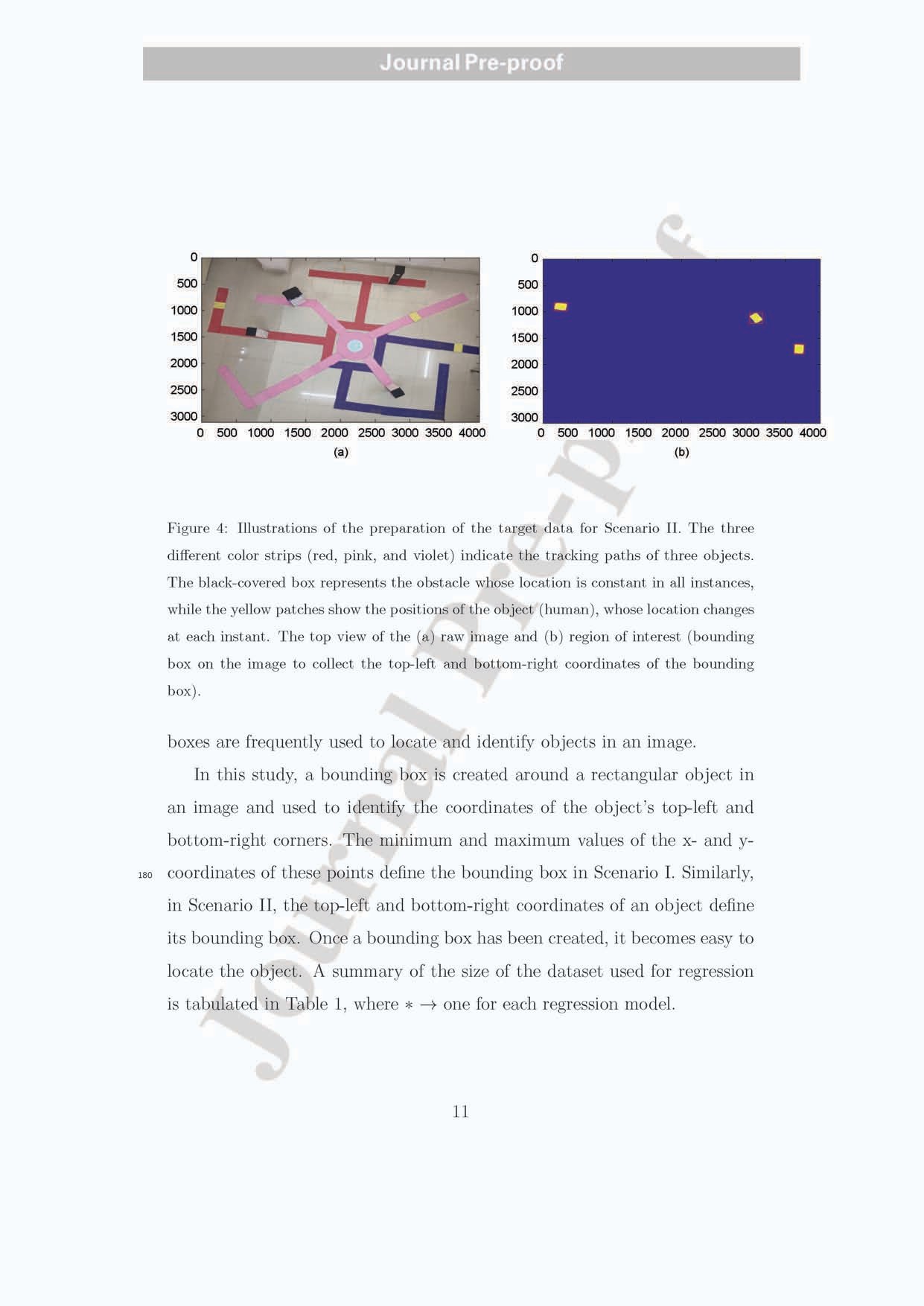
* ((r*>*155) & (r*<*210)) & ((g*>*135) & (g*<*190)) & ((b*>*60) & (b*<*120))
* ((r*>*190) & (r*<*230)) & ((g*>*150) & (g*<*210)) & ((b*>*95) & (b*<*150))

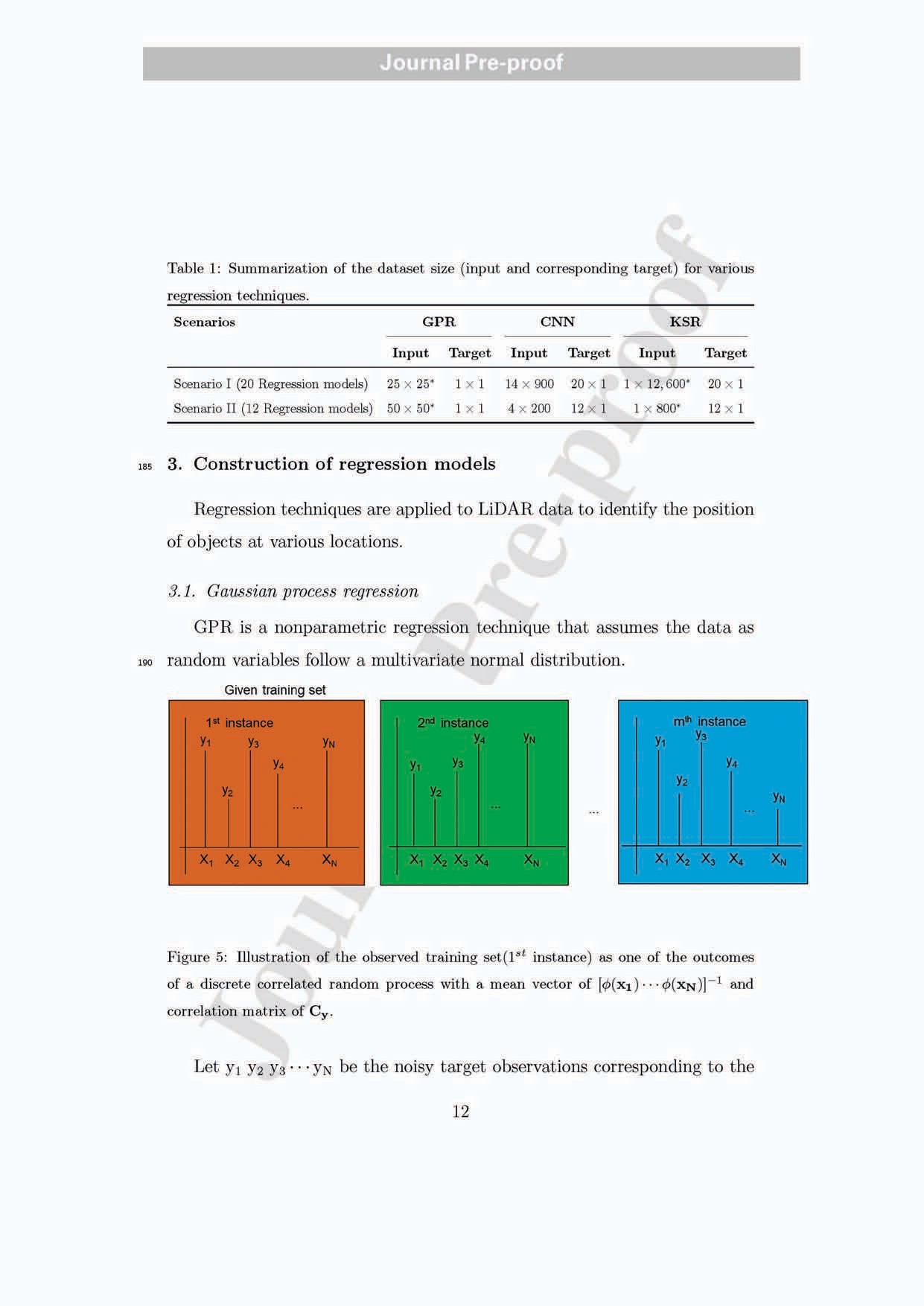
160 • ((r*>*180) & (r*<*230)) & ((g*>*180) & (g*<*230)) & ((b*>*85) & (b*<*160))

The morphological processes of erosion and dilation are performed on the image to remove the unwanted details of the object and, more precisely, re- trieve the area of interest (target object). Morphological operations modify the shape or structure of objects in an image. The procedure involves en-

165 larging or contracting the boundaries of the objects in an image to isolate particular features or eliminate noise or other undesirable elements. Dilation







195

200

input vectors **x1 x2 x3** *· · ·* **xN** for N instances. y(**x**) = *ϕ*(**x**) + *ϵ* where, **x**

*→*input vector, y *→*corresponding noisy observation, *ϵ →* Gaussian-distributed

random variable with a mean zero and variance 1 , and *ϕ*(**x**) is the nonlinear basis function. This training dataset is the typical outcome of the correlated discrete random process (Fig.5). The predicted value corresponding to the test input vector (**xtest**) is obtained as the conditional mean of the random variable ytest, conditioned on a typical outcome of the random process (train-

*λ*

ing dataset). The joint density function of random vector [y1 y2 y3 *· · ·* yN ytest]

can be shown to follow a multivariate Gaussian distribution with mean zero and covariance matrix C.

y = [y1 y2 *. . .* yN]*T* (1)



C = 

**Cy ktest**

**T**



1  (2)

where,

**ktest** k(**xtest***,* **xtest**) + *λ*

 k(**x1***,* **x1**) + 1 *· · · k*(**x1***,* **xN**) 

*λ*

 1 . 

