Pose Estimation of Spacecraft Using Dual Transformers and Efficient Bayesian Hyperparameter Optimization

Dr. N. Kannaiya Raja[1], Janjhyam Venkata Naga Ramesh[2], Prof. Ts. Dr. Yousef A.Baker El-Ebiary[3], Elangovan Muniyandy[4], Dr. N. Konda Reddy[5], Dr. Vanipenta Ravi Kumar[6], Dr Prasad Devarasetty[7]
Sr. Associate Professor, School of Computing Science and Engineering,
VIT Bhopal University, Bhopal, Madhya Pradesh-466114, India[1]
Adjunct Professor, Department of CSE, Graphic Era Hill University, Dehradun,248002, India[2]
Adjunct Professor, Department of CSE, Graphic Era Deemed To Be University , Dehradun, 248002, Uttarakhand, India[2]
Faculty of Informatics and Computing, UniSZA University, Malaysia[3]
Department of Biosciences-Saveetha School of Engineering,
Saveetha Institute of Medical and Technical Sciences, Chennai - 602 105, India[4]
Applied Science Research Center, Applied Science Private University, Amman, Jordan[4]
Associate Professor, Department of Engineering Mathematics, K L University, Greenfields,
Vaddeswaram, Guntur Dist, Andhra Pradesh-522302, India[5]
Assistant professor, Department of Mathematics, Annamacharya University, New Boyanapalli, Rajampet, India[6]
Department of Computer Science and Engineering, DVR & Dr HS MIC College of Technology,
Kanchikacherla, Andhra Pradesh, India[7]

Abstract*—Spacecraft pose estimation is an essential contribution to facilitating central space mission activities like autonomous navigation, rendezvous, docking, and on-orbit servicing. Nonetheless, methods like Convolutional Neural Networks (CNNs), Simultaneous Localization and Mapping (SLAM), and Particle Filtering suffer significant drawbacks when implemented in space. Such techniques tend to have high computational complexity, low domain generalization capacity for varied or unknown conditions (domain generalization problem), and accuracy loss with noise from the space environment causes such as fluctuating lighting, sensor limitations, and background interference. In order to overcome these challenges, this study suggests a new solution through the combination of a Dual-Channel Transformer Network with Bayesian Optimization methods. The innovation is at the center with the utilization of EfficientNet, augmented with squeeze-and-excitation attention modules, to extract feature-rich representations without sacrificing computational efficiency. The dual-channel architecture dissects satellite pose estimation into two dedicated streams—translational data prediction and orientation estimation via quaternion-based activation functions for rotational precision. Activation maps are transformed into transformer-compatible sequences via 1×1 convolutions, allowing successful learning in the transformer's encoder-decoder system. To maximize model performance, Bayesian Optimization with Gaussian Process Regression and the Upper Confidence Bound (UCB) acquisition function makes the optimal hyperparameter selection with fewer queries, conserving time and resources. This entire framework, used here in Python and verified with the SLAB Satellite Pose Estimation Challenge dataset, had an outstanding Mean IOU of 0.9610, reflecting higher accuracy compared to standard models. In total, this research sets a new standard for spacecraft pose estimation, by marrying the versatility of deep learning with probabilistic optimization to underpin the future generation of intelligent, autonomous space systems.

Keywords*—Dual-channel transformer model; Bayesian optimization; EfficientNet; pose estimation; SLAB dataset

I. Introduction

Spacecraft pose estimation is a critical and very important face of space missions or any spacecraft operation that focuses on establishing the pose of a spacecraft in line with the predefined frame of reference, often earth or another celestial body [1]. The validity of this pose estimation is critical and has some importance in satellite docking, formation flying, planetary landing, and navigation. It utilizes the cameras, star trackers, and inertial measurement units as the sources of data that through the adopted algorithms are used in estimating the pose of a spacecraft. Another noticeable issue of spacecraft pose estimation is the fact that space environment may impact the operation of the sensors and, thus, add errors [2]. Also, the requirement to perform real-time processing. currently a number of ambitious missions have been planned and initiated in near future more and more demand of autonomy is being felt during space operations thus there is a pressing need to revolutionize the spacecraft pose estimation and make it more efficient reliable and accurate for proper execution of mission and to reduce operational risks involved while exploring space [3]. Spacecraft pose estimation is a process that comes with several difficulties, which arise from the fact that space environment is demanding and highly uncongenial for any equipment, which means that any existing equipment is likely to be less accurate or reliable in the space environment as it is in the earth's environment. Thus, the first significant issue is a lack of extensive and high-quality visual information [4]. Lack of adequate illumination at night or in outer space scenarios that involve faint light may affect the functionalities of the sensors such as cameras because the contrast of prominent features

becomes blurred. The high level of contrast in regions which are well illuminated and the rest which are in shade also poses a great problem in feature detection and thus poses estimation., micrometeoroids, cosmic rays, and orbiting space debris may interfere with an instrument's ability to acquire an accurate reading, blur the sensors, or inject noise right into the information collected, resulting in unwanted disturbance and noise when estimating the configuration. Another main issue is the necessity of developing computational efficient and real-time pose estimation algorithms [5]. These are systems in which computational power is generally low, and thus strict limitations are imposed on what kind of algorithms can be executed. Accurate determination of the pose is generally a complex implementation which often demands very complex mechanisms such as those that employ machine learning or superior filtration mechanisms which may be slightly complex. The problems are compounded by the requirement of extremely fast processing as any delay in pose estimation can lead to wrong navigation or mission failure. Moreover, spacecraft usually function in such conditions which are far from being static and often are characterized by rapidly changing positions and speeds of celestial bodies and other space vehicles. This makes the dynamic nature of the pose estimation a constant process hence making the algorithm