IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 15 February 2024, accepted 3 March 2024, date of publication 12 March 2024, date of current version 18 March 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3374197

APPLIED RESEARCH

From Augmentation to Inpainting: Improving Visual SLAM With Signal Enhancement Techniques and GAN-Based Image Inpainting

CHARALAMBOS THEODOROU[1,2], VLADAN VELISAVLJEVIC[1], (Senior Member, IEEE), VLADIMIR DYO[3], AND FREDI NONYELU[2]

[1]School of Computer Science and Technology, University of Bedfordshire, LU1 3JU Luton, U.K.
[2]Briteyellow Ltd., MK43 0BT Bedford, U.K.
[3]Department of Electronic Engineering, Royal Holloway, University of London, TW20 0EX Egham, U.K.

Corresponding author: Charalambos Theodorou (Theodorou.Charalampos@study.beds.ac.uk)

This work was supported by Briteyellow Ltd.

ABSTRACT This paper undertakes a comprehensive investigation that surpasses the conventional examination of signal enhancement techniques and their effects on visual Simultaneous Localization and Mapping (vSLAM) performance across diverse scenarios. Going beyond the conventional scope, the study extends its focus towards the seamless integration of signal enhancement techniques, aiming to achieve a substantial enhancement in the overall vSLAM performance. The research not only delves into the assessment of existing methods but also actively contributes to the field by proposing innovative denoising techniques that can play a pivotal role in refining the accuracy and reliability of vSLAM systems. This multifaceted approach encompasses a thorough exploration of the intricate relationships between signal enhancement, denoising strategies, their cumulative impact on the performance of vSLAM in real-world applications and the innovative use of Generative Adversarial Networks (GANs) for image inpainting. The GANs effectively fill in missing spaces following object detection and removal, presenting a novel state-of-the-art approach that significantly enhances overall accuracy and execution speed of vSLAM. This paper aims to contribute to the advancement of vSLAM algorithms in real-world scenarios, demonstrating improved accuracy, robustness, and computational efficiency through the amalgamation of signal enhancement and advanced denoising techniques.

INDEX TERMS Signal enhancement, denoising techniques, visual SLAM, object detection, simultaneous localization and mapping (SLAM), generative adversarial network (GAN).

I. INTRODUCTION

By enabling real-time 3D reconstruction and localization from visual sensor inputs, Visual Simultaneous Localisation and Mapping algorithms have revolutionized computer vision [1]. These algorithms have various applications and use cases, including robotics, augmented reality, and autonomous navigation. Noise in visual data, however, can significantly affect the performance of vSLAM algorithms, degrading their accuracy and robustness.

A variety of factors can cause noise in visual data, including sensor imperfections, environmental conditions, and lighting changes [2]. It is challenging for vSLAM algorithms to estimate and map pose accurately when these sources of noise introduce uncertainty and distortions. For vSLAM systems in real-world scenarios to be reliable and effective, it is crucial to address noise-related challenges [3].

Various signal enhancement techniques have been developed in the past to address these challenges [4]. To improve

The associate editor coordinating the review of this manuscript and approving it for publication was Shun-Feng Su.

the algorithm's generalization ability, synthetic variations are introduced into the training data. By adding Gaussian noise and light noise to the training data, vSLAM can learn to handle diverse and noisy input images effectively [5].

The introduction of noise during signal enhancement alone, however, is not enough. Maintaining accurate pose estimation and mapping requires appropriate handling of the noise. vSLAM algorithms are made more efficient by denoising techniques that mitigate the adverse effects of noise [6]. Using these techniques, the algorithm can make more accurate and reliable estimations by removing or reducing noise while maintaining important image features.

Using Generative Adversarial Networks (GANs) for image inpainting tasks has emerged as a technologically advanced solution in recent years [7], especially in complex and dynamic environments like train stations. The GAN is comprised of a generator and discriminator network that operates within a game-theoretic framework. Generator networks generate realistic image data in order to fill in missing or removed regions in a frame, while discriminator networks act as pseudo-critics, separating actual from generated images. In order to ensure visual plausible and contextually appropriate imagery is restored in the image voids following object removal, the generator minimizes the feedback from the discriminator [8].

Furthermore, GANs have demonstrated remarkable efficacy in super-resolution, translation, and synthesis of pictures [9]; yet their potential for integration with vSLAM systems to attain increased accuracy has not been thoroughly investigated. Traditionally, vSLAM systems rely on geometric and photometric consistency approaches to estimate occlusion, which is needed to fill in occlusion gaps for real-time 3D reconstruction and robot navigation [10]. This constraint prevents them from providing their consumers with realistic and convincing outcomes because they are unable to handle semantic information in the scene.

In this paper, we propose a novel comprehensive framework for vSLAM. By seamlessly integrating advanced signal enhancement techniques, denoising methods, and GANs for image inpainting, this framework surpasses traditional methodologies. Instead of merely identifying challenges within vSLAM systems, our approach introduces a unified and synergistic approach to addressing these challenges. The framework is designed to significantly enhance the accuracy, robustness, and computational efficiency of vSLAM algorithms in different real-world scenarios. The state-of-the-art innovation is leveraging the power of GANs for image inpainting after object detection and removal, our proposed framework not only ensures more accurate spatial awareness but also optimises the execution speed of vSLAM during mapping. This paper outlines the theoretical foundation, technical details, and results of our proposed framework, providing a roadmap for the future development and deployment of advanced vSLAM systems.

II. RELATED WORK

Conventional vSLAM methods estimate the camera posture and rebuild the surroundings using precise feature extraction and matching algorithms. However, because of their low-texture, these approaches frequently perform poorly in highly dynamic or low-textured situations [12]. One of the main tactics employed in recent studies to overcome these restrictions is the practise of adding noise to the data. Random noise was applied to the image's characteristics to boost resilience in difficult situations. Additionally, a technique to increase the accuracy and dependability of 3D reconstruction has been proposed by adding Gaussian noise [13] to depth measurements. The results of this work stimulate additional investigation into the application of noise-based augmentation techniques in order to increase the robustness and accuracy of vSLAM systems. This study demonstrates that signal enhancement can be effective in increasing the performance of vSLAM systems.

vSLAM methods based on noise-based augmentation could exhibit varying degrees of performance based on the application or dataset they were intended for. Different noise models or augmentation techniques can be required depending on the environment or sensor configuration. The accuracy and dependability of vSLAM algorithms may be hampered by the introduction of extra uncertainties and artefacts caused by noise in the data [14]. To get the best outcome, it is imperative to carefully strike a balance between eliminating noise and maintaining important information. Because noise based augmentation approaches can increase the amount of processing power needed for vSLAM systems, they may be necessary for real-time applications [15].

One of the most promising methods in the field of robotics and computer vision is the combination of vSLAM and denoising techniques [16]. The noise and outliers in the visual data make it challenging for conventional vSLAM methods to estimate pose and reconstruct the environment [17]. To address these issues, vSLAM pipelines have been designed to incorporate a range of denoising techniques [18]. It has been demonstrated that by lowering the noise related to the visual data, denoising techniques based on non-local means filtering can increase the accuracy of camera pose estimation. To improve vSLAM's resistance to noise and outliers in the vSLAM data, a wavelet-based denoising technique was also taken into consideration [19]. In their experiments, the quality of the environment mapping and the pose estimation accuracy were both higher. Further investigation is encouraged as the results obtained with denoising techniques show that vSLAM systems are more accurate and reliable.

Several denoising techniques can be applied to vSLAM, with varying outcomes possible based on the type of noise introduced to the visual data. Denoising algorithms should be carefully chosen or modified depending on the type of noise, as some noise types may respond better to certain

denoising techniques than to others. When compared to vSLAM algorithms that do not use denoising techniques, denoising techniques may introduce artefacts that affect the algorithms and potentially result in lower accuracy [20]. It's critical to find a balance between noise reduction and maintaining important visual elements. Denoising techniques have the potential to cause significant computation overheads during real-time vSLAM operations, which could restrict the system's applicability and efficiency in settings with limited resources.

In this work, the problem of filling in empty spaces after object detection and removal is tackled by combining GANs with vSLAM. At the moment, a number of techniques are available to deal with occlusions and incomplete scene information: Patch-based Inpainting [21], Planar Inpainting [22], Structure from Motion (SfM) Refinement [23], Texture Blending [24], Depth Completion [25], and Structure from Motion (SfM) Refinement. In earlier research, the question of how to fuse GANs with vSLAM to bridge the gaps left by vSLAM and other methods has not been thoroughly investigated. Since vSLAM systems have become available, GANs present a special chance to synthesise realistic content in obscured areas by training them to understand the underlying structure and appearance of the scene and utilising their generative powers to produce realistic content. This indicates that a system can significantly improve the precision and comprehensiveness of its reconstruction of the environment by utilising vSLAM. This innovative method for handling occlusions and imperfect scene comprehension, when coupled with vSLAM, creates new avenues for resolving the issues these situations present.

GAN-based synthesis needs a large and diverse training dataset in order to function well in vSLAM. It is important to remember that biassed or incomplete training data can affect the robustness and dependability of vSLAM systems and lead to the generation of inaccurate or unrealistic content. GAN-based synthesis may introduce new artefacts or inconsistencies that affect reconstructed environments. Getting the synthesised content to blend in with the rest of the scene is still a challenge. Large computational resources may be required for both training and synthesis when using GAN models for vSLAM applications in order to quickly train the network and complete the synthesis. This may restrict how much memory and processing power can be allocated to low-resource platforms or applications with strict latency requirements.

III. SIGNAL ENHANCEMENT TECHNIQUES

This study aims to present a thorough examination of the techniques applied to introduce noise and enhance data quality. Various methods were explored to introduce controlled perturbations to the data, such as Gaussian noise [26] and, light noise [27]. Various noise augmentation techniques were applied to enhance the resilience and generalisation of the vSLAM system in challenging scenarios. Through increased accuracy and reliability, does noise-based signal enhancement enhance vSLAM performance? This was investigated experimentally.