Cy = 

k(**x2***,* **x1**) k(**x2***,* **x2**) + *λ*

.

.

(3)

.  

 k(**xN***,* **x1**) k(**xN***,* **xN**) + 1 

*λ*



**ktest** = 

k(**x1***,* **xtest**) 

. 

(4)

 k(**xN***,* **xtest**) 

and k(**xi***,* **xj**) =*ϕ*(**xi***T* )*ϕ*(**xj**). It can be demonstrated that the conditional den-

205 sity function, denoted by the notation f(ytest*/***x1 x2 x3**. . . **xN** y1 y2 y3 *. . .* yN **xtest**),

has a Gaussian distribution with a mean of

yˆtest = **kT**

(**Cy**)*−***1**y*,* (5)

j **test**

and variance *k*(**xtest***,* **xtest**)+ 1 *−***kT** (**Cy**)*−***1ktest**. Furthermore, Cy = K + 1 I,

*λ* **test** *λ*

where K = *ϕT ϕ* denotes the kernel smoothing matrix, and **k**(**xtest***,* **xi**) denotes the kernel function *ϕ*(**xtest**)*T ϕ*(**xi**). *k*(**xm***,* **xn**) represents the (m*,* n)th element

210 of the kernel smoothing function. Below is presented the generated kernel function.

2

**i**

k(**xi***,* **xk**

) = *θ*0

*e−θ*1

(**xi***−***xk**)2  +*θ*2+*θ*3(**xTxk**) (6)

where the hyperparameters are *θ*0,*θ*1,*θ*2,*θ*3, optimized using PSO. The steps in GPR in our proposed work are shown in algorithm 1.

**Algorithm 1** Gaussian process regression

## Require:

**x1 x2** *. . .* **xN** *▷* Train data

y1 y2 *. . .* yN

**xtest xtest** *. . .* **xtest**

*▷* Test data

**1 2 M**

ytest ytest *. . .* ytest

1 2 M

**Ensure:** *θ*0*, θ*1*, θ*2*, θ*3*, λ ▷* Find *θ*0, *θ*1, *θ*2, *θ*3 such that J is minimized

**function** (k(**xi***,* **xk**

) = *θ*0

*e−θ*1

(**xi***−***xk**)2  +*θ*2+*θ*3(**xTxk**))

Compute the covariance matrix Cy using the training data for the particular *θ*0, *θ*1, *θ*2, *θ*3 (Eq. 3)

2

**i**

**for** *j →* 1 to *M* **do**

Compute ktest corresponding to **xtest** (Eq. 4)

**j**

yˆtest *→* **kT**

(**Cy**)*−***1**y *▷* Compute the predicted value

yˆtest

j

## end for end function

**test** j

PSO is a heuristic optimization algorithm used to find the global min-

215 imum of a given objective function (Kennedy and Eberhart., 1995). The algorithm iteratively updates a population of particles in a search space, in which each particle represents a potential solution to the problem (Kennedy and Eberhart., 1995; Gopi., 2020). The algorithm 2 combines individual and collective learning to move the particles toward the global minimum.

**Algorithm 2** Particle swarm optimization with fitness function

1: Load input and target data

2: Load the best values and center point

3: Initialize the particle population

4: Set the maximum number of iterations and convergence threshold

5: Initialize the global and local best positions and fitness values

6: **while** the number of iterations is less than the maximum allowed and the convergence criteria are not met **do**

7: **for** each particle in the population **do**

8: Update the velocity and position of the particle

9: Evaluate the fitness function using the new position 10: **if** the fitness value is better than the local best **then** 11: Update the local best position and fitness value 12: **end if**

13: **if** the fitness value is better than the global best **then** 14: Update the global best position and fitness value 15: **end if**

16: **end for**

17: **if** the difference between the previous and current global best fitness values is less than the convergence threshold **then**

18: Terminate the algorithm

19: **end if**

20: **end while**

21: Return the global best position and fitness value

220 The probability that the individual test data *ytest*, generated from the

*j*

multivariate Gaussian density function model, depends upon the covariance

matrix (*cy*) and mean vector *y*ˆ*test*. These are constructed using a kernel

*j*

225

function that depends upon the hyperparameters *θ*0 *θ*1 *θ*2 *θ*3. The function J1 (Eq. 7) is formulated as the product of the probability density functional values of all the individual test data. Thus, the minimization problem is constructed using the fitness function J=  1 as follows.

1

*J*

*J* = Y 1 *e−* 1 [*ytest−y*ˆ*test*]*T c−*1[*ytest−y*ˆ*test*] (7)

*M*

1

*j*=1

2 *j j*

(2*π*)*M/*2*|cy|*

*y j j*

*J* = arg

*θ*0 *θ*1 *θ*2 *θ*3

1

*min*

*J*1

(8)

*3.2. Regression model using kernel smoothing method*

230

235

240

The algorithm 3 shows the procedure for the kernel smoothing routine used in this study. LiDAR data coordinate estimation of objects is obtained

by applying the kernel smoothing technique. The 1 *×* m input data corre-

sponding to tn target points are obtained from the diagonal coordinates of the bounding box image. The dataset is divided into 50% for training and the remaining for testing. Of the testing data, 25% are for validation and 25% reserved for testing. Initially, the Gaussian kernel function with the training and testing data within a specified range of sigma from 0*.*01 to 50 is used to retrieve each target point. Each target point is obtained by regres- sively adding the target to the result of the kernel function and the target specified for the training data. According to the kernel smoothing algorithm, each point is calculated from the validation data, with the *σ* value set to 0.8. Finally, the predicted target points and average error are determined using the test data.

k=Ntest

Σ

N

tk

*—* Σ K(x *,* x )t T tk

*—* Σ K(x *,* x )t (9)

k=1

test

i=1

i k i

test

i=1

i k i

where xi stands for the training set, ti for the matching target, xk for the test

vector, tk for the estimated target, and Ntest for the number of test data.

test

**Algorithm 3** Kernel smoothing

## Require:

x1*,* x2 *. . .* xN *▷* training data

t1*,* t2 *. . .* tN

xtest*,* xtest *. . .* xtest

*▷* test data

1 2 M

ttest*,* ttest *. . .* ttest

1 2 M

## Ensure: *σ*

**function** (k(xj*,* xtest*, σ*))

*−*(xj*−*xtest)T(xj*−*xtest)

