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

Generative Adversial Networks for Enhanced Visual Localization in Autonomous Systems

S. Sindhu[1], M. Saravanan[2*]

Submitted: 19/01/2024 **Revised**: 28/02/2024 **Accepted**: 05/03/2024

Abstract*:* Autonomous system localization is pivotal for determining their position within their environment. While the Global Positioning System (GPS) is a widely used method, its limitations, such as imprecise pose estimation, necessitate alternative approaches. Visual localization is one such approach that localize the system with images captured by cameras, offers a promising solution. In this research, we employ Generative Networks and Deep Learning techniques to calculate Autonomous system positions relative to the world. Landmarks are detected using generative networks, and the autonomous system is localized using binarized spiking neural networks based on the identified landmarks. The proposed model achieves a mean Intersection over Union (mIoU) score of 0.85, showcasing a 6.25% improvement over existing models. The presented framework enhances system localization accuracy, minimizing pose errors in both outdoor environments and GPS-denied locations.

Keywords: *Localization, Autonomous systems, GAN, Binarized Spiking Neural Networks, KITTI*

1. Introduction

THE CZECH WRITER KAREL CAPEK COINED THE TERM "ROBOT" IN 1920. An autonomous system is a machine capable of independent movement and performing complex tasks without human assistance [1]. These robots can be divided into different categories, namely pre-programmed robots, autonomous robots, remote-controlled robots and assistance robots. While some robots can operate independently, others require human interaction to carry out their tasks. Autonomous mobile robots (AMRs) find applications in various industries such as farming, healthcare, logistics, and smart cities [2]. Robotic systems rely on sensing, planning, and action to navigate their environment effectively. Autonomous mobile robots employ sophisticated algorithms to make efficient decisions and evaluate the situation. Through sensors, robots can interact with the external world. These sensors receive electric signals processed by the robot's controller unit, enabling interaction with the physical environment. Proprioceptive sensors [3] measure the robot's internal parameters, while exteroceptive sensors measure the external world. Commonly used exteroceptive sensors in autonomous robots include light sensors (lidar) for light detection, proximity sensors for obstacle detection, infrared sensors for object motion detection, and touch sensors for physical touch detection. Additionally, robots incorporate sensors related explicitly to mobility, such as an Inertial Measurement Unit (IMU) for orientation and velocity measurement, a Global Positioning System (GPS) for latitude and longitude information, and vision sensors for image capture and orientation identification.

[1]*Department of Data Science and Business Systems,*
College of Engineering and Technology,
SRM Institute of Science and Technology,
Kattankulathur, India.
Email: sindhus2@srmist.edu.in
[2]*Department of Networking and Communications,*
College of Engineering and Technology,
SRM Institute of Science and Technology,
Kattankulathur, India.
Email: saravanm7@srmist.edu.in
**(Corresponding Author)*

The GPS measurement error, which can exceed ten meters due to signal reflections, clock inaccuracies, and atmospheric factors, poses an impracticality for vehicle navigation, particularly in scenarios such as tunnels and densely built urban areas.

Localization of autonomous systems helps to determine the system's current location in its environment [4]. Determining the robot's location in a static environment is relatively easy since it is the only object in motion [5]. However, in a dynamic environment, localization becomes considerably more challenging as the presence of other moving objects confuses the robot regarding its position.

To determine its location, the robot relies on its exteroceptive sensors, including the laser sensor, vision system, IMU, and Lidars, to collect data pertaining to surrounding environment. By fusing this sensor data with the robot's odometry, it is able to determine its own position. Despite the presence of a GPS, the robot cannot directly measure its precise location. [6] Rather, it must rely on the data from its sensors to make the most accurate estimation of its position.