A. GAUSSIAN SIGNAL ENHANCEMENT

A Gaussian distribution is statistical noise with the normal distribution's probability density function. Gaussian noise, which is typically defined by its mean and variance, is an arbitrary variation in a digital image's brightness or colour information. When the diversity of the training set is deliberately increased, Gaussian noise has been shown to be a powerful regularisation technique. During this process, a new variation of the original image with varying degrees of random noise is added to the image dataset, making the training dataset more extensive and representative. Even though the machine learning model is forced to learn strong, generalizable feature representations in order to avoid overfitting, the noisy images may appear visually distorted.

It is important to use caution when adding Gaussian noise to a model since it can significantly affect the learning process. Excessive noise can mask key features, which complicates the process of learning the model. A low-noise image might not benefit much from the addition of noise. Therefore, adjusting the Gaussian noise level precisely is essential.

Furthermore, depending on the task at hand and the type of image used, Gaussian noise can have a variety of effects. Excessive noise may cause the model to miss critical features in image-intensive tasks, such as medical imaging [28]. In contrast, noise augmentation can be extremely effective for tasks requiring high-level features.

B. LIGHT SIGNAL ENHANCEMENT

Variations in brightness [29], contrast [30], gamma correction [31], or the addition of random noise, such as Gaussian or speckle noise [32], to simulate sensor noise or specific lighting conditions are all examples of light noise. These enhancements can be used individually or in tandem to create more complex and diverse lighting conditions.

The addition of light noise to the training images forces the model to learn from the primary features of the object rather than relying on specific lighting conditions. This can lead to a more generalized understanding of the objects in the images, which can boost the performance on new, unseen data. The primary challenge in implementing light noise augmentation lies in finding the right balance [33], too little noise and the model may overfit to the training data, failing to generalize to new images [34], too much noise, on the other hand, can risk confusing the model, leading to poor training and validation performance. Therefore, determining the optimal amount and type of light noise is critical and usually achieved through empirical tuning and validation on a held-out dataset.

It is also vital to make sure the noise matches the tasks and images when using light noise augmentation. Because indoor

lighting differs from outdoor lighting, light noise that works well for outdoor photos might not work well for indoor ones. Furthermore, distinct approaches might be needed for tasks like semantic segmentation or object detection compared to image classification or regression.

IV. DE-NOISING TECHNIQUES

A. TOTAL VARIATION (TV)

Total Variation (TV) [35] is a widely used image processing method that reduces noise artifacts while preserving important image details. A mathematical framework that promotes sparsity in gradients is used to restore images corrupted by Gaussian noise.

This method formulates an optimisation problem whose objective is to minimize the variation of gradients in the denoised image while maintaining consistency with the observed noisy image. Iteratively updating the image pixels reduces noise while preserving important image structures.

Rather than exhibiting relatively smooth gradients, edges and boundaries exhibit higher gradient variations, due to the inherent piecewise smoothness property of natural images. By encouraging sparsity in gradient information, this method preserves important image features and reduces noise [36].

Images must be analysed and interpreted accurately in many domains, including medical imaging, computer vision, and remote sensing. This technique enhances image quality without compromising detail, and enables subsequent processing tasks that require clean image data.

B. GAUSSIAN FILTERING

The Gaussian filtering is commonly used in image denoising to suppress noise while preserving image structure. Gaussian filters are linear filters with Gaussian impulse responses for smoothing and reducing noise [37].

Gaussian kernels are convolved with noisy images as spatial low-pass filters. Gaussian filters preserve image detail by attenuating high-frequency noise components. A Gaussian kernel can be smoothed by adjusting its standard deviation parameter, preserving image details while reducing noise [38].

Denoising using Gaussian noise - Gaussian noise is an additive white noise with a Gaussian distribution. Noise is commonly encountered in imaging systems, causing the true image intensity to fluctuate. The Gaussian filter smoothes out this noise, resulting in a cleaner image.

Digital photography, video processing, and computer graphics are all areas in which it can be applied. In particular, when Gaussian noise dominates. Gaussian denoising is a straightforward and computationally efficient method for reducing noise, enhancing image quality, and facilitating subsequent analysis or visual interpretation [39].

C. LIGHT TOTAL VARIATION (TV)

Light-denoising Total Variation (TV) reduces noise in digital images while preserving details, especially in low-light conditions. By promoting sparsity in gradients in images, total variation regularization effectively restores images corrupted by noise [40].

Denoising an image involves finding a denoised image whose gradients are consistent with the observed noisy image while minimizing its total variation. Regularization preserves edges and structures while reducing noise.

Low-light photography, surveillance, and astronomy all benefit from the technique, which captures clear images under challenging lighting conditions. TV uses light denoising to enhance low-light images, improving visibility and facilitating subsequent analysis tasks. When capturing low-light images, it reduces noise while preserving important details.

V. INPAINTING TECHNIQUES TO FILL IN MISSING SPACE AFTER OBJECT REMOVAL

After an object is removed, several techniques can be used to fill in the missing objects in vSLAM. These methods use information from the surrounding environment to reconstruct missing regions.

A suitable technique is chosen based on factors such as the nature of the scene, available sensor data, occlusion level, and output quality. Filling in missing objects in vSLAM applications can be improved by combining different techniques or adapting them to the specific situation.

Background inpainting assumes that the background remains relatively static during object removal. Background pixels are inpainted based on the surrounding context to estimate the missing areas. A variety of algorithms, such as patch-based methods [41], texture synthesis [42], or deep learning-based approaches [43], can be used to accomplish this.

Techniques for completing structures [44] can be used when the removed object had a significant impact on the structure of the scene after it was removed. The geometric and semantic cues that have persisted in the environment can be used to reconstruct the structure of a scene using these techniques.

If the removed object contains important semantic information, such as a person or a vehicle, semantic inpainting techniques can be used. This method uses semantic segmentation or object detection algorithms to fill in a missing object with plausible semantic content [45].

Visual tracking algorithms and a visual tracking and reconstructing technique can be used in certain scenarios to partially reconstruct some missing items, allowing for partial reconstruction without the need for replacements. Based on tracking and analysis of the remaining fragments, the system estimates the trajectory of the object and partially reconstructs its appearance.

Optical flow refers to the pattern of apparent motion of objects in an image or a sequence of images, representing the displacement of pixels between consecutive frames. In VSLAM-based image inpainting, optical flow information is utilized to estimate the motion of objects and camera

viewpoints within a scene. This information is then leveraged to fill in missing or corrupted regions in images during the inpainting process. By understanding how pixels move over time, the algorithm can intelligently interpolate or extrapolate information to reconstruct a coherent and visually consistent representation of the scene [46].

Deep learning, a data-driven approach, has yielded promising results for filling in gaps. GANs and Variational Autoencoders (VAEs) are two generative models that can be trained on large datasets to produce visually consistent and realistic content for missing regions [45].

GANs are deep learning models that consist of a generator and a discriminator. The discriminator attempts to distinguish between real and generated content, whereas the generator attempts to generate as realistic content as possible. Training a GAN on images with and without the object of interest can be used to fill in the empty space after object removal. The generator network learns from the surrounding context to generate plausible content for the missing region, whereas the discriminator network learns from the surrounding context to differentiate between generated and actual images [47].

During the filling-in process, the generator generates content for the missing region based on the context information available. Iteratively, the generated content is refined based on discriminator feedback. The goal is to train the GAN to produce visually consistent and realistic content that blends seamlessly into its surroundings.

The use of GAN-based techniques has shown promising results in the generation of high-quality and visually convincing inpainted regions. GANs can capture complex patterns and textures using deep learning and adversarial training, resulting in more realistic and visually appealing results.

As a result of incorporating GAN-based inpainting into Visual SLAM pipelines, missing objects can be filled-in automatically in real-time or after processing in real-time scenarios that require automatic filling in. By incorporating diverse scene contexts into the reconstruction process and by generating plausible content for the removed objects, data-driven approaches enhance the visual quality and completeness of the reconstructed scene.

VI. EXPERIMENTS AND RESULTS

The purpose of this section is to investigate the robustness and breaking point of our framework (vSLAM based) under Gaussian and Light augmentations. Also, denoising techniques were also explored to enhance the performance of vSLAM under different types of noise. To fill in missing space after object detection and removal, GANs were explored and integrated into vSLAM and YOLOR.

A. TRAINING PHASE AND DATASET

The dataset [46] that was used was a custom dataset that contains RGB-D images of various train stations, videos of the train stations (Outdoors and Indoors), x,y,z data, ground truth data and sequence data. The dataset contains total 16,139 images without data augmentation and 12 video recordings with original image size - 920 × 1080 then resized to – 416 x 416 for object detection and to 64 x 64 for the training of GAN.

Google Collab, a cloud based Jupyter notebook environment that Google provides, was the main platform used to train the GAN. NVIDIA Tesla P100 and NVIDIA Tesla K80 were the GPU models utilised. When it came to programming languages and frameworks, Python was primarily used for creating the GAN architecture and training pipeline, whereas ROS was primarily used for vSLAM. PyTorch, a deep learning framework, was used to construct and train the GAN model. The GAN model was implemented using PyTorch in the Google Colab environment as part of the training environment setup. To make model building and training easier, necessary libraries and dependencies were installed, such as PyTorch and related programmes. The training data, comprising input images with missing regions and corresponding ground truth images, were accessed from cloud storage (Google Drive).

B. EVALUATION METRICS

A vSLAM system's mean error, an important validation metric, measures how close the system's estimated camera poses or mapped points are to their actual positions as determined from ground truth data. A key indicator of the accuracy and precision of the system is this metric.

The Mean Error is typically computed using two common formulas: Root Mean Square Error (RMSE) and Mean Absolute Error (MAE).

1) **RMSE** is calculated by taking the square root of the squared difference between the estimated and ground truth values. By emphasizing larger errors, the formula provides insight into vSLAM's overall precision. The smaller the RMSE, the higher the accuracy and reliability.