K(xj*,* xtest) *→* e

*e →* 0

**for** *j →* 1 to *M* **do**

*s →* 0

**for** *i →* 1 to *N* **do**

*σ*2 *▷* Gaussian kernel function

*s →* s + K(xtrain*,* xtest*, σ*) *×* ttrain

N

i j i

## end for

*e → e* + (s *−* ttest)2 *▷* e *→* Error

j

## end for

Find *σ* such that e is minimized

## end function

245 *3.3. Regression model using convolution neural networks*

250

255

260

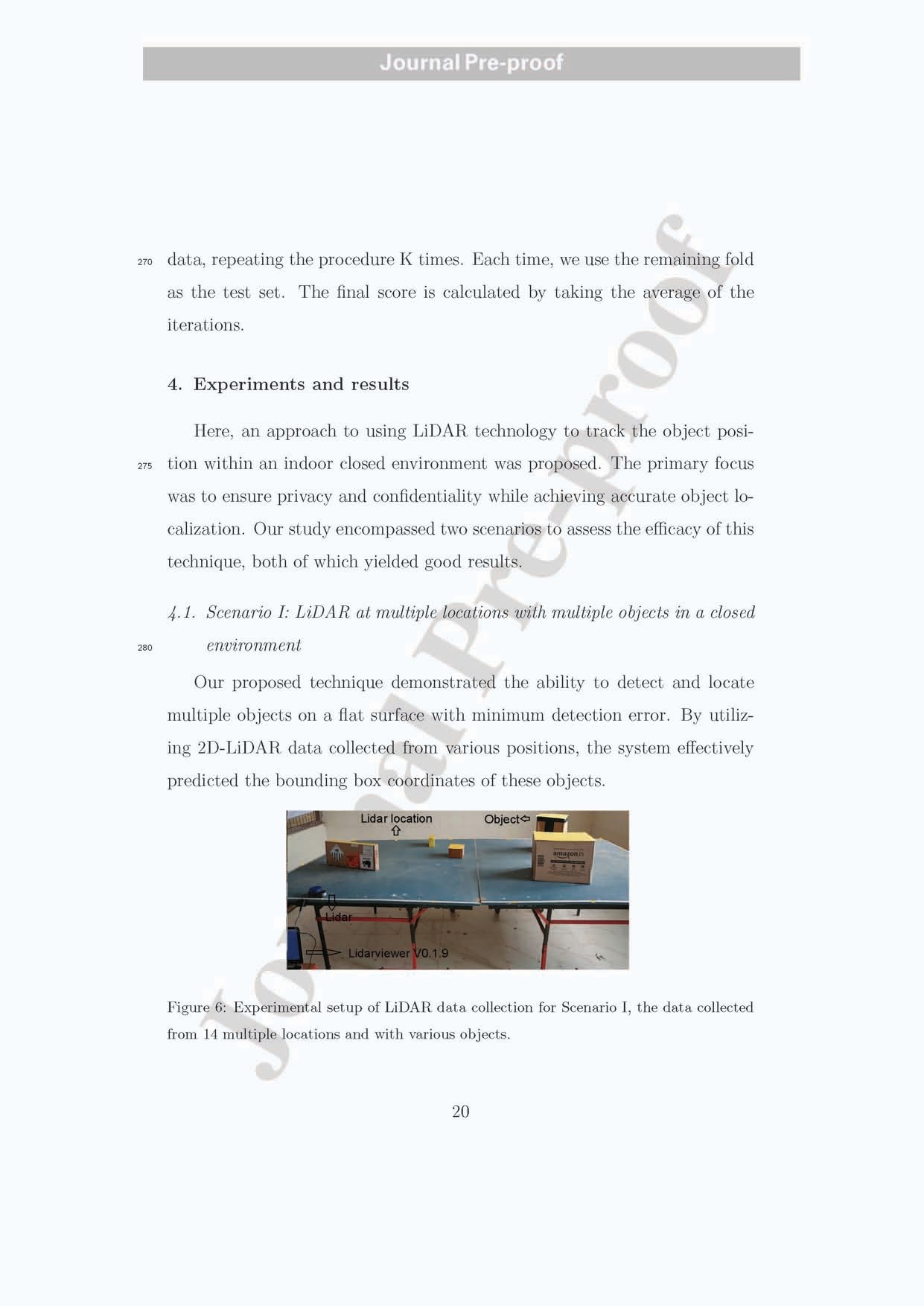
265

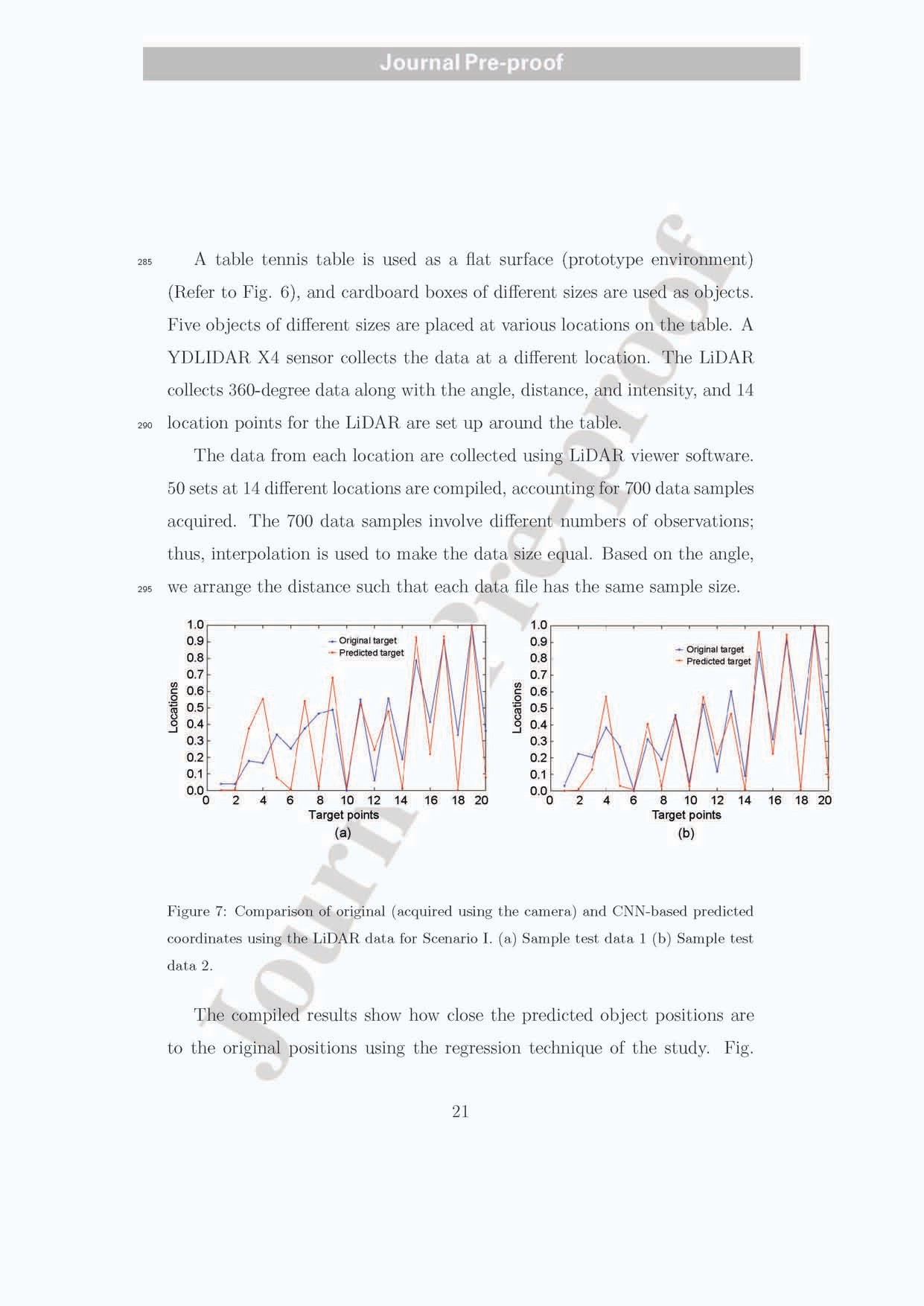
A deep learning model named CNN regression 1 is employed to fore- cast continuous-valued outputs, also known as “regressions.” CNN regression models use convolutional filters through many layers to extract features from the input data. However, a CNN regression model generates a continuous- valued output representing a prediction or estimate of the desired quantity,