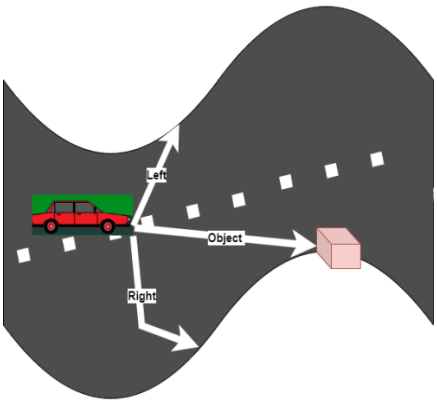

Fig. 1. Road Scenario for Localization

The above Figure 1 shows a road scenario that demonstrates how an autonomous vehicle can be located. The autonomous system

uses localization techniques to determine its exact location on the road's right and left side borders and any on-road objects. Sensors such as LIDAR, Radar, and cameras are the primary means of external observation for autonomous vehicles [7].

The two primary approaches in Localization are Satellite Localization and visual Localization [8]. Precise Localization is possible with the Global Navigation Satellite System (GNSS). The drawback of satellite-based Localization includes the unavailability of signals in tunnels and narrow streets. The GNSS-based Localization [9] receives the object's location but not the full pose. Visual Localization uses the recorded map of the area with feature extraction.

The primary support for autonomous systems relies on the following things: perception, Localization, planning, and control.

Perception of an autonomous system relies on its ability to collect data and gain relevant insights about the environment. The Autonomous system makes use of Localization to assist in locating itself. Planning is the process of making deliberate decisions. Typically, this involves moving the vehicle from its starting location to its destination while avoiding obstacles and optimizing path-planning algorithms. Control is the higher-level process that includes the execution of steps involved in planning [10].

2. Literature Survey

Localization techniques were developed to address the "Where am I?" These methods empower robots, particularly in GPS-denied environments, to navigate around obstacles safely, avoiding collisions [11].

Visual Odometry (VO) is the process of localizing autonomous vehicles with the monocular or stereo images collected from the camera installed on the autonomous system. Visual odometry localizes the Autonomous system by detecting the key points and matching between frames, whereas the direct method estimates the position by minimizing the photometric error in pixels [12].

Feature-based methods [13] with semi-dense alignment are computationally less expensive and give accurate pose estimation. Another feature-based technique proposed by [14] is compatible with stereo and RGB-D sensors. Semantic segmentation–aided Visual odometry is also popular in VO Pipeline [15].

Recently, Deep learning-based approaches have been popular in ego-motion estimation [16] by employing CNNs with RNNs. The proposed method outperforms state-of-the-art methods with reduced translational and rotational errors. Gated recurrent unit-based implementation minimizes the error in the cornering section by taking classical visual odometry as input [17].

In 2021, Li, G. et al. [18] introduces a real-time visual Simultaneous Localization and Mapping (SLAM) system leveraging deep learning, incorporating a multi-task feature extraction network and self-supervised feature points. SLAM serves as the foundational framework for enabling intelligent mobile robots to navigate unfamiliar surroundings. The proposed system utilizes a simplified Convolutional Neural Network (CNN) to detect feature points and descriptors, replacing the conventional feature extractor, thereby improving the accuracy and stability of the visual SLAM system.

In addition to camera images, Visual Inertial odometry (VIO) [19] uses IMU measurement to estimate the position of the Autonomous System. The authors in [20] introduced SelfVIO architecture that employs CNNs and LSTM to handle the data. This visual SLAM method is tested on benchmark datasets, including KITTI, EuRoC and Cityscapes. The Visual Inertial odometry is implemented using the classical Extended Kalman filter [21].

In reference [22], researchers introduced a deep sensor fusion approach that combines data from a 2D laser scanner and an IMU (Inertial Measurement Unit) to facilitate mobile robot localization. Their method involves the development of an architecture based on recurrent convolutional neural networks (RCNN) to integrate information from laser scans and inertial measurements, enabling accurate pose estimation between consecutive scans for robot localization. Chikara et al. [23] introduced a novel approach known as Deep Convolutional Neural Network optimized with Genetic Algorithms (DCNN-GA) for the localization of autonomous vehicles. This method was designed to enable the autonomous navigation of Unmanned Aerial Vehicles (UAVs) within indoor building corridors, leveraging deep neural networks to process images. Determining the optimal combination of hyperparameters for improved prediction accuracy is a complex challenge when working with deep neural networks. In this study, the authors handled this challenge by employing genetic algorithms to fine-tune the hyperparameters of the convolutional neural network. Authors in [24] have presented a Domain adaptation for semantic and geometric-aware image-based localization (Dasgil) for mobile robots. Long-term visual localization in a changing environment is a complicated problem in autonomous vehicles and mobile robotics due to seasons, light variations, etc. This paper proposes a multitasking architecture to combine geometric and semantic information into a multi-scale latent representation for visual location recognition. In order to benefit from high-quality ground truth without any human effort, an efficient multi-scale feature discriminator for adversarial training is proposed to achieve domain adaptation from KITTI virtual synthetic dataset to real world dataset.