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \hat{x}_i)^2} \tag{1}$$

2) **MAE** is computed by averaging the absolute differences between the estimated and ground truth values. Unlike RMSE, MAE does not square the errors, providing an understanding of the average absolute error.

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|x_i - \hat{x}_i| \tag{2}$$

The median error quantifies how far the estimated camera poses or mapped points deviate from the true values. The median error represents the middle value of the errors when sorted ascendingly, as opposed to the mean error, which calculates the average error.

The Median Error can be calculated as follows:

$$\text{Median Error} = \text{Median}\left(|x_1 - \hat{x}_1|, |x_2 - \hat{x}_2|, \ldots, |x_n - \hat{x}_n|\right) \tag{3}$$

where:

- n is the number of data points.
- x_i represents the true value (ground truth) for data point i.
- $\hat{x}_i$ represents the estimated (predicted) value for data point i.

In practice, the median error is less sensitive to outliers compared to mean error metrics like RMSE or MAE. This makes it particularly useful when dealing with datasets that may contain sporadic large errors, which can skew the mean error. A smaller median error suggests that the vSLAM system's estimates are generally close to the ground truth values, making it a valuable indicator of the system's robustness and overall performance.

Relative Pose Error (RPE) is a critical metric for evaluating the accuracy and reliability of vSLAM systems. It is critical in validating the performance of these systems because it quantifies the errors in relative poses (transformations) estimated by the vSLAM algorithm versus ground truth poses, which are typically obtained from external sensors or reference data.

The RPE can be calculated as follows:

$$\text{Relative Pose Error (RPE)} = \frac{1}{n}\sum_{i=1}^{n} \text{error}(T_i, \hat{T}_i) \quad (4)$$

where:

- n is the number of pose pairs in the sequence.
- T_i represents the ground truth transformation for pose i.
- $\hat{T}_i$ represents the estimated transformation for pose i.
- $\text{error}(T_i, \hat{T}_i)$ calculates the error between the ground truth and estimated transformations.

In RRPE, each error term is squared, and then the square root is taken over the mean of these squared errors. RRPE provides a measure of the overall accuracy and precision of the relative pose estimates, giving more weight to larger errors.

The SSIM metric is widely used in vSLAM systems to assess the quality of denoised images. It is important for validating denoising algorithms because it quantifies the similarity between the denoised image and the original, noise-free image. Because SSIM evaluates both structural and luminance information, it is especially effective at capturing the perceptual quality of denoised images.

The SSIM can be calculated as follows:

$$\text{SSIM}(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \quad (5)$$

where:

- x represents the original noise-free image.
- y represents the denoised image.
- μ_x and μ_y are the means of x and y, respectively.
- σ_x and σ_y are the standard deviations of x and y, respectively.
- σ_{xy} is the covariance of x and y.
- c_1 and c_2 are constants introduced to prevent division by zero.

The SSIM formula compares the structural similarity, luminance, and contrast of the denoised image to the original image. A higher SSIM value (ranging from -1 to 1) indicates a better match between the denoised image and the original image, signifying superior denoising performance.

The PSNR is a commonly used metric in vSLAM systems for assessing the quality of denoised images. PSNR is a quantitative measure of how well a denoising algorithm preserves image fidelity by comparing the peak signal level (maximum possible pixel value) to the noise introduced during denoising.

The PSNR can be calculated as follows:

$$\text{PSNR}(x, y) = 10 \cdot \log_{10}\left(\frac{\text{MAX}^2}{\text{MSE}}\right) \quad (6)$$

where:

- x represents the original, noise-free image.
- y represents the denoised image.
- MAX is the maximum possible pixel value (e.g., 255 for an 8-bit image).
- MSE is the Mean Squared Error, which measures the average squared difference between corresponding pixels in x and y.

A higher PSNR value indicates lower noise levels and, consequently, better image quality. It is often measured in decibels (dB). A higher PSNR suggests that the denoising algorithm has effectively reduced noise while preserving the essential features of the image.

C. SIGNAL ENHANCEMENT

1) GAUSSIAN NOISE

The original dataset was first subjected to Gaussian noise with standard deviations of 30, 50, and 70. When we subjected our system to these levels of noise, we discovered that it demonstrated increased robustness in accordance with our hypotheses. Even after noise was added to the data, key performance metrics like Mean, Median, RMSE, and RPE improved, indicating that the generalisation to noisy data was successful. Figure 1 shows a selection of images from the custom dataset, as well as different Guassian noise standard deviations.

In order to determine the breaking point of the system, high levels of standard deviations were applied on the dataset at 100, 150, 200, and 250. Increasing noise levels led to significant performance degradation after 150 standard deviations, which indicated the system was struggling to handle such high-level noise. The system broke as soon as the performance declined beyond 200 standard deviations. These thresholds inform the limitations of a system by determining what levels of noise are acceptable. Figure 2 below shows a sample of images from the custom dataset with added different Guassian noise standard deviations as mentioned above in order to determine when the system will fail.

In Table 1, two vSLAM configurations are compared: ORB-SLAM3 paired with YOLOR and ORB-SLAM3 paired

(a) Original Image (b) STD Deviation Noise = 30 (c) STD Deviation Noise = 50 (d) STD Deviation Noise = 70

FIGURE 1. **Comparison of classification performance under varying gaussian noise levels of 30, 50, and 70 STD deviation.**

TABLE 1. **Gaussian noise with standard deviations of 30, 50, and 70 error comparison.**

Sequences	ORB-SLAM3 / YOLOR				ORB-SLAM3 / YOLORv2			
	Mean	Median	RMSE	RPE%	Mean	Median	RMSE	RPE%
Fr2/xyz_walking	0.0071	0.0065	0.0088	86.12%	0.0058	0.0044	0.0053	87.24%
Fr2/rpy_walking	0.0143	0.0151	0.0152	89.85%	0.0123	0.0128	0.0115	91.51%
Fr2/train_station_walking	0.0539	0.0480	0.0716	84.30%	0.0511	0.0457	0.0681	86.19%
Fr2/signs_walking	0.0489	0.0631	0.0441	83.97%	0.0462	0.0602	0.0418	86.32%
Fr2/xyz_static	0.0356	0.0414	0.0247	54.39%	0.0331	0.0486	0.0204	57.63%
Fr2/rpy_static	0.0095	0.0063	0.0075	46.11%	0.0075	0.0037	0.0039	48.32%
Fr2/train_station_static	0.0495	0.0338	0.0247	28.42%	0.0474	0.0305	0.0193	30.01%
Fr2/signs_static	0.0044	0.0097	0.0098	16.96%	0.0029	0.0064	0.0066	18.58%

(a) STD Deviation Noise = 100 (b) STD Deviation Noise = 150 (c) STD Deviation Noise = 200 (d) STD Deviation Noise = 250

FIGURE 2. **Comparison of classification performance under varying gaussian noise levels of 100, 150 and 250 STD deviation.**

with YOLORv2. In the latter configuration, additional Gaussian noise with standard deviations of 30, 50, and 70 is explicitly introduced. This comparative study incorporates Mean, Median, Root Mean Square Error (RMSE), and Relative Pose Error (RPE%) metrics across multiple sequences such as 'Fr2/xyz_walking', 'Fr2/rpy_walking', 'Fr2/train_station_walking', 'Fr2/signs_walking', 'Fr2/xyz_static', 'Fr2/rpy_static', 'Fr2/train_station_static', and 'Fr2/signs_static'. The first four are high dynamic and the other 4 are low dynamic sequences.

Each sequence represents a unique simulation setting, where 'walking' indicates active motion while 'static' represents a stationary camera position. There are two prefixes, 'xyz' and 'rpy', which are likely to symbolize translation (x, y, z) and rotation (roll, pitch, yaw) of the camera, respectively.

The ORB-SLAM3 and YOLORv2 setup, however, seem to exhibit enhanced accuracy and precision in camera pose estimation despite the addition of Gaussian noise. Although YOLORv2 with added noise may yield less error on average, its performance in maintaining the correct trajectory (RPE) is not uniformly superior.

Due to the absence of motion-induced errors, both setups perform well in 'static' scenarios.

Despite increased noise, the results demonstrate the potential benefits of using updated object detection techniques, like YOLORv2, in vSLAM systems. These systems are resilient to noise, highlighting their robustness. It is, however, necessary to conduct more exhaustive testing to understand their effect on trajectory estimation under different noise levels.

TABLE 2. Gaussian noise with standard deviations of 100, 150, 200 and 250 error comparison.

Sequences	ORB-SLAM3 / YOLOR				ORB-SLAM3 / YOLORv2			
	Mean	Median	RMSE	RPE%	Mean	Median	RMSE	RPE%
Fr2/xyz_walking	0.0071	0.0065	0.0088	86.12%	0.0071	0.0112	0.0129	88.43%
Fr2/rpy_walking	0.0143	0.0151	0.0152	89.85%	0.1104	0.0129	0.0134	92.10%
Fr2/train_station_walking	0.0539	0.0480	0.0716	84.30%	0.0493	0.0478	0.0647	88.19%
Fr2/signs_walking	0.0489	0.0631	0.0441	83.97%	0.0482	0.0571	0.0410	87.77%
Fr2/xyz_static	0.0356	0.0414	0.0247	54.39%	0.0369	0.0417	0.0199	58.61%
Fr2/rpy_static	0.0095	0.0063	0.0075	46.11%	0.0105	0.0102	0.0123	50.22%
Fr2/train_station_static	0.0495	0.0338	0.0247	28.42%	0.0528	0.0347	0.1006	31.59%
Fr2/signs_static	0.0044	0.0097	0.0098	16.96%	0.0073	0.0123	0.0115	20.33%

Table 2 compares the two previous configurations of vSLAM with Gaussian noise with standard deviations of 100, 150, 200, and 250.

Based on the analysis, it was evident that certain sequences exhibit a slight increase in error rates despite having amplified Gaussian noise applied. For instance, the mean error for 'Fr2/rpy_walking' increased significantly from 0.0143 to 0.1104, and the RMSE for 'Fr2/train_station_static' grew from 0.0247 to 0.1006. In these scenarios, the algorithm was pushed to its breaking point by the noise levels, resulting in an increase in error.