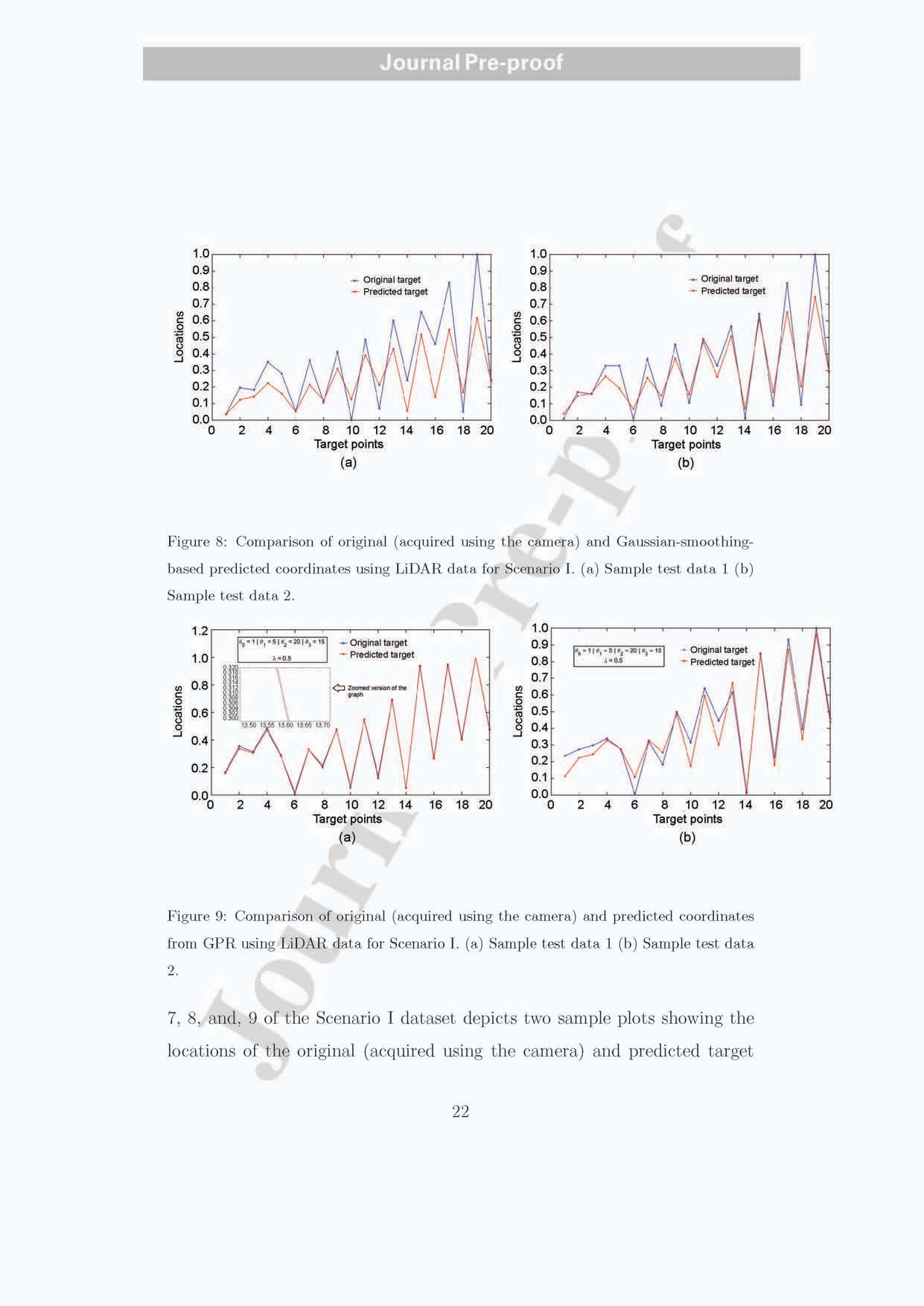
rather than a class label or probability, as the final result. The network ar- chitecture is created with input data of n *×* m *×* 1 (n is the number of object locations) and a target of tn(number of x- and y-coordinate points of all the

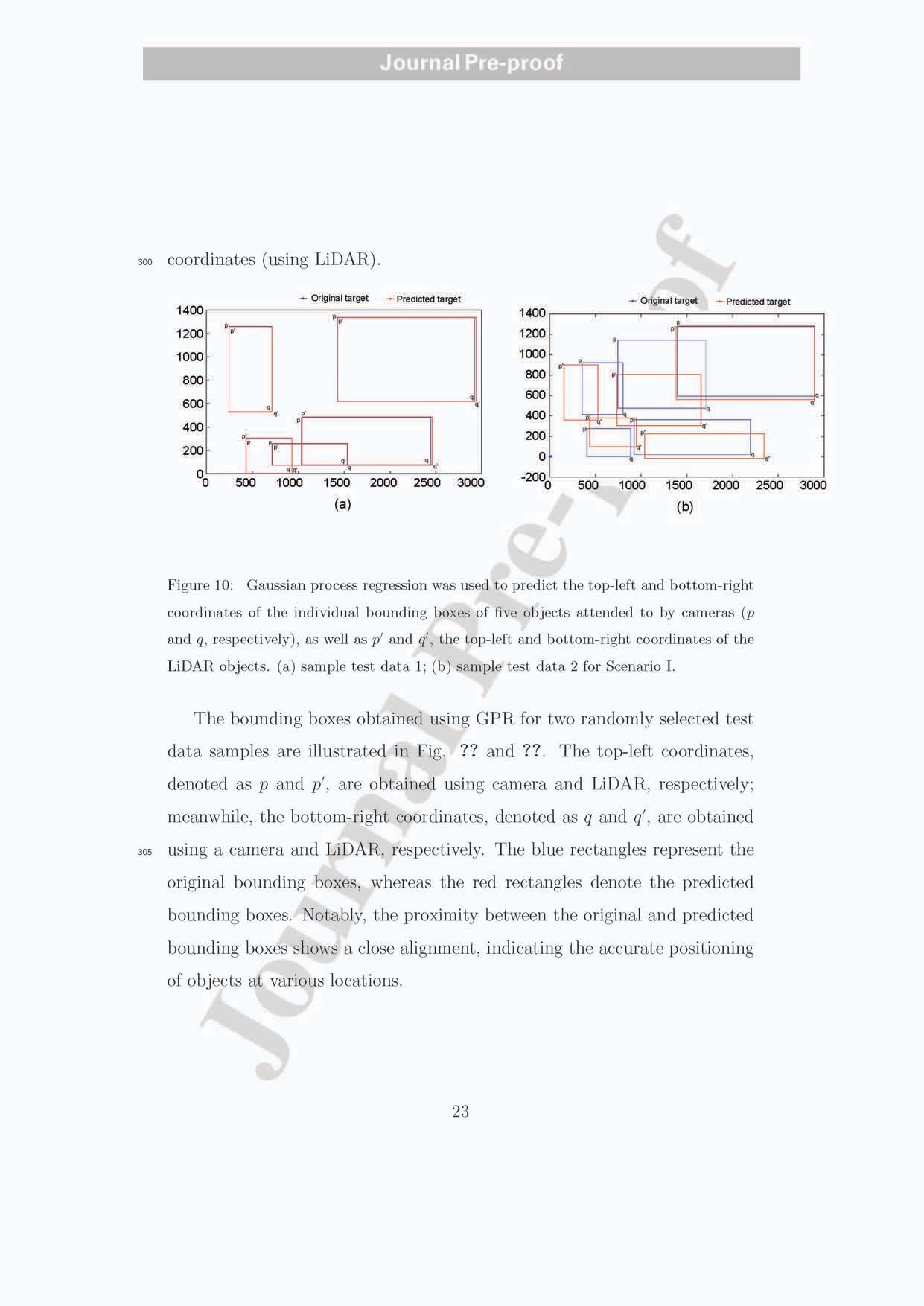
objects placed at one instant) points. The number of convolutional layers used to learn the input data is three. Convolutional filters are applied to the input data during the processing stage, with each filter activating different features from the data sample (Miao et al., 2016). A “Relu layer” is used as an activation function, providing quicker and more effective training with positive values kept and negative values converted to zero. Then, pooling is performed to reduce the number of parameters needed by the network to learn through nonlinear downsampling of the output. A sigmoid layer is fi- nally used at the end of the network before the regression layer to convert into a positive values.

K-fold cross-validation is used to validate the CNN model. The number of folds K is set to 5. K - 1 folds are chosen as the training dataset, and the rest as the test dataset. From “m” data samples, 50% are used to train the model as training data and the remaining are used to validate the model as test data. On each cross-validation iteration, we train a new model independent of the model trained on the previous iteration and then validate it with the test









# Journal Pre-proof

310

315

320

325

330

*4.2. Scenario II: LiDAR at the center point (single location) with multiple objects in a closed environment*

The performance of the technique in detecting multiple objects using Li- DAR data from a single, fixed location was evaluated. The experiments involved the real-time tracking of human subjects within the LiDAR’s obser- vation area. The results showed that a strategically positioned single LiDAR device could successfully detect moving objects nearby.

An environmental path was set in a room where three people moved around a track. In Fig. 11, three different colored strips (red, pink, and violet) indicate the walking track that the humans use to move around the path. A black-covered object indicates an object whose position is the same in all instances. The lidar is fixed at the center, collecting 360° data. Data were

collected for 14 instances (14 different times), and eight different positions of

a moving object were collected at each instant.