Chen X et al. [25] leveraged Lidar scan data obtained from Autonomous vehicles to tackle the challenge of addressing the loop closing problem in Localization. They used a modified Siamese network to estimate the correspondence between Lidar data points and utilized the overlap method for detecting loop closures. Additionally, they integrated Monte Carlo localization into their existing approach, yielding enhanced results in accurately localizing the system within urban environments. Wen, S. et al. [26] Combined the path planning algorithm with Simultaneous Localization and mapping. FastSLAM is used for localizing the robot and Dueling DQN algorithm is used for path planning for a robot.

3. Landmark Detection Utilizing Generative Network

Localization is the process of determining a vehicle's position and orientation within its environment. Landmark detection plays a significant role in this process by identifying and tracking key reference points or features in the vehicle's surroundings.

A. Generator Phase

The generator receives input as a bounding box and, optionally, the image area enclosed by that box. Its role is to generate a set of landmark points within the specified bounding box. These points correspond to the detected landmarks. The generator is

conditioned on the input image and bounding box information to ensure the generated landmarks are consistent with the context.

As shown in Fig. 2, the generator stage contains two deconvolution layers, an encoding network and a decoding network. In this process, cameras capture images from the outdoor environment to detect the landmark, initially mapped before being provided to the encoding phase. Here, the mapping is done with the help of the deconvolution layer of the DDcGenNet, and the output of this mapping is known as the bounding boxes. Next, the encoding phase receives the mapped features. The encoder transforms the input data (e.g., an image containing landmarks) into a compact, informative representation known as a feature vector or latent space representation.

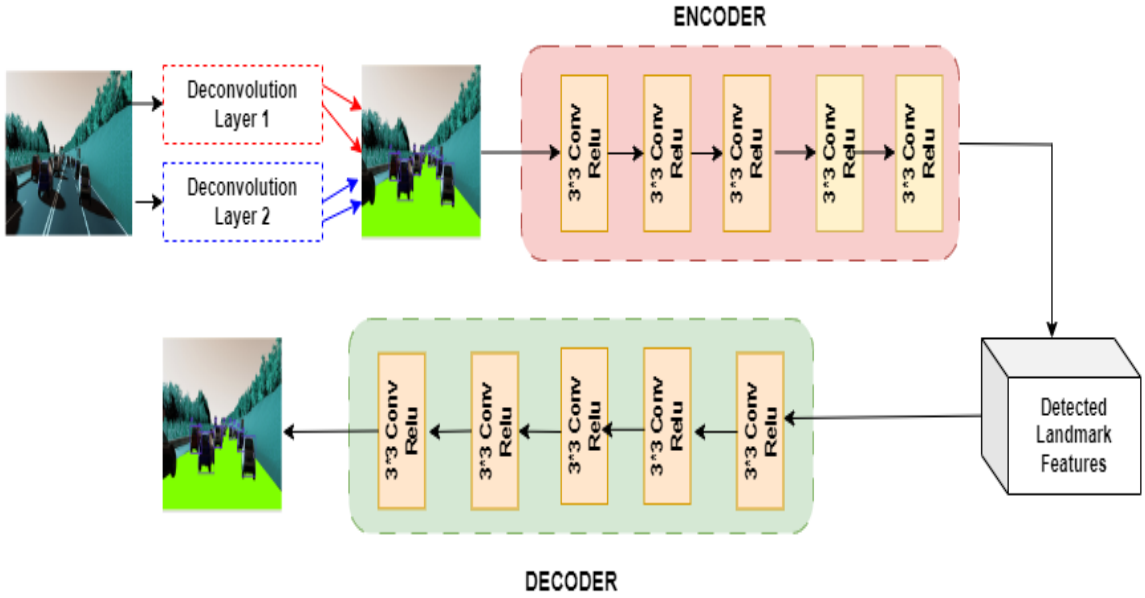

Fig. 2. Architecture of Generator

Further, the decoder phase receives these images and comprises five convolution layers. The primary role of the decoder is to transform the encoded feature representation back into a format that closely resembles the original input. The network gradually reconstructs the input data while maintaining relevant features by applying a series of convolutional layers and activations in the decoder**.**

B. Discriminator Phase

The Discriminator phase contains two discriminators. Discriminator 1 (D1) assesses the quality of the generator's output regarding landmark detection. Discriminator 2 (D2) distinguishes between real images with actual landmark annotations and fake images generated by the generator. Fig. 3 illustrates the structure of the discriminator.

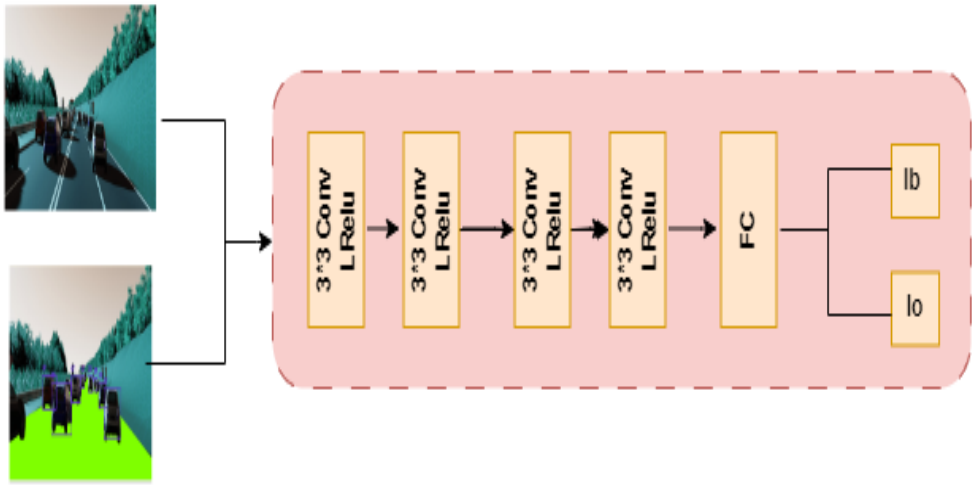

Fig. 3. Architecture of Discriminator

C. Training Process

The steps in training of Generator and Discriminator are explained in Algorithm 1. The training process begins with a loop that iterates for a specified number of epochs. We proposed a Dual discriminator (Dr and Df). Real images (r1, r2, ... rn) and fake images (f1, f2, ... fn) are sampled. Real images come from the dataset, while fake images are generated by the generator using random noise (z) and conditional information (c). The parameters of the discriminator Df and Dr are updated using the Adam optimizer to minimize the loss.

Training continues as long as the adversarial loss is above a certain threshold (Lmax) and a maximum number of training steps

(Tmax) has not been reached. This aims to ensure that the discriminators become better at their task. After training the discriminators, the code proceeds to train the generator.

This pseudocode represents a training strategy for a Generative Network, where the generator aims to produce images that can successfully deceive the discriminator(s).

Input: *Random noise (z) and conditional information*

Output: *Generated landmark data (G (z, c))*

Parameter Description: *Number of steps to train G, Dr, Df, TG,T_{Dr},T_{Df}*

Tmax ←Maximum steps to train networks

Lmax & Lmin Adversial losses for Generator and Discriminator, $L_G$max-Total loss of Generator

```
for epoch in range(num_epochs):
    #Trainina the Discriminators Dr, Df
    Sample for real Images (r1 ,r2 ,..rn) and fake images (f1,f2,...fn)
    Obtain the Generated data G(r1,r2 ,..rn),...G(f1,f2,...fn)
    Update discriminator Parameter by AdamOptimizer to reduce Df Loss(I)
    Update discriminator Parameter by AdamOptimizer to reduce Dr Loss (II)
    While LDr > Lmax and TDr<Tmax, repeat (I)
        TDr = TDr+1
    End While
    While LDf >Lmax and TDf<Tmax, repeat (II)
        TDf = TDf+1
    End While
    #Training the Generator
    Sample for real Images (r1,r2 ...rn),(z,c)
    Update generator Parameter by AdamOptimizer to reduce Generator Loss (III)
    While (LDf < Lmin or LDr < Lmin) and Tg <Tmax
    Update generator Parameter by AdamOptimizer to minimize adversial loss
        TG = TG+1;
    End While
    While LG > LGmax and TG <Imax repeat III
    End While
End For
```

Algorithm 1 Training Process for Landmark Detection

4. Localization of Autonomous Systems

A. System Overview

Detecting robot localization in outdoor environment is a challenging task due to its hard environment. Here the localization problem of the robot is represented as the robot pose $RP(F) = [a \quad b \quad \phi]^T$, where a and b are in meters and ϕ is the orientation in degree. Once the landmarks are detected in the environment using generative network, the next step is to estimate the current pose (position and orientation) of the autonomous system. Binarized Spiking Neural Networks can be employed for this task.

Spiking neural networks (SNNs) are a type of artificial neural network inspired by the behaviour of biological neurons, and binarized SNNs are a variation that uses binary activations. The binarized SNN is designed to process the encoded landmark data. In SNNs, neurons fire in discrete, spiking patterns over time. Binarized SNNs use binary activation values (usually 0 or 1) instead of continuous activations. The network's architecture, including the number of neurons, layers, and connections, should be carefully designed for the localization task.

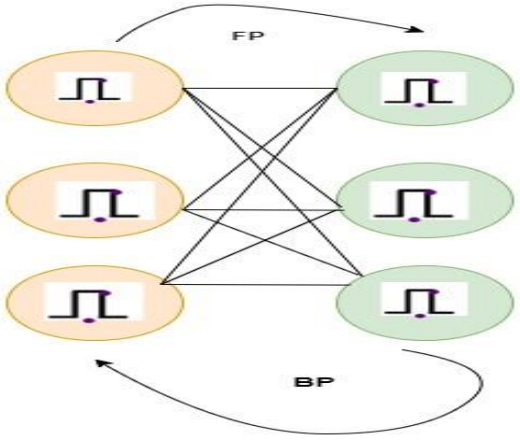

Fig. 4. Structure of SNN

BSNNs convert the Input Images into Spike Train.Single spike Temporal coding Method is used for conversion.

The combination of Gans for landmark detection and BSNNs for localization provides a powerful solution for autonomous vehicles. Generative Network help in identifying landmarks and features in the environment, while BSNNs efficiently process this information to estimate the vehicle's position and orientation, enabling it to navigate autonomously.

5. Results and Discussions

We implemented the novel landmark detection method with stereo images using the TensorFlow deep learning framework in Python, and we verified it through the KITTI dataset.

A. KITTI Dataset

Researchers compiled the dataset by conducting driving experiments in various urban traffic scenarios located in Karlsruhe, Germany. The KITTI dataset consists of 22 image sequences, with the initial 11 sequences (sequences 00–10) offering ground truth data derived from high-precision GPS and

laser sensors. This dataset presents several noteworthy challenges, including the presence of dynamic moving objects such as vehicles, cyclists, and pedestrians.

B. Evaluation Metrics

a) Mean Intersection over union (MIoU)

Intersection over Union (IoU) evaluates the overlap between truth and predicted regions.

IoU = (Intersection Area) / (Union Area)

Mean Intersection over union is calculated by the below formula where N is the Number of classes considered. Fig.5 shows the comparison of mean IoU value for Landmark detection.

$$mIoU = \left(\frac{1}{N}\right) * \sum IoU \qquad \text{(Eq 1)}$$

Eq 1 depicts the mean intersection over union process for landmark detection.

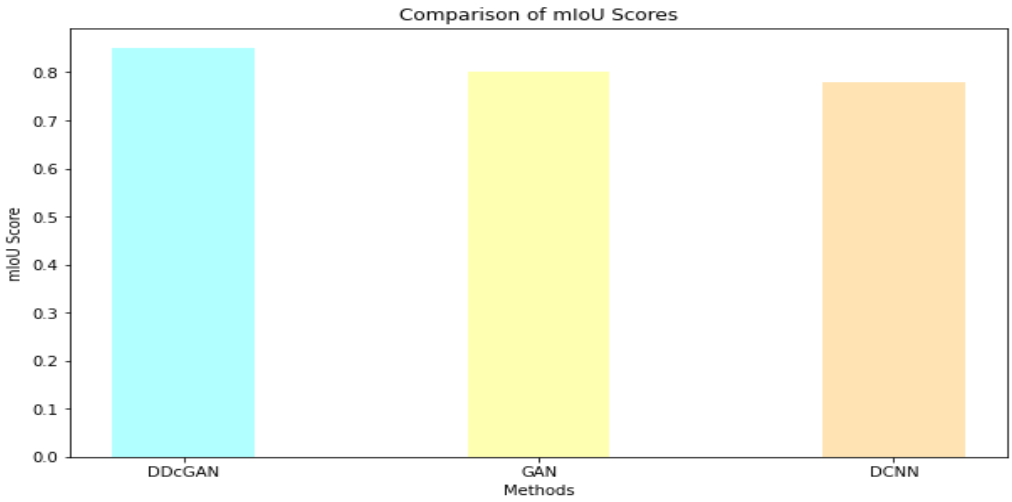

Fig. 5. MIoU Score Comparison

C) Precision and Recall

The following equations can be utilized to calculate precision and recall:

Precision=TP/(TP+FP) Recall=TP/(TP+FN)

TP--Model correctly predicts the positive class i.e., amount of accurately matched bounding boxes when comparing detection boxes to reference boxes within the dataset.

FP--Model incorrectly predicts the positive class. i.e., bounding boxes that were either missed or incorrectly positioned within the detection results.

FN--Model incorrectly predicts the negative class. i.e., bounding boxes that were present in the reference dataset but did not appear in the detection results.

Fig. 6. Shows the improved performance of our proposed method having better precision, recall and F1 score values.

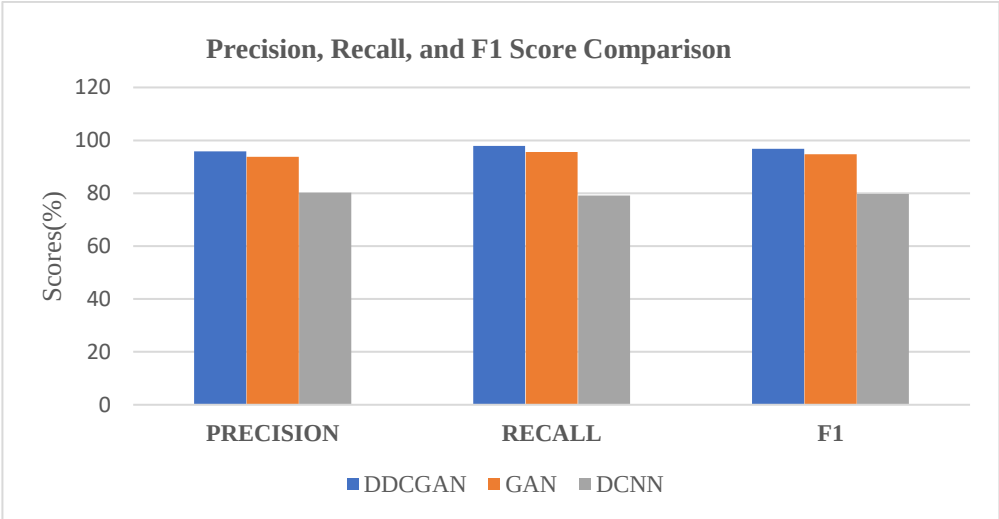

Fig. 6. Precision, Recall and F1 score comparison with Existing approaches

D) Translational Error

We conducted a performance evaluation of our novel localization method using the widely recognized KITTI odometry dataset. We considered sub-sequences of lengths ranging from 100 to 800 meters for estimation. Fig 7 compares the translational errors for path lengths up to 800 meters, benchmarking our method against state-of-the-art techniques. The translational error is calculated using the below formula in Eq 2.

$$error_{trans=} = \sqrt{(x_{est} - x_{gt})^2 + (y_{est} - y_{gt})^2 + (z_{est} - z_{gt})^2} \quad (Eq\ 2)$$

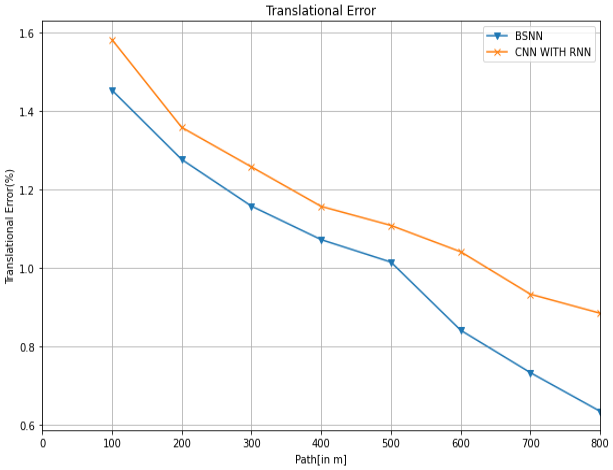

Fig 7. Translational error vs Sequence length

6. Conclusion and Future Work

The proposed work has significantly contributed to landmark detection and localization of Autonomous systems. The utilization of the Generative model has substantially improved the precision of landmark detection, as evidenced by achieving an impressive mIoU score of 0.85. This represents a substantial 6.25% enhancement compared to existing models, underscoring the effectiveness of our proposed approach.

Furthermore, optimizing Binarized Spiking Neural Networks has significantly reduced false localization within autonomous systems, enhancing their reliability and accuracy. Additionally, integrating the Fusion-Enabled OELM Framework has facilitated the blending of images and IMU data, leading to a notable reduction in translational and rotational errors in localization.

In the future, real-world scenarios will be considered to enhance the system's ability to localize itself accurately, both in outdoor environments and in challenging GPS-denied locations. The data collected from other exteroceptive sensors will also be considered for evaluation.

7. References

[1] Siegwart, R., Nourbakhsh, I. R., & Scaramuzza, D. (2011). Introduction to autonomous mobile robots. MIT press.

[2] Dobriborsci, D., Kapitonov, A., & Nikolaev, N. (2017, July). The basics of the identification, localization and navigation for mobile robots. In 2017 International Conference on Information and Digital Technologies (IDT) (pp. 100-105). IEEE.

[3] Zhang, T., Li, Q., Zhang, C. S., Liang, H. W., Li, P., Wang, T. M., & Wu, C. (2017). Current trends in the development of intelligent unmanned autonomous systems. Frontiers of information technology & electronic engineering, 18, 68-85.

[4] Tao, Z., Bonnifait, P., Fremont, V., & Ibanez-Guzman, J. (2013, November). Mapping and localization using GPS, lane markings and proprioceptive sensors. In 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (pp. 406-412). IEEE.

[5] Conduraru, I., Doroftei, I., & Conduraru, A. (2014). Localization methods for mobile robots-a review. Advanced Materials Research, 837, 561-566.

[6] Panigrahi, P. K., & Bisoy, S. K. (2022). Localization strategies for autonomous mobile robots: A review. Journal of King Saud University-Computer and Information Sciences, 34(8), 6019-6039.6. Everett, H. R. (1995). Sensors for mobile robots. CRC Press.

[7] Cenkeramaddi, L. R., Bhatia, J., Jha, A., Vishkarma, S. K., & Soumya, J. (2020, November). A survey on sensors for autonomous systems. In 2020 15th IEEE Conference on Industrial Electronics and Applications (ICIEA) (pp. 1182-1187). IEEE.

[8] Couturier, A., & Akhloufi, M. A. (2021). A review on absolute visual localization for UAV. Robotics and Autonomous Systems, 135, 103666.

[9] Liu, X., Ballal, T., & Al-Naffouri, T. Y. (2019, September). GNSS- based localization for autonomous vehicles: Prospects and challenges. In Proc. 27th Eur. Signal Process. Conf. (EUSIPCO) (pp. 2-6).

[10] Pendleton, S. D., Andersen, H., Du, X., Shen, X., Meghjani, M., Eng, Y. H., . & Ang Jr, M. H. (2017). Perception, planning, control, and coordination for autonomous vehicles. Machines, 5(1), 6.

[11] Huang, S., & Dissanayake, G. (1999). Robot localization: An introduction. Wiley Encyclopedia of Electrical and Electronics Engineering, 1-10.

[12] Scaramuzza, D., & Fraundorfer, F. Visual Odometry [Tutorial]. IEEE Robot. Autom. Mag. 2011, 18, 80–92. oi:10.1109/MRA.2011.943233.

[13] Krombach, N., Droeschel, D., Houben, S., & Behnke, S. (2018). Feature-based visual odometry prior for real-time semi-dense stereo SLAM. Robotics and Autonomous Systems, 109, 38-58.

[14] Aladem, M., & Rawashdeh, S. A. (2018). Lightweight visual odometry for autonomous mobile robots. Sensors, 18(9), 2837.

[15] An, L., Zhang, X., Gao, H., & Liu, Y. (2017). Semantic segmentation–aided visual odometry for urban autonomous driving. International Journal of Advanced Robotic Systems, 14(5), 1729881417735667.

[16] Pandey, T., Pena, D., Byrne, J., & Moloney, D. (2021). Leveraging deep learning for visual odometry using optical flow. Sensors, 21(4), 1313.

[17] Kim, S., Kim, I., Vecchietti, L. F., & Har, D. (2020). Pose estimation utilizing agated recurrent unit network for visual localization. Applied Sciences, 10(24), 8876.

[18] Li, G., Yu, 9L. and Fei, S., 2021. A deep-learning real-time visual SLAM system based on multi-task feature extraction network and self-supervised featurepoints. Measurement, 168, p.108403.

[19] Scaramuzza, D., & Zhang, Z. (2019). Visual-inertial

odometry of aerial robots. arXiv preprint arXiv:1906.03289.
[20] Almalioglu, Y., Turan, M., Saputra, M. R. U., de Gusmão, P. P., Markham, A., & Trigoni, N. (2022). SelfVIO: Self-supervised deep monocular Visual–Inertial Odometry and depth estimation. Neural Networks, 150, 119-136.
[21] Bloesch, M., Omari, S., Hutter, M., & Siegwart, R. (2015, September). Robust visual inertial odometry using a direct EKF- based approach. In 2015 IEEE/RSJ international conference on intelligent robots and systems (IROS) (pp. 298-304). IEEE.
[22] Li, C., Wang, S., Zhuang, Y. and Yan, F., (2019). Deep sensor fusion between 2D laser scanner and IMU for mobile robot localization. IEEE Sensors Journal, 21(6), pp.8501-8509.
[23] Chhikara, P., Tekchandani, R., Kumar, N., Chamola, V. and Guizani, M., (2020). DCNN-GA: A deep neural net architecture for navigation of UAV in indoor environment. IEEE Internet of Things Journal, 8(6), pp.4448-4460.
[24] Hu, H., Qiao, Z., Cheng, M., Liu, Z. and Wang, H., (2020). Dasgil: Domain adaptation for semantic and geometric-aware image- based localization. IEEE Transactions on Image Processing, 30, pp.1342-1353.
[25] Chen, X., Läbe, T., Milioto, A., Röhling, T., Behley, J. and Stachniss, C., (2022). OverlapNet: a siamese network for computing LiDAR scan similarity with applications to loop closing and localization. Autonomous Robots, 46(1), pp.61-81.
[26] Wen, S., Zhao, Y., Yuan, X., Wang, Z., Zhang, D. and Manfredi,L., 2020. Path planning for active SLAM based on deep reinforcement learning under unknown environments. IntelligentService Robotics, 13(2), pp.263-272.
[27] Geiger, A., Lenz, P., Stiller, C., & Urtasun, R. (2013). Vision meets robotics: The kitti dataset. The International Journal of Robotics Research, 32(11), 1231-1237.