For sequences like 'Fr2/xyz_walking', the ORB-SLAM3 / YOLORv2 configuration maintained similar error rates (mean error remaining at 0.0071) despite the increased noise, indicating that the system is resilient and capable of sustaining its performance under more challenging noise conditions.

An interesting observation is that the Relative Pose Error (RPE%) for all the sequences in the ORB-SLAM3 / YOLORv2 configuration showed an increase compared to ORB-SLAM3 / YOLOR, for example in 'Fr2/xyz_walking' from 86.12% to 88.43%. As a result, the trajectory estimation system's robustness appears to be declining under these conditions due to more trajectory drift caused by the Gaussian noise.

As noise levels rise, the performance of YOLORv2 algorithm starts to degrade, potentially indicating the algorithm's breaking point. Overall, the results suggest that YOLORv2 algorithm can withstand a certain amount of noise. In addition to providing insights for future improvements and enhancements of the system, such findings also contribute to its robustness under high noise conditions.

Table 3 represents the execution speed in terms of median time in milliseconds per frame (ms/frame) for different sequences of the ORB-SLAM3 / YOLOR and ORB-SLAM3 / YOLORv2 algorithms. The comparison is done after training with Gaussian data with noise of standard deviations of 30, 50, and 70.

In terms of execution speed, the ORB-SLAM3 / YOLORv2 performs better than the original ORB-SLAM3 / YOLOR, showing higher efficiency. For instance, in the "Fr2/xyz_walking" sequence, the original ORB-SLAM3 / YOLOR algorithm has a median time of 66.45 ms/frame, whereas the improved ORB-SLAM3 / YOLORv2 algorithm performs faster with a median time of 64.03 ms/frame. Similarly, for the "Fr2/rpy_walking" sequence, the ORB-SLAM3 / YOLORv2 shows a median time of 61.36 ms/frame, compared to 64.83 ms/frame in ORB-SLAM3 / YOLOR.

Both dynamic (walking) and static sequences show improvement in performance. Static sequences seem to have a smaller gap between the two algorithms. For example, the "Fr2/signs_static" sequence shows a relatively close median time, with ORB-SLAM3 / YOLOR at 69.43 ms/frame and ORB-SLAM3 / YOLORv2 at 68.82 ms/frame.

As a result, ORB-SLAM3 / YOLORv2 performs better in terms of execution speed than the original ORB-SLAM3 / YOLOR algorithm, even with the addition of Gaussian noise with high standard deviations.

TABLE 3. Execution speed of ORB-SLAM3v2 after training with gaussian data with noise of 30, 50, and 70 standard deviations.

Sequences	Algorithms	
	SLAM3 / YOLOR	SLAM3 / YOLORv2
	Median (ms/frame)	Median (ms/frame)
Fr2/xyz_walking	66.45	64.03
Fr2/rpy_walking	64.83	61.36
Fr2/train_station_walking	65.75	63.78
Fr2/signs_walking	65.91	63.44
Fr2/xyz_static	66.08	64.01
Fr2/rpy_static	66.93	65.82
Fr2/train_station_static	67.24	65.11
Fr2/signs_static	69.43	68.82

Based on Gaussian data with noise of 100, 150, 200, and 250 standard deviations, Table 4 presents the execution speed measured in milliseconds per frame (ms/frame) for a variety of sequences.

Based on analysis, it is evident that ORB-SLAM3 / YOLORv2 consistently outperforms the original ORB-SLAM3 / YOLOR in terms of execution speed across all sequences, indicating its superiority. In the "Fr2/xyz_walking" sequence, the original ORB-SLAM3 / YOLOR algorithm has a median execution speed of 68.12 milliseconds per frame, while the improved ORB-SLAM3 / YOLORv2 algorithm has a median execution time of 66.89 milliseconds/frame. Similarly, ORB-SLAM3 / YOLORv2 demonstrates a significant improvement with a median time of 65.34 ms/frame,

compared with 67.95 ms/frame for the original ORB-SLAM3 / YOLOR.

The efficiency of the algorithm is improved in both dynamic (walking) and static sequences. As for static sequences, however, the difference between these two algorithms is smaller when compared to dynamic sequences. For instance, in the ''Fr2/signs_static'' sequence, the original ORB-SLAM3 / YOLOR shows a median time of 73.52 ms/frame, while the ORB-SLAM3 / YOLORv2 slightly outperforms it with 72.11 ms/frame.

Even when Gaussian noise with high standard deviations is introduced, the ORB-SLAM3 / YOLORv2 algorithm performs better in terms of execution speed. Aspects such as accuracy, resilience, and speed should be considered as part of a comprehensive comparison.

TABLE 4. Execution speed of ORB-SLAM3v2 after training with gaussian data with noise of 100, 150, 200 and 250 standard deviations.

Sequences	Algorithms	
	SLAM3 / YOLOR	SLAM3 / YOLORv2
	Median (ms/frame)	Median (ms/frame)
Fr2/xyz_walking	68.12	66.89
Fr2/rpy_walking	67.95	65.34
Fr2/train_station_walking	69.42	66.78
Fr2/signs_walking	69.01	67.10
Fr2/xyz_static	70.22	68.57
Fr2/rpy_static	70.31	69.27
Fr2/train_station_static	71.02	69.57
Fr2/signs_static	73.52	72.11

2) LIGHT NOISE

Light noise with standard deviations and mean intensities of 50, 100, and 150 was incorporated into the dataset to test robustness. In response to these variations in light intensity, our system showed enhanced resilience. In spite of such light noise, essential performance metrics such as Mean Median RMSE RPE were consistently improved, suggesting an effective adaptability to changes in illumination. An example of the dataset after adding light noise is shown in Figure 3.

To determine the system's breaking point, we escalated the mean intensities and standard deviations of the light noise to 200, 300, 350, and 400 Figure 4. The system struggled to handle such high variations in light intensity from a mean intensity and standard deviation of 300 onwards, as evidenced by the performance metrics. There was a dramatic decline in performance after a mean intensity and standard deviation of 400, which indicates the system basically broke. To understand the system's boundaries, this threshold reveals the upper limit of light noise intensity that can be tolerated for signal enhancement.

Table 5 shows the results of the framework under different levels of light noise and standard deviations of 50, 100, and 150 to determine the algorithm's breaking point under these conditions.

Compared to the ORB-SLAM3 / YOLOR configuration, the ORB-SLAM3 / YOLORv2 configuration shows an increase in mean, median, and RMSE values when light noise is applied. For instance, in the 'Fr2/xyz_walking' sequence, the mean error increased from 0.0071 to 0.0193, and the RMSE from 0.0088 to 0.0337. Under the influence of light noise, error rates are increasing, suggesting that ORB-SLAM3 / YOLORv2 might be approaching breaking point.

A significant difference exists between ORB-SLAM3 / YOLORv2 and ORB-SLAM3 / YOLOR, in that all sequences in the YOLORv2 configuration have a lower RPE than those in the YOLORv3 configuration. Although the vSLAM system's average error rate increases under these conditions, the trajectory is more stable over time, although the overall error percentage is lower.

Despite starting to deteriorate under severe light noise, ORB-SLAM3 / YOLORv2 still maintains a relatively consistent trajectory despite an approach towards its breaking point. It is crucial to perform such stress tests for vSLAM algorithms in order to identify their breaking points and improve their robustness under high noise conditions.

The purpose of Table 6 is to investigate the breaking point of ORB-SLAM3 / YOLORv2 under a wide range of light noise levels, with standard deviations of 200, 300, 350, and 400.

Compared to the ORB-SLAM3 / YOLOR configuration, the application of such high levels of light noise significantly increases mean, median, and RMSE values for the ORB-SLAM3 / YOLORv2 configuration. For example, in the 'Fr2/xyz_walking' scenario, the mean error jumped from 0.0071 to a substantial 0.2268, and the RMSE increased from 0.0088 to 0.2472. During these extreme conditions, the ORB-SLAM3 / YOLORv2 algorithm is approaching or may have even crossed the breaking point.

Comparing ORB-SLAM3 / YOLORv2 with ORB-SLAM3 / YOLOR, RPE for all sequences exhibits a considerable decrease. It appears that despite the light noise dramatically increasing average error rates, the system's trajectory remains relatively consistent over time, but at a significantly lower overall percentage. Under the applied conditions, this is an indication that the system is nearing or crossing its breaking point.

Finally, the performance of the ORB-SLAM3 / YOLORv2 algorithm degrades when subjected to severe light noise. These results highlight the algorithm's limitations in high noise conditions, which can be used to refine the algorithm and improve its robustness and resilience.

It is observed that the ORB-SLAM3 / YOLOR algorithm exhibits a median time of 68.23 ms/frame in the ''Fr2/xyz_walking'' sequence, while the improved ORB-SLAM3 / YOLORv2 algorithm executes the sequence with a median time of 63.51 ms/frame.

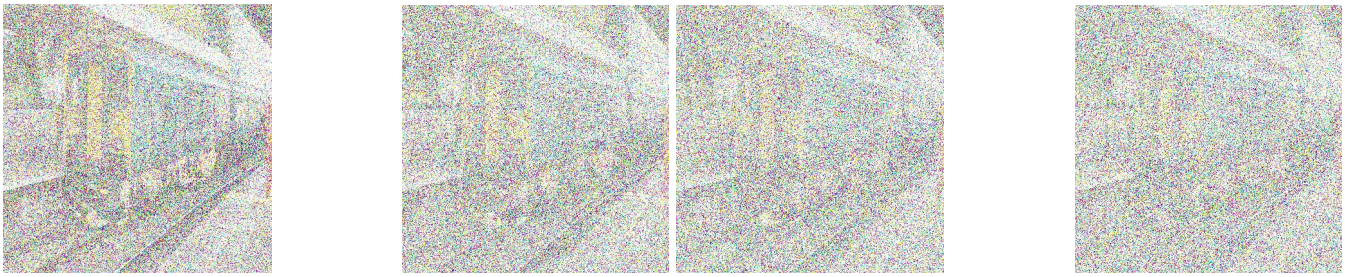
In addition, the ORB-SLAM3 / YOLORv2 algorithm displayed improved performance in the ''Fr2/rpy_walking'' sequence, demonstrating a median time of 61.04 ms/frame

(a) Original Image (b) STD Deviation = 50 (c) STD Deviation = 100 (d) STD Deviation = 150

FIGURE 3. Comparison of classification performance under varying light noise levels of 50, 100 and 150 STD deviation.

TABLE 5. Light noise with standard deviations of 50, 100, and 150 error comparison.

Sequences	ORB-SLAM3 / YOLOR				ORB-SLAM3 / YOLORv2			
	Mean	Median	RMSE	RPE%	Mean	Median	RMSE	RPE%
Fr2/xyz_walking	0.0071	0.0065	0.0088	86.12%	0.0193	0.0366	0.0337	76.01%
Fr2/rpy_walking	0.0143	0.0151	0.0152	89.85%	0.0714	0.1094	0.1103	76.84%
Fr2/train_station_walking	0.0539	0.0480	0.0716	84.30%	0.1059	0.1159	0.1249	71.94%
Fr2/signs_walking	0.0489	0.0631	0.0441	3.97%	0.0750	0.1355	0.1093	73.11%
Fr2/xyz_static	0.0356	0.0414	0.0247	54.39%	0.0958	0.1642	0.0774	47.32%
Fr2/rpy_static	0.0095	0.0063	0.0075	46.11%	0.0305	0.0458	0.0390	36.49%
Fr2/train_station_static	0.0495	0.0338	0.0247	28.42%	0.1294	0.1201	0.0742	16.20%
Fr2/signs_static	0.0044	0.0097	0.0098	16.96%	0.0337	0.0301	0.0495	8.95%



(a) STD Deviation = 200 (b) STD Deviation = 300 (c) STD Deviation = 350 (d) STD Deviation = 400

FIGURE 4. Comparison of classification performance under varying light noise levels of 200, 300, 350 and 400 STD deviation.

TABLE 6. Light noise with standard deviations of 200, 300, 350 and 400 error comparison.

Sequences	ORB-SLAM3 / YOLOR				ORB-SLAM3 / YOLORv2			
	Mean	Median	RMSE	RPE%	Mean	Median	RMSE	RPE%
Fr2/xyz_walking	0.0071	0.0065	0.0088	86.12%	0.2268	0.2356	0.2472	37.11%
Fr2/rpy_walking	0.0143	0.0151	0.0152	89.85%	0.2691	0.2594	0.2697	36.19%
Fr2/train_station_walking	0.0539	0.0480	0.0716	84.30%	0.2523	0.2810	0.2713	35.42%
Fr2/signs_walking	0.0489	0.0631	0.0441	83.97%	0.2850	0.2958	0.2849	33.27%
Fr2/xyz_static	0.0356	0.0414	0.0247	54.39%	0.3074	0.3227	0.2981	32.24%
Fr2/rpy_static	0.0095	0.0063	0.0075	46.11%	0.3362	0.3391	0.3144	13.64%
Fr2/train_station_static	0.0495	0.0338	0.0247	28.42%	0.3477	0.2313	0.3275	1.28%
Fr2/signs_static	0.0044	0.0097	0.0098	16.96%	0.1168	0.1269	1.0596	0.34%

as opposed to 67.90 ms/frame in the original ORB-SLAM3 / YOLOR algorithm.

Performance has improved across static and dynamic sequences (walking). Differences between the two algorithms are smaller, however, in static low dynamic sequences. As an example, ORB-SLAM3 / YOLORv2 outperforms the original algorithm with a median time of 69.77 ms/frame in the "Fr2/signs_static" sequence. In summary, the

ORB-SLAM3 / YOLORv2 algorithm performs better than the original ORB-SLAM3 / YOLOR algorithm despite varying levels of light noise.

TABLE 7. Execution speed of ORB-SLAM3v3 after training with 50, 100, and 150 light noise.

Sequences	Algorithms	
	SLAM3 / YOLOR	SLAM3 / YOLORv2
	Median (ms/frame)	Median (ms/frame)
Fr2/xyz_walking	68.23	63.51
Fr2/rpy_walking	67.90	61.04
Fr2/train_station_walking	67.93	65.37
Fr2/signs_walking	68.68	64.14
Fr2/xyz_static	69.29	65.11
Fr2/rpy_static	67.92	66.20
Fr2/train_station_static	69.11	67.98
Fr2/signs_static	72.26	69.77

D. DE-NOISING TECHNIQUES

In the exploration phase of our research, we scrutinized various noise denoising techniques tailored to the specific types of noise augmentation - Gaussian, and light noise. For Gaussian noise, we evaluated techniques including Gaussian filtering, median filtering, wavelet denoising, non-local means filtering, and total variation denoising. After a thorough assessment, only Gaussian filtering and total variation denoising were selected for our experiments due to their efficacy in maintaining data continuity while effectively reducing the noise component and preserving the edges and structural integrity of data, respectively.

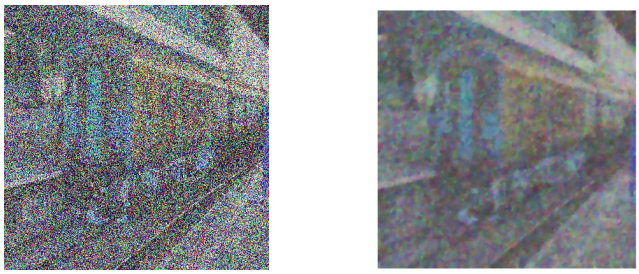
Lastly, for light noise, we assessed various techniques, including Poisson noise reduction, non-local means filtering, total variation denoising, wavelet denoising, and maximum likelihood estimation. Here, total variation denoising emerged as the sole choice due to its superior proficiency in handling variations in light intensity noise while preserving the edges and details of images.

In sum, our choice of denoising techniques - Gaussian filtering and total variation denoising for Gaussian noise, Wiener deconvolution and total variation denoising for light noise - highlights a mindful, evidence-driven approach, tailored to align with the specific characteristics and requirements of our dataset.

1) GAUSSIAN DENOISING – TOTAL VARIATION (TV)

Gaussian Denoising – TV was applied on the image dataset represented by Figure 5, an innovative approach aimed at mitigating the effect of noise within these images. The main objective of implementing this noise reduction technique is to enhance the visual quality and accuracy of the images that are fundamental to the performance of vSLAM. vSLAM was then train and tested with these denoised images, aiming to examine the impact of Gaussian Denoising – TV on the overall performance of the system.

Eight different scenarios are compared in Table 8. The main evaluation metrics for this evaluation was the total variation error, represented through measures such as the



(a) Noisy Image (b) Denoised Image

FIGURE 5. Gaussian denoising – Total Variation (TV), noisy and denoised image.

Structural Similarity Index Measure (SSIM), Peak Signal to Noise Ratio (PSNR), and Relative Pose Error (RPE). It appears that our denoising techniques have an impact on the vSLAM framework's performance.

The SSIM results show structural similarities between pre- and post-denoising images. Fr2/rpy_walking shows the highest SSIM value at 0.9472, indicating excellent structural detail preservation. A denoised SSIM of 0.3225 for 'Fr2/signs_static' indicates some structural changes.

The PSNR measures denoised image quality by comparing the maximum power of the signal with the corrupting noise. With a PSNR of 41.2317 dB, the sequence 'Fr2/rp_walking' achieves the best denoising quality, indicating the lowest amount of noise. At 20.3251 dB, the PSNR of the 'Fr2/signs_static' sequence is the lowest, suggesting that the image has more noise.

RPE is used to measure movement accuracy over time in vSLAM. After applying Gaussian denoising to the Fr2/rpy_walking sequence, we achieve RPE of 93.66% after the initial RPE was 92.10%. As a result, the precision of trajectory estimation in vSLAM has increased. Based on the initial and improved RPE results of the sequence 'Fr2/signs_static', 20.33 and 22.12 are the lowest.

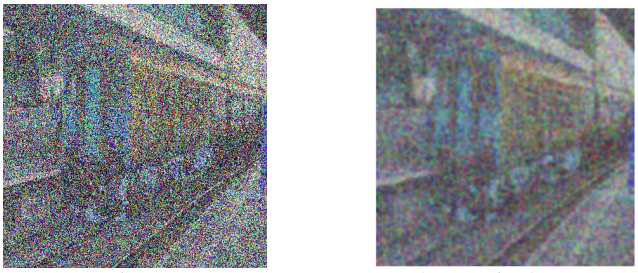
TABLE 8. Gaussian denoising – total variation error comparison.

Sequences	SLAM3 / YOLORv2			
	SSIM	PSNR	RPE Orig	Improve
Fr2/xyz_walking	0.9211	32.3408	88.43%	89.54%
Fr2/rpy_walking	0.9472	41.2317	92.10%	93.66%
Fr2/train_station_walking	0.9134	31.8691	88.19%	89.79%
Fr2/signs_walking	0.9029	30.5930	87.77%	89.23%
Fr2/xyz_static	0.7263	26.2951	58.61%	60.43%
Fr2/rpy_static	0.6524	24.4511	50.22%	52.98%
Fr2/train_station_static	0.4524	22.2795	31.59%	33.36%
Fr2/signs_static	0.3225	20.3251	20.33%	22.12%

2) GAUSSIAN DENOISING – GAUSSIAN FILTERING

Furthermore, Gaussian Filtering was applied on the dataset to further test the effectiveness of the denoising technique, and to compare TV with Gaussian Filtering. This process, typically used to reduce detail and noise, is applied to improve the effectiveness of vSLAM Figure 6. In order to evaluate the influence of Gaussian Filtering on our vSLAM system's

overall performance, a vSLAM model was traine with these filtered images.



(a) Noisy Image (b) Denoised Image

FIGURE 6. Gaussian denoising – gaussian filtering, noisy and denoised image.

Table 9 presents the error comparison results for Gaussian denoising and Gaussian filtering on various ORB-SLAM3 / YOLORv2 sequences.

SSIM value for 'Fr2/rpy_walking' sequence is 0.9295 after Gaussian filtering, suggesting excellent structural integrity retention. A SSIM of 0.2787 shows a loss of structural detail in the 'Fr2/signs_static' sequence.

A noise reduction process of efficiency is demonstrated by the sequence 'Fr2/rpy/walking', which has the highest PSNR and demonstrates the best noise reduction process at 40.3397 dB. The sequence 'Fr2/signs_static' records the lowest PSNR at 19.0541 dB, showing more noise-affected image post-processing.

After Gaussian filtering, the RPE for the 'Fr2/rpy_walking' sequence improved to 92.21% from 92.10%. 'Fr2/signs_static' has the lowest original and improved RPE, which is 20.33 and 21.59.

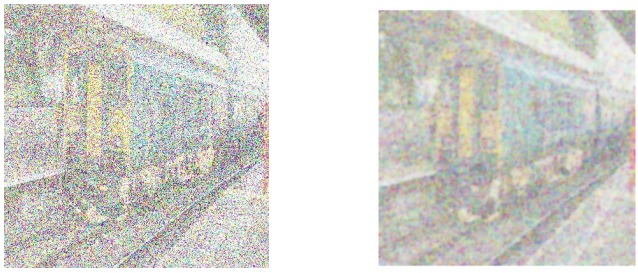
TABLE 9. Gaussian denoising – gaussian filtering error comparison.

Sequences	SLAM3 / YOLORv2			
	SSIM	PSNR	RPE Orig	Improve
Fr2/xyz_walking	0.9123	31.8394	88.43%	88.89%
Fr2/rpy_walking	0.9295	40.3397	92.10%	92.21%
Fr2/train_station_walking	0.8921	30.7901	88.19%	88.53%
Fr2/signs_walking	0.8864	29.1350	87.77%	88.67%
Fr2/xyz_static	0.7072	24.5319	58.61%	59.22%
Fr2/rpy_static	0.6367	23.1401	50.22%	51.40%
Fr2/train_station_static	0.3462	20.8923	31.59%	32.75%
Fr2/signs_static	0.2787	19.0541	20.33%	21.59%

3) LIGHT DENOISING – TOTAL VARIATION

In addition, Light Denoising - TV was applied to the custom image dataset to mitigate light-induced noise. Known for its ability to preserve edges while smoothing out noise, this method is essential for enhancing the performance of vSLAM. By implementing and analysing Light Denoising - TV, we explored its potential for strengthening vSLAM performance under varying light conditions. The results of this approach are demonstrated in Table 7.

Using this denoising method, Table 10 compares error rates for several ORB-SLAM3 / YOLORv2 sequences.



(a) Noisy Image (b) Denoised Image

FIGURE 7. Light denoising – Total Variation (TV), noisy and denoised image.

The highest SSIM value in the 'Fr2/rpy_walking' sequence is 0.9122, indicating superior preservation of structural information. After denoising, 'Fr2/signs_static' registers a SSIM of 0.2510, showing an overall loss of structural details.

The highest PSNR value is recorded in the 'Fr2/rpy_walking' sequence at 38.6434 dB, suggesting effective noise reduction and high-quality denoising. PSNR is found to be the lowest in the sequence 'Fr2/signs_static' at 17.7923 dB, suggesting that the level of noise is higher

Based on the 'Fr2/rpy_walking' sequence, an initial RPE of 89.34% is observed, which improves to 91.51% after applying Light Denoising, suggesting improved trajectory estimation accuracy. There is the lowest initial and improved RPE in the 'Fr2/signs_static' sequence, which are 16.28% and 18.87%, respectively.

Light Denoising with Total Variation improves trajectory accuracy and image quality in vSLAM performance using Light Denoising with Total Variation. Denoising demonstrates effectiveness by observing improvements across SSIM, PSNR, and RPE values across sequences, albeit to varying degrees.

TABLE 10. Light denoising – total variation.

Sequences	SLAM3 / YOLORv2			
	SSIM	PSNR	RPE Orig	Improve
Fr2/xyz_walking	0.9016	30.0531	84.13%	85.79%
Fr2/rpy_walking	0.9122	38.6434	89.34%	91.51%
Fr2/train_station_walking	0.8704	28.6952	85.60%	87.42%
Fr2/signs_walking	0.8591	27.5331	83.71%	85.13%
Fr2/xyz_static	0.6845	22.9526	55.24%	57.68%
Fr2/rpy_static	0.6123	21.5294	47.11%	49.21%
Fr2/train_station_static	0.3207	18.5924	37.38%	39.64%
Fr2/signs_static	0.2510	17.7923	16.28%	18.87%

E. GANs COMPARED TO OPTICAL FLOW FOR IMAGE INPAINTING

The GAN architecture Figure 8 is comprised of two main modules: a generator and a discriminator. This model [48] generates synthetic images of an indoor train station given an input random latent vector using an input random latent vector.

Generator is a CNN that generates a synthetic RGB image from a noise vector (of dimensions LATENT_DIM $\times$ 1 $\times$ 1).

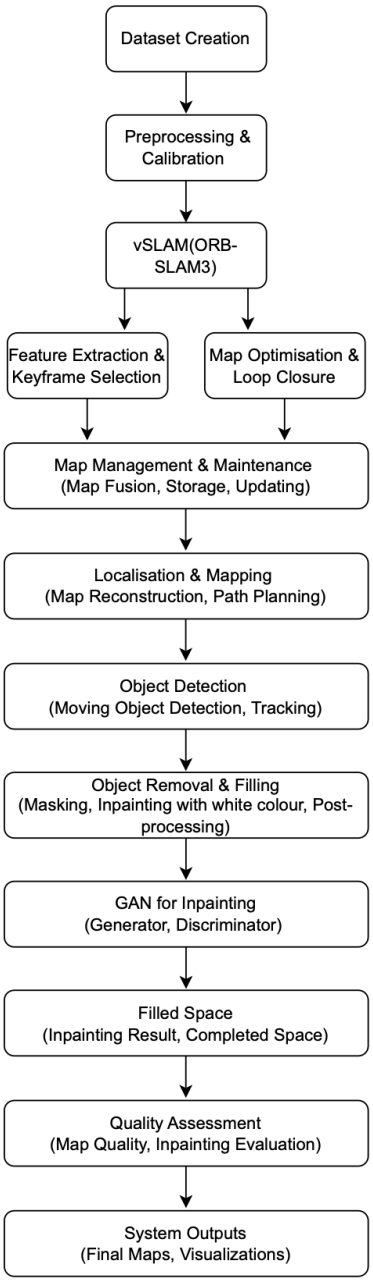


FIGURE 8. **Overall system architecture.**

The generator model's architecture can be summarised as follows:

1) This layer maps the input noise vector into a 512-channel feature map of 4 × 4 dimensions using transposed convolution.
2) ReLU activation function and Batch Normalization.
3) A series of Transposed Convolution, Batch Normalisation, and ReLU layers are used to increase the spatial dimension of the feature maps. The number of channels in each layer is reduced while the spatial dimension is increased.
4) Using a Tanh activation function to bound the output values between -1 and 1, the final Transposed Convolution layer maps the feature maps to a RGB image (3 channels) of size 64 × 64.

Using convolutional neural networks, the Discriminator distinguishes between real and synthetic images. A scalar output represents the network's confidence that the input image is real based on an RGB image (real or synthetic). Following is the architecture:

1) A Convolution layer that is responsible to map the input images to a 128-channel feature map.
2) LeakyReLU activation function.
3) Using Convolution, Batch Normalization, and LeakyReLU layers, the spatial dimensions of the feature maps are decreased while the number of channels is increased.
4) To reduce the spatial dimension to 1 × 1, an Adaptive Average Pooling layer is applied.
5) An Adaptive Average Pooling layer is applied to reduce the spatial dimension to 1 × 1.
6) To map the feature vector to a single scalar value, a linear layer was used.
7) The output values are bound between 0 and 1 using a sigmoid activation function.

To update the weights of the generator and discriminator, the Adam optimization algorithm is used to train the GAN model using binary cross-entropy loss (BCE). There are two phases to the training process:

1) **Discriminator training phase:** To maximize the likelihood of correctly classifying synthetic and real images, the discriminator's weights are updated. The BCE losses for real and synthetic images are minimized by minimizing their sum.
2) **Generator training phase:** As part of this phase, the generator's weights are updated to maximize the log-likelihood that the discriminator will classify its synthetic images as real. The synthetic images are minimized by minimizing the BCE loss.

The synthetic images generated by the generator are saved for later evaluation in every training epoch. A set number of epochs or another stopping condition is reached at the end of the training process.

Table 11 shows the evaluation metrics of the overall framework of vSLAM and GAN for mage inpainting. The high SSIM of 0.9147 denotes a remarkable preservation of structural information between the inpainted and original images, emphasizing the fidelity achieved by the inpainting process. The PSNR of 31.6943 dB further corroborates the exceptional quality of the reconstructed images, highlighting minimal loss during the inpainting operation. The MSEE value of 410 reinforces the accuracy of inpainting, indicating the closeness of the generated images to the ground truth. Frechet Inception Distance (FID) is a metric for quantifying the realism and diversity of images generated by GANs. Moving beyond image quality, the FID of 23 suggests a reasonable disparity between the generated and real images,

(a) Original Image (b) After Object Removal (c) GAN for Image Inpainting

FIGURE 9. Image comparison after image inpainting.

(a) Original Image (b) After Object Removal (c) GAN for Image Inpainting

FIGURE 10. Image comparison after image inpaintng.

indicating a balance between diversity and realism in the inpainting results. In the realm of vSLAM, the reported pose accuracy of 0.21 meters attests to the system's capability to accurately localize itself within the environment. The high map consistency score of 0.89 signifies a robust and coherent mapping of the surroundings, crucial for applications like autonomous navigation. Furthermore, the vSLAM loop closure detection rate of 88% underscores the system's proficiency in recognizing and closing loops, enhancing long-term localisation reliability. The trajectory error of 4% indicates the precision with which the system tracks its movement through space, showcasing a low margin of error. Notably, the impressive execution time of 0.3 seconds for the inpainting process highlights the efficiency of the entire framework, rendering it suitable for real-time applications. Finally, Figure 9 and Figure 10 shows the original image, the image after object removal and the picture after the use of GAN for image inpainting.

VII. DISCUSSION

A thorough examination of various types of noise and their effects on vSLAM system performance is provided, as is a discussion of several denoising techniques and their effects on image quality.

TABLE 11. Performance metrics.

Metric	Value
SSIM	0.9147
PSNR	31.6943 dB
MSE	410
FID	23
vSLAM Pose Accuracy	0.21 meters
vSLAM Map Consistency	0.89
vSLAM Loop Closure Detection	88%
vSLAM Trajectory Error	4%
Inpainting Execution Time	0.3 seconds

A Gaussian noise and light noise were initially investigated for their impacts on system performance. The robustness and vulnerability of the system are affected by each type of noise. A clear and intricate balance exists between the level of noise and system performance across these variations. By enhancing data with Gaussian noise, for instance, insights into the relationship between data noise and performance can be gained. Light noise signal enhancement helps us understand how light noise intensity impacts system performance. Effective and efficient noise management strategies are necessary to manage these complex relationships.

A variety of denoising techniques were evaluated to address noise-induced performance variations, including Gaussian filtering and MAP estimation denoising. SSIM, PSNR, RPE, and RMSE have shown varying degrees of improvement with each of these techniques.

According to the findings, effective denoising techniques can significantly improve the performance of vSLAM systems by managing noise. Given the variability in the results across different sequences, these techniques should be carefully chosen and tailored to the specific type and degree of noise. More research is needed to ensure that these denoising methods achieve optimal performance across all metrics.

While GANs can require a significant amount of processing power and memory during the training phase, the inference process—which is where the trained GAN creates new images—usually includes much less computational overhead. When it comes to resource-constrained devices like smartphones or tablets, GAN inference may often be completed effectively, even in real-time or almost real-time, thanks to developments in hardware acceleration techniques and optimisations [49]. As a result, even if the initial training of GANs would be difficult in these kinds of settings, it might still be possible to use a trained GAN for image inpainting in the vSLAM system as long as the right optimisations are made to lessen computational limitations. By leveraging the efficiency of GAN inference, the integration of these advanced techniques into the vSLAM framework may still offer substantial benefits without unduly compromising the system's performance or applicability in resource-constrained settings.

The suggested system's practical application is intended to address issues that arise in indoor dynamic situations, with a concentration on train stations because of their high population density. The need to improve older people's safety and navigational efficiency in such busy surroundings serves as the foundation for this approach. Nevertheless, a variety of issues, such as the existence of several people and obstructions, which complicate scene perception and tracking, hinder the effectiveness of vSLAM in these situations. Moreover, the fluctuations in illumination additionally impact the performance constraints of VSLAM algorithms.

VIII. CONCLUSION

vSLAM performance is significantly impacted by signal enhancement techniques, including Gaussian noise and light noise. In our study, GANs are integrated with vSLAM and object detection models and used to address the task of removing objects from spaces and filling in the gaps.

With Gaussian and light noise, vSLAM has demonstrated a significant ability to improve robustness, generalisation, and performance. Their effectiveness is still dependent on a careful balance of these technologies' application and an understanding of the characteristics of the specific tasks and images.

Gaussian noise augmentation, for example, have demonstrated great promise in improving model performance and managing the complexity of indoor dynamic real-world environments. A combination of light noise augmentation and task-image consistency is also a powerful tool for enhancing model generalisation, despite its simplicity.

Our study's real breakthrough, however, is the unprecedented combination of GANs and vSLAM. We developed a mechanism to fill in missing spaces caused by object removal by leveraging the generative capabilities of GANs, a challenge not adequately addressed in previous studies. This method has opened up new possibilities for dealing with complex real-world imagery and has the potential to significantly improve the overall performance of vSLAM systems.

In summary, promising results for achieving enhanced performance in handling intricate real-world imagery is the integration of conventional signal enhancement methods with the innovative use of GANs. These approaches have the potential to greatly increase the robustness and performance of machine learning models, especially in the context of vSLAM, but their careful application and continuous improvement are necessary. This creates exciting new opportunities for future research and development.

REFERENCES

[1] X. Fang, Q. Li, J. Zhu, Z. Chen, D. Zhang, K. Wu, K. Ding, and Q. Li, "Sewer defect instance segmentation, localization, and 3D reconstruction for sewer floating capsule robots," *Autom. Construct.*, vol. 142, Oct. 2022, Art. no. 104494.

[2] O. F. Kar, T. Yeo, A. Atanov, and A. Zamir, "3D common corruptions and data augmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 18941–18952.

[3] A. Sabater, L. Montesano, and A. C. Murillo, "Event transformer. A sparse-aware solution for efficient event data processing," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2022, pp. 2676–2685.

[4] K. Maharana, S. Mondal, and B. Nemade, "A review: Data pre-processing and data augmentation techniques," *Global Transitions Proc.*, vol. 3, no. 1, pp. 91–99, Jun. 2022.

[5] A. Bansal, E. Borgnia, H.-M. Chu, J. S. Li, H. Kazemi, F. Huang, M. Goldblum, J. Geiping, and T. Goldstein, "Cold diffusion: Inverting arbitrary image transforms without noise," 2022, *arXiv:2208.09392*.

[6] A. Rosinol, J. J. Leonard, and L. Carlone, "NeRF-SLAM: Real-time dense monocular SLAM with neural radiance fields," 2022, *arXiv:2210.13641*.

[7] X. Gao, "A method for face image inpainting based on generative adversarial networks," Doctoral dissertation, Auckland Univ. Technol., Auckland, New Zealand, 2022.

[8] A. You, J. K. Kim, I. H. Ryu, and T. K. Yoo, "Application of generative adversarial networks (GAN) for ophthalmology image domains: A survey," *Eye Vis.*, vol. 9, no. 1, pp. 1–19, Dec. 2022.

[9] J. He, W. Shi, K. Chen, L. Fu, and C. Dong, "GCFSR: A generative and controllable face super resolution method without facial and GAN priors," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 1879–1888.

[10] Y. Li, Z. Zhao, J. Fan, and W. Li, "ADR-MVSNet: A cascade network for 3D point cloud reconstruction with pixel occlusion," *Pattern Recognit.*, vol. 125, May 2022, Art. no. 108516.

[11] H. Wang, Y. Chen, Y. Cai, L. Chen, Y. Li, M. A. Sotelo, and Z. Li, "SFNet-N: An improved SFNet algorithm for semantic segmentation of low-light autonomous driving road scenes," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 11, pp. 21405–21417, Nov. 2022.

[12] C. Theodorou, V. Velisavljevic, V. Dyo, and F. Nonyelu, "Visual SLAM algorithms and their application for AR, mapping, localization and wayfinding," *Array*, vol. 15, Sep. 2022, Art. no. 100222.

[13] M. Xu, S. Yoon, A. Fuentes, and D. S. Park, "A comprehensive survey of image augmentation techniques for deep learning," *Pattern Recognit.*, vol. 137, May 2023, Art. no. 109347.
[14] N. Merrill, Y. Guo, X. Zuo, X. Huang, S. Leutenegger, X. Peng, L. Ren, and G. Huang, "Symmetry and uncertainty-aware object SLAM for 6DoF object pose estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 14881–14890.
[15] H. Seo, H. Kim, G. Kim, and S. Young Chun, "DITTO-NeRF: Diffusion-based iterative text to omni-directional 3D model," 2023, *arXiv:2304.02827*.
[16] Y. Tian, H. Yue, B. Yang, and J. Ren, "Unmanned aerial vehicle visual simultaneous localization and mapping: A survey," *J. Phys., Conf. Ser.*, vol. 2278, no. 1, May 2022, Art. no. 012006.
[17] R. Duan, Y. Feng, and C.-Y. Wen, "Deep pose graph-matching-based loop closure detection for semantic visual SLAM," *Sustainability*, vol. 14, no. 19, p. 11864, Sep. 2022.
[18] I. Abaspur Kazerouni, L. Fitzgerald, G. Dooly, and D. Toal, "A survey of state-of-the-art on visual SLAM," *Expert Syst. Appl.*, vol. 205, Nov. 2022, Art. no. 117734.
[19] S. Phadikar, N. Sinha, R. Ghosh, and E. Ghaderpour, "Automatic muscle artifacts identification and removal from single-channel EEG using wavelet transform with meta-heuristically optimized non-local means filter," *Sensors*, vol. 22, no. 8, p. 2948, Apr. 2022.
[20] S. N. S. S. Daud and R. Sudirman, "Wavelet based filters for artifact elimination in electroencephalography signal: A review," *Ann. Biomed. Eng.*, vol. 50, no. 10, pp. 1271–1291, Oct. 2022.
[21] R. Ramm, M. Heinze, P. Kühmstedt, A. Christoph, S. Heist, and G. Notni, "Portable solution for high-resolution 3D and color texture on-site digitization of cultural heritage objects," *J. Cultural Heritage*, vol. 53, pp. 165–175, Jan. 2022.
[22] Z. Yan, K. Wang, X. Li, Z. Zhang, J. Li, and J. Yang, "RigNet: Repetitive image guided network for depth completion," in *Proc. Eur. Conf. Comput. Vis.*, Cham, Switzerland. Cham: Springer, 2022, pp. 214–230.
[23] D. Cirillo, F. Cerritelli, S. Agostini, S. Bello, G. Lavecchia, and F. Brozzetti, "Integrating post-processing kinematic (PPK)–structure-from-motion (SfM) with unmanned aerial vehicle (UAV) photogrammetry and digital field mapping for structural geological analysis," *ISPRS Int. J. Geo-Inf.*, vol. 11, no. 8, p. 437, Aug. 2022.
[24] J. Zhang, Y. Liu, C. Guo, and J. Zhan, "Optimized segmentation with image inpainting for semantic mapping in dynamic scenes," *Int. J. Speech Technol.*, vol. 53, no. 2, pp. 2173–2188, Jan. 2023.
[25] P. Kulshreshtha, N. Lianos, B. Pugh, and S. Jiddi, "Layout aware inpainting for automated furniture removal in indoor scenes," in *Proc. IEEE Int. Symp. Mixed Augmented Reality Adjunct (ISMAR-Adjunct)*, Oct. 2022, pp. 839–844.
[26] N. E. Khalifa, M. Loey, and S. Mirjalili, "A comprehensive survey of recent trends in deep learning for digital images augmentation," *Artif. Intell. Rev.*, vol. 55, no. 3, pp. 2351–2377, Mar. 2022.
[27] S. Zhang, S. Zhao, D. An, J. Liu, H. Wang, Y. Feng, D. Li, and R. Zhao, "Visual SLAM for underwater vehicles: A survey," *Comput. Sci. Rev.*, vol. 46, Nov. 2022, Art. no. 100510.
[28] A. Kazerouni, E. K. Aghdam, M. Heidari, R. Azad, M. Fayyaz, I. Hacihaliloglu, and D. Merhof, "Diffusion models for medical image analysis: A comprehensive survey," 2022, *arXiv:2211.07804*.
[29] C. Fu, H. Dong, J. Ye, G. Zheng, S. Li, and J. Zhao, "HighlightNet: Highlighting low-light potential features for real-time UAV tracking," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2022, pp. 12146–12153.
[30] I. Rubinoff, D. A. Miller, R. Kuranov, Y. Wang, R. Fang, N. J. Volpe, and H. F. Zhang, "High-speed balanced-detection visible-light optical coherence tomography in the human retina using subpixel spectrometer calibration," *IEEE Trans. Med. Imag.*, vol. 41, no. 7, pp. 1724–1734, Jul. 2022.
[31] G. Ramesh, J. Logeshwaran, J. Gowri, and A. Mathew, "The management and reduction of digital noise in video image processing by using transmission based noise elimination scheme," *ICTACT J. Image Video Process.*, vol. 13, no. 1, pp. 1–5, 2022.
[32] Y. Jin, R. Dollevoet, and Z. Li, "Numerical simulation and characterization of speckle noise for laser Doppler vibrometer on moving platforms (LDVom)," *Opt. Lasers Eng.*, vol. 158, Nov. 2022, Art. no. 107135.
[33] X. Liu, W. Ma, X. Ma, and J. Wang, "LAE-Net: A locally-adaptive embedding network for low-light image enhancement," *Pattern Recognit.*, vol. 133, Jan. 2023, Art. no. 109039.
[34] H. Wu, B. Zhang, and N. Liu, "Self-adaptive denoising net: Self-supervised learning for seismic migration artifacts and random noise attenuation," *J. Petroleum Sci. Eng.*, vol. 214, Jul. 2022, Art. no. 110431.
[35] Q. Ma, Y. Wang, and T. Zeng, "Retinex-based variational framework for low-light image enhancement and denoising," *IEEE Trans. Multimedia*, vol. 25, pp. 5580–5588, 2022.
[36] Y. Bai, X. Chen, A. Kirillov, A. Yuille, and A. C. Berg, "Point-level region contrast for object detection pre-training," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 16040–16049.
[37] R. Li and B. Zheng, "A spatially adaptive hybrid total variation model for image restoration under Gaussian plus impulse noise," *Appl. Math. Comput.*, vol. 419, Apr. 2022, Art. no. 126862.
[38] L. Zhang, Y. Qian, J. Han, P. Duan, and P. Ghamisi, "Mixed noise removal for hyperspectral image with l_0-l_{1-2}SSTV regularization," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 15, pp. 5371–5387, 2022.
[39] E. Goceri, "Evaluation of denoising techniques to remove speckle and Gaussian noise from dermoscopy images," *Comput. Biol. Med.*, vol. 152, Jan. 2023, Art. no. 106474.
[40] H. Cao, X. Gu, M. Zhang, H. Zhang, and X. Chen, "Vignetting correction based on a two-dimensional Gaussian filter with harmony for area array sensors," *IEEE Trans. Comput. Imag.*, vol. 8, pp. 576–584, 2022.
[41] X. Li, "Real-time denoising enables high-sensitivity fluorescence time-lapse imaging beyond the shot-noise limit," *Nature Biotechnol.*, vol. 41, no. 2, pp. 282–292, Feb. 2023.
[42] H. Feng, L. Wang, Y. Wang, and H. Huang, "Learnability enhancement for low-light raw denoising: Where paired real data meets noise modeling," in *Proc. 30th ACM Int. Conf. Multimedia*, Oct. 2022, pp. 1436–1444.
[43] Q. Liu, Z. Tan, D. Chen, Q. Chu, X. Dai, Y. Chen, M. Liu, L. Yuan, and N. Yu, "Reduce information loss in transformers for pluralistic image inpainting," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 11337–11347.
[44] J. Jain, Y. Zhou, N. Yu, and H. Shi, "Keys to better image inpainting: Structure and texture go hand in hand," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2023, pp. 208–217.
[45] H. Xiang, Q. Zou, M. A. Nawaz, X. Huang, F. Zhang, and H. Yu, "Deep learning for image inpainting: A survey," *Pattern Recognit.*, vol. 134, Feb. 2023, Art. no. 109046.
[46] C. Theodorou, V. Velisavljevic, and V. Dyo, "Visual SLAM for dynamic environments based on object detection and optical flow for dynamic object removal," *Sensors*, vol. 22, no. 19, p. 7553, Oct. 2022.
[47] R. Shah, A. Gautam, and S. K. Singh, "Overview of image inpainting techniques: A survey," in *Proc. IEEE Region 10 Symp. (TENSYMP)*, Jul. 2022, pp. 1–6.
[48] Y. Yu, L. Zhang, H. Fan, and T. Luo, "High-fidelity image inpainting with GAN inversion," in *Proc. Eur. Conf. Comput. Vis.*, 2022, pp. 242–258.
[49] H. Jia, Q. Wang, O. Tov, Y. Zhao, F. Deng, L. Wang, C.-L. Chang, T. Hou, and M. Grundmann, "BlazeStyleGAN: A real-time on-device StyleGAN," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2023, pp. 4689–4693.

CHARALAMBOS THEODOROU received the bachelor's degree in computer science from Kingston University London, Kingston, U.K., in 2019, and the master's degree in data science from the University of Bath, Bath, U.K., in 2020. He is currently pursuing the Ph.D. degree in computer vision with the University of Bedfordshire, U.K. He is currently a Machine Learning Engineer with Briteyellow Ltd. His research interests include computer vision, machine learning, data science, SLAM, and AR.

VLADAN VELISAVLJEVIC (Senior Member, IEEE) received the Ph.D. degree in signal processing from EPFL, Switzerland, in 2005. He was a Senior Research Scientist with Deutsche Telekom Laboratories, Technical University of Berlin, Germany, from 2006 to 2011. He has been an Associate Professor and a Professor with the University of Bedfordshire, since 2011 and April 2023, respectively. His research interests include signal, image, and video compression and enhancement, multiple and free viewpoint imaging and video, interactive video transmission, wavelet theory, multi-resolution signal representation, and machine learning for signal recognition and classification. He has been a member of the IEEE CAS MSA TC and VSPC TC, since 2020 (VSPC Publicity Sub-Committee Co-Chair, from 2020 to 2022). He was the General Co-Chair of IEEE MMSP 2017 and IEEE ICME 2020. He has been serving as the Area Chair for IEEE ICIP, ICME, ISCAS, and VCIP, since 2018, and the TPC Chair for the MCC Symposium at ICNC 2013 and 2014. He was the Co-Chair of the IEEE ComSoc MMTC Interest Group on 3D Processing and Communications, from 2017 to 2019. He organized a Special Issue as the Lead Guest Editor for IEEE JOURNAL OF SELECTED TOPICS IN SIGNAL PROCESSING, in February 2015, and two Special Sessions at 3DTV-Con 2015 and IEEE ICIP 2011. He was awarded a Certificate of Merit for the role of the General Co-Chair of IEEE ICME 2020 and the Outstanding Area Chair of IEEE ICME 2019. He is an Associate Editor of IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, since January 2021, IEEE TRANSACTIONS ON MULTIMEDIA, since August 2021, and *Signal Processing: Image Communication* (Elsevier), since 2015.

VLADIMIR DYO received the M.Sc. and Ph.D. degrees from University College London, London, U.K., in 2003 and 2009, respectively. He is currently a Senior Lecturer of computer networking with the School of Computer Science and Technology, University of Bedfordshire, Luton, U.K. He has been involved in a number of research and industry projects on environmental monitoring, intelligent transportation, and IoT system design. His research interests include the IoT, energy harvesting, and applied machine learning.

FREDI NONYELU received the M.Sc. degree in innovative product design engineering from Cranfield University, in 1988, and the degree from the Cranfield University Business School, in 1989. He is currently the Founder and the Chief Executive of Briteyellow Ltd. Briteyellow's mission is to transform the way people experience indoor spaces through ultra-precision positioning sensors and mixed-reality applications. He is a serial technology entrepreneur with more than 20 years of experience in technology innovation. In the past decade, he was the Technical Lead for Innovate U.K. collaborative research and development projects in high-performance 3D cloud computing, remote monitoring for vulnerable people in need, and 5G connected passenger guidance systems. He is the former Chair of Milton Keynes Science and Innovation Steering Group, Biztech Technology Forum, a peer-to-peer group of technology leaders across the South East Midlands.