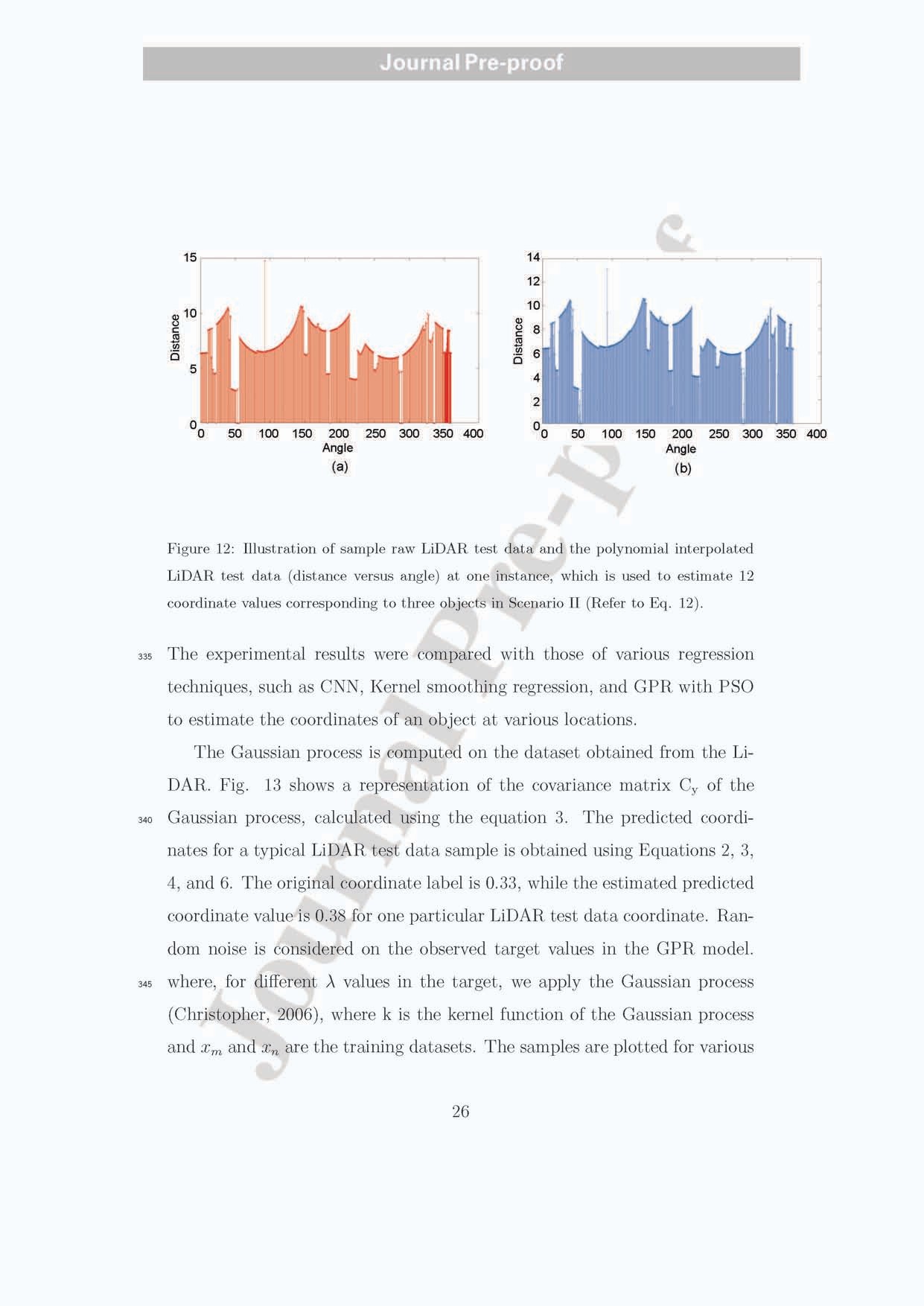
Polynomial interpolation is used to obtain the length of the LiDAR data, identical to several attempts. The target points are collected for both envi- ronments in a closed space by mounting the Android mobile phone camera onto the top view. The camera captures the target image of multiple objects at different locations.

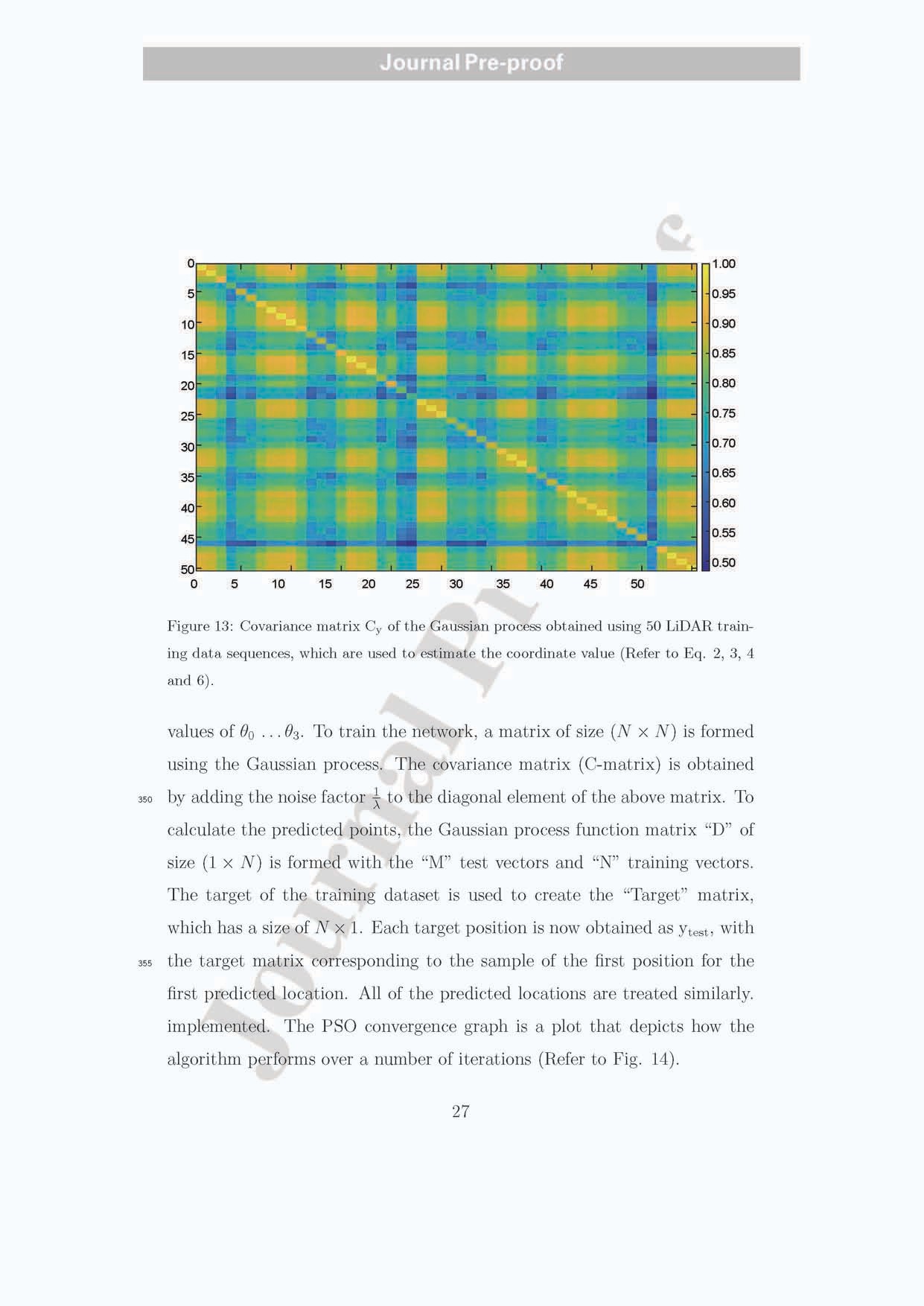
The data are distributed as training and testing sets in a ratio of 5:5, such that the training data includes 50 sets and the testing data comprise the remaining sets. A sample of the raw data and interpolated LiDAR data are shown in Fig. 12.

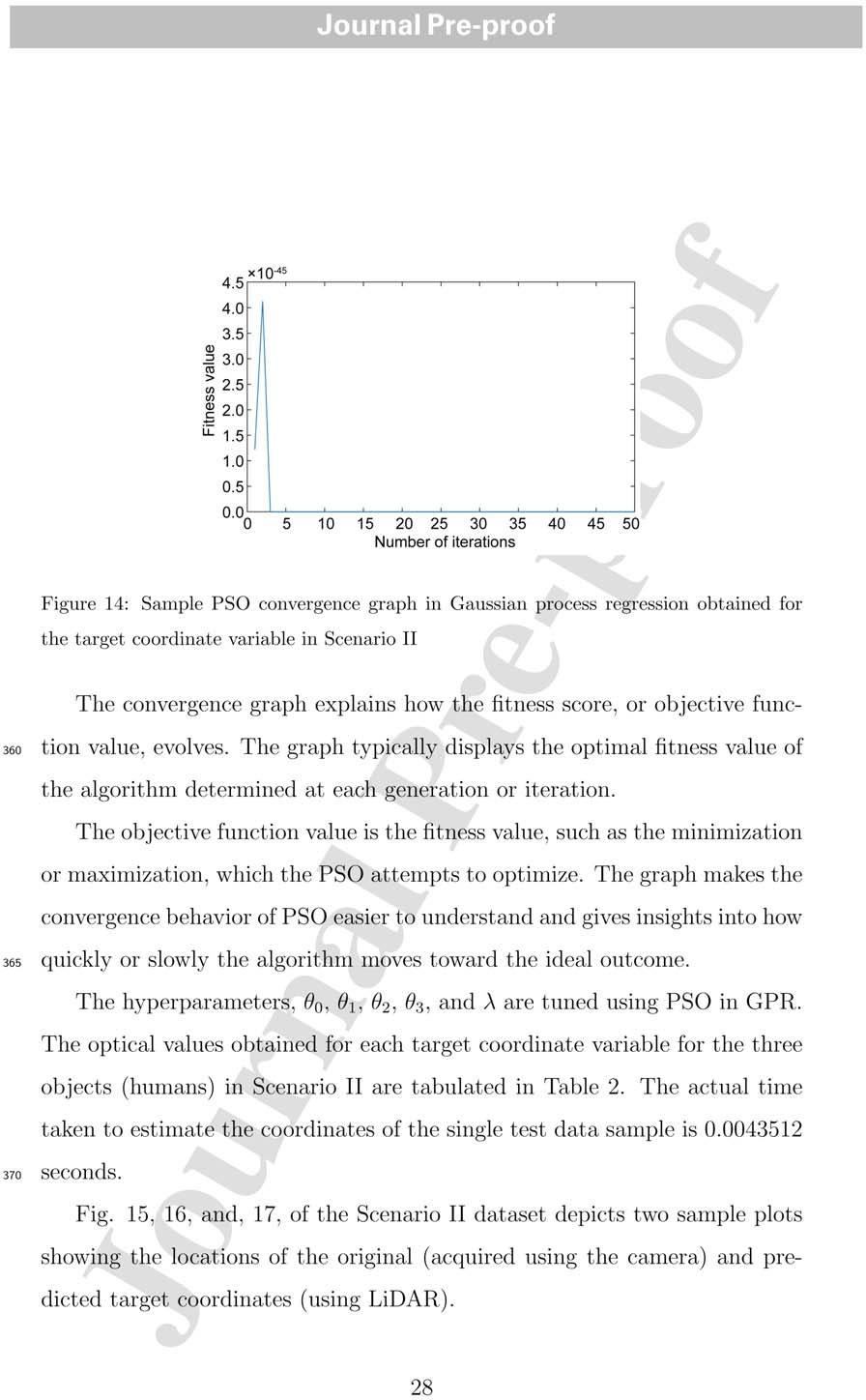
Furthermore, our research focused on the fine-tuning of the GPR hyper- parameters using PSO, resulting in optimized regression error minimization.

24









# Journal Pre-proof

Table 2: Tuned hyperparameter obtained using particle swarm optimization in Gaussian process regression for three objects with four coordinate variables each (Scenario II).

Target Hyperparameter values

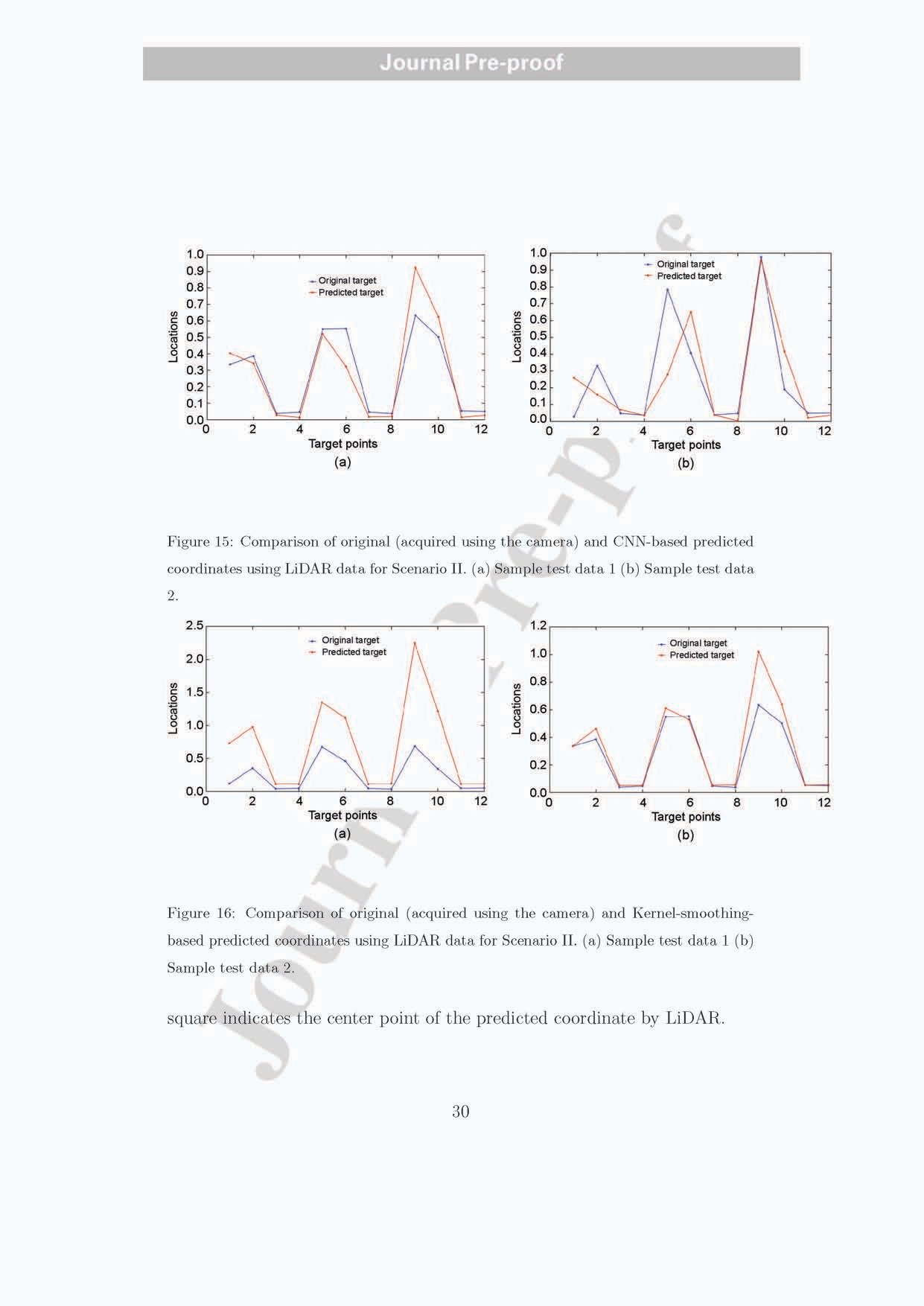
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Coordinate variables | *θ*0 | *θ*1 | *θ*2 | *θ*3 | *λ* |
| 1 | 12.359 | 56.254 | 51.840 | 9.275 | 17.189 |
| 2 | 9.002 | 50.690 | 46.310 | 5.046 | 60.473 |
| 3 | 0.130 | 55.443 | 54.637 | -0.380 | 55.126 |
| 4 | 0.548 | 24.425 | 62.118 | -0.573 | 108.492 |
| 5 | -8.172 | 94.264 | 77.497 | 50.605 | 57.844 |
| 6 | 9.181 | 79.162 | 35.202 | 14.843 | 40.920 |
| 7 | 18.989 | 55.319 | 88.712 | -7.182 | 67.899 |
| 8 | -4.958 | 33.939 | 58.522 | 36.850 | 61.056 |
| 9 | 34.409 | -23.239 | 86.437 | 5.585 | 65.577 |
| 10 | 10.041 | 93.836 | 31.346 | -1.744 | 48.303 |
| 11 | 0.853 | 51.439 | 67.845 | 15.337 | 48.158 |
| 12 | 4.256 | 42.023 | 47.473 | 6.650 | 15.271 |

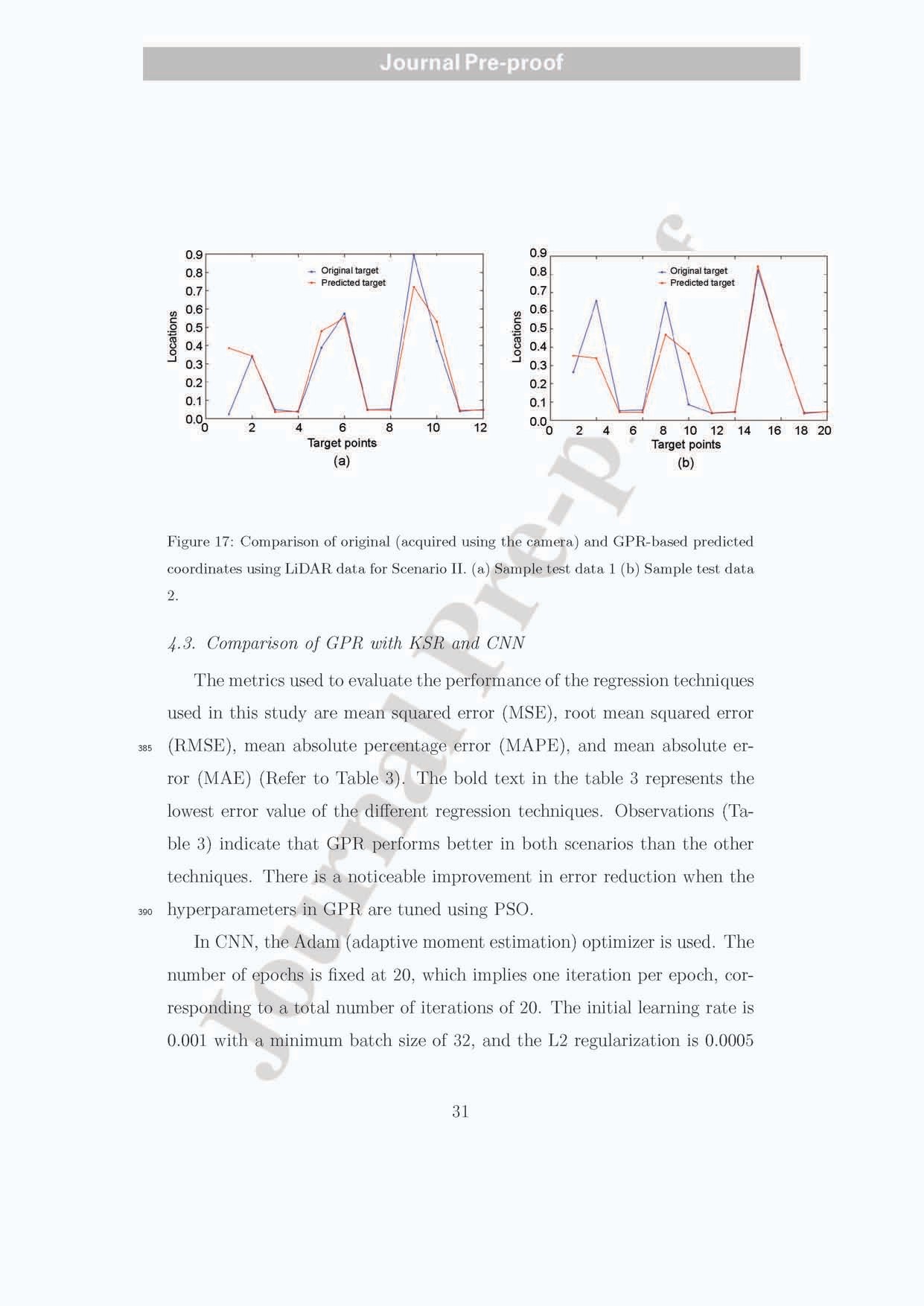
375

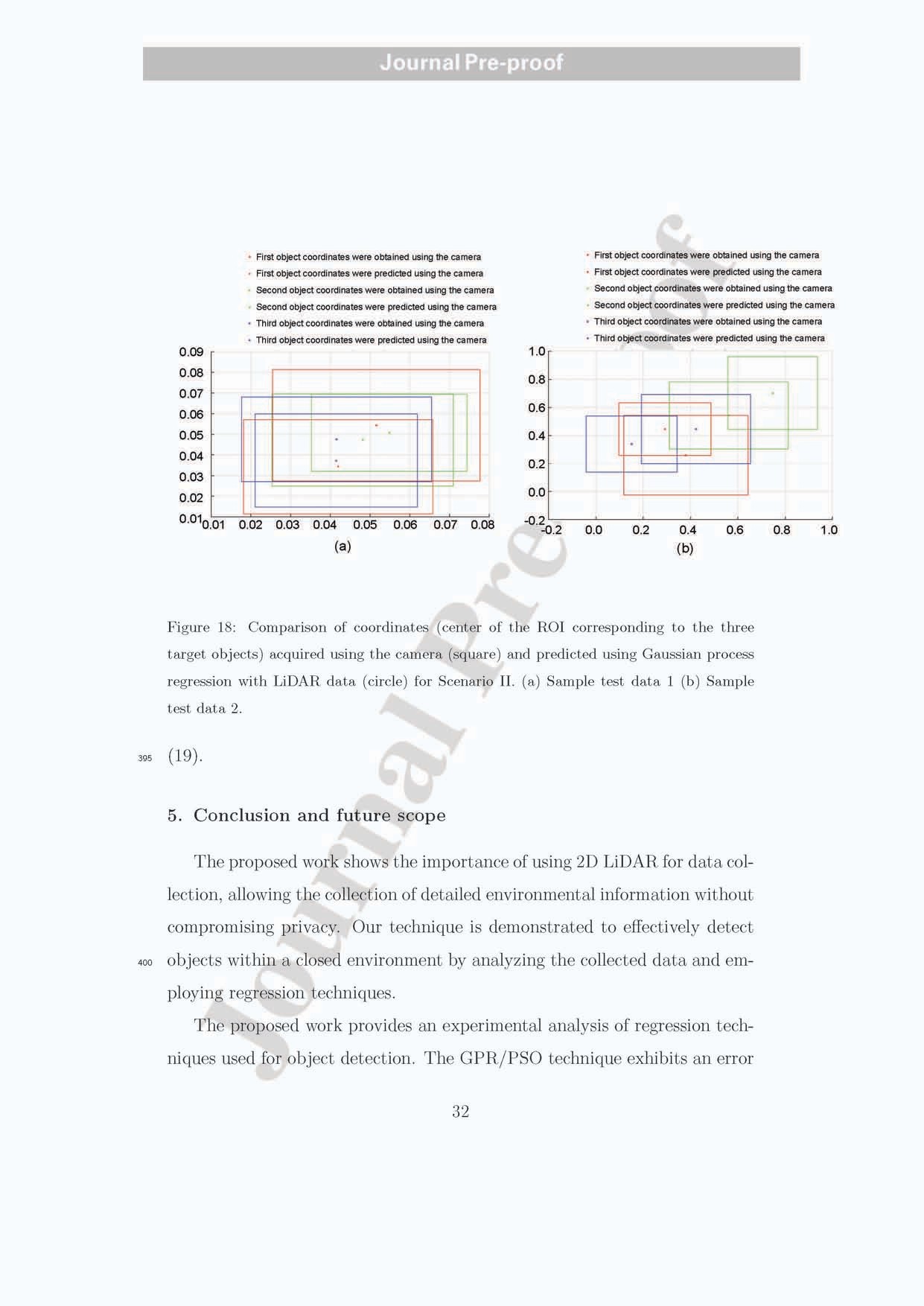
380

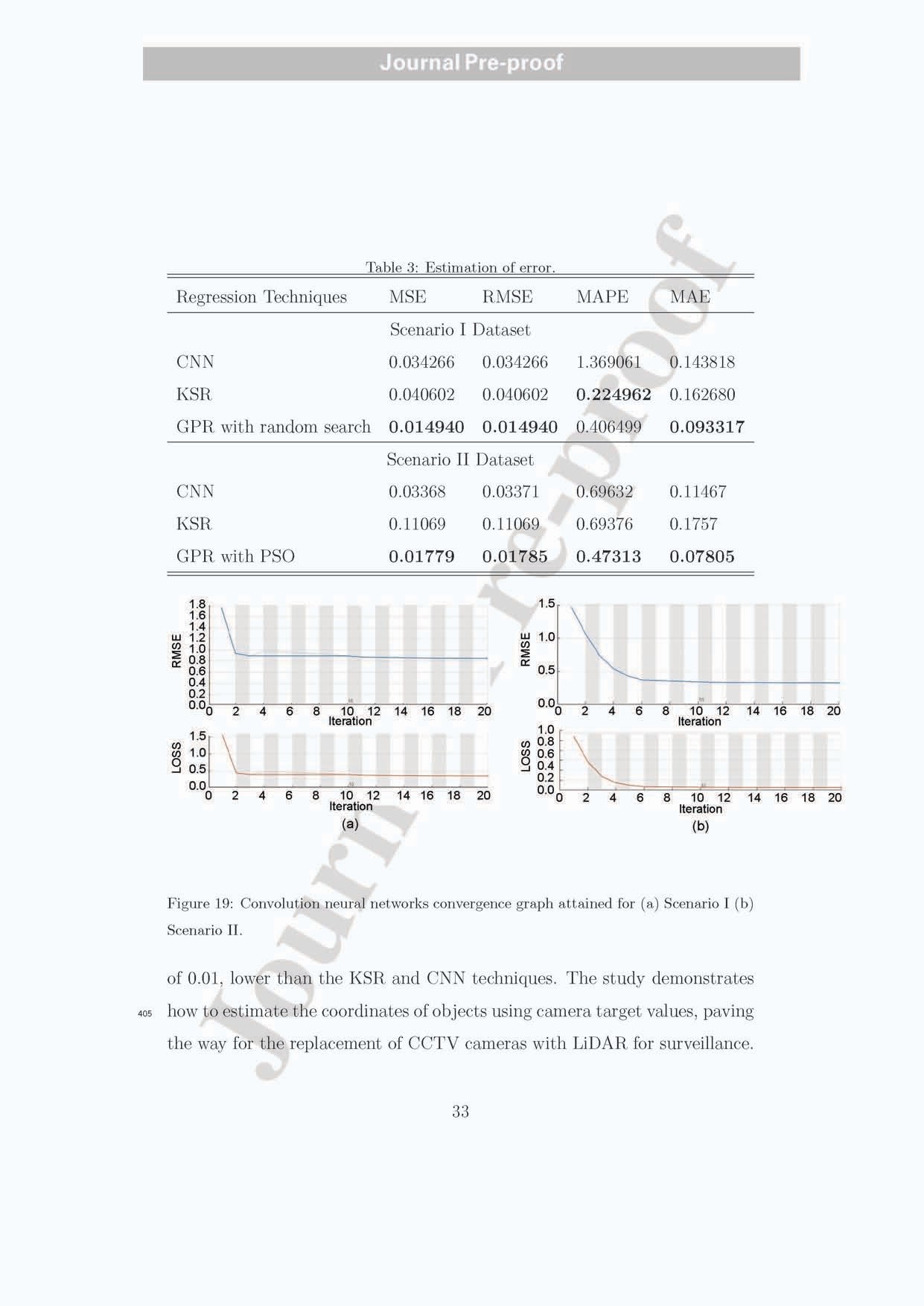
Fig. **??** and **??** shows sample plots illustrating bounding boxes for two randomly selected test data samples. These plots were generated using the GPR technique, with hyperparameters tuned using PSO in Scenario II. The location of the original object coordinates was obtained using the camera, and the predicted object coordinates were determined using LiDAR. The three colors (red, blue, and green) indicate the coordinates of the various objects. The circle shows the center point of the object coordinate, and the

29









410

415

420

In the future, the integration of LiDAR and regression techniques in IoT applications holds significant potential. Combining the capabilities of LiDAR with the power of regression algorithms can enhance various domains, such as autonomous systems, robotics, surveillance systems, and smart-city appli- cations. By leveraging LiDAR’s privacy-preserving nature, safer and more efficient IoT solutions can be realized in the future.

## 6. Declaration of competing interest

The authors declare that they have no known competing financial inter- ests or personal relationships that could have appeared to influence the work reported in this paper.

## References

Alejandra, Carolina, H., Clara, G., Jonathan, C., Ram´on, B., 2016. Ob- ject detection applied to indoor environments for mobile robot navigation. Physical Sensors .

Behroozpour, B., Phillip, A.M.S., Wu, M.C., Bernhard, E.B., 2017. Lidar system architectures and circuits, in: IEEE Communications Magazine,

pp. 135 – 142.

Benedek, C., Majdik, A., sNagy, B., Rozsa, Z., Sziranyi., T., 2021. Posi-

425 tioning and perception in lidar point clouds. Digital Signal Processing 119.

Bu, F., Le, T., Du, X., Vasudevan, R., Vasudevan, R., Johnson-Roberson., M., 2020. Pedestrian planar lidar pose (pplp) network for oriented pedes-

trian detection based on planar lidar and monocular images. IEEE

430 Robotics and Automation Letters 5, 1626 – 1633.

Christopher, M.B., 2006. Pattern recognition and machine learning, in: In- formation Science and Statistics, Springer New York, NY, pp. XX, 738.

Demetri, S., Gian, Pietro, P., Bruzzone., L., 2019. Laps: Lidar-assisted placement of wireless sensor networks in forests. ACM Transactions on

435 Sensor Networks 15, 1–40.

Drazen, B., Takayuki, K., Tetsushi, I., Takahiro, M.., 2013. Person tracking in large public spaces using 3-d range sensors. IEEE Transactions on Human- Machine Systems 43.

440

445

Gopi., E.S., 2020. Pattern recognition and computational intelligence tech- niques using matlab, in: Transactions on Computational Science and Com- putational Intelligence (TRACOSCI), springer.

Kennedy, J., Eberhart., R., 1995. Particle swarm optimization, in: Proceed- ings of ICNN’95 - International Conference on Neural Networks.

Li, J., Stevenson., R., 2021. 2d lidar and camera fusion using motion cues for indoor layout estimation, in: IEEE 24th International Conference on Information Fusion (FUSION).

Liu, W., Chen, S., Hauser, E., 2017. Lidar-based bridge structure defect detection. Experimental Techniques. 35, 27–34.

Lvwen, H., Siyuan, C., Jianfeng, Z., Bang, C., Mingqing, L., 2017. Real-

450

455

460

465

time motion tracking for indoor moving sphere objects with a lidar sensor. Sensors for Transportation .

MahmudulHasan, Hanawa, J., Goto, R., Suzuki, R., Fukuda, H., Kuno, Y., Kobayashi., Y., 2022. Lidar-based detection, tracking, and property esti- mation: A contemporary review. Neurocomputing 506, 393–405.

Miao, S., Wang, Z.J., Rui, L., 2016. A cnn regression approach for real-time 2d/3d registration. IEEE Transactions on Medical Imaging. 35, 1352 – 1363.

Miawarni, H., Sardjono, T.A., Setijadi, E., Wijayanti, Arraziqi, D., Gumelar, A.B., Purnomo., M.H., 2020. Fall detection system for elderly based on 2d lidar: A preliminary study of fall incident and activities of daily living (adl) detection, in: International Conference on Computer Engineering, Network, and Intelligent Multimedia (CENIM).

Pintea, S., Karao˘glu, S., van Gemert, J., Smeulders, A., 2016. Large scale gaussian process for overlap-based object proposal scoring. Computer Vi- sion and Image Understanding. 150, 95–108.

Sohn, K., Bernardi, A., Neiroukh., O.R., 2021. Effects of input resolution on lidar pedestrian localization using deep learning, in: International Confer- ence on Electronics, Information, and Communication (ICEIC).

Wang, R., An, M., Shao, S., Yu, M., Wang, S., Xu., X., 2021. Lidar sensor-

470 based object recognition using machine learning. Journal of Russian Laser Research 42, 484 – 493.

Wu, Y., Wang, Y., Zhang, S., Ogai., H., 2021. Deep 3d object detection networks using lidar data: A review. IEEE Sensors Journal 21, 1152 – 1171.

475 Yao, Y., Fatholahi, S., Chen, Y., ., J.L., 2023. Indoor lidar point clouds upsampling for object detection enhancement, in: Joint Urban Remote Sensing Event (JURSE).

**Declaration of interests**

☒ The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

* The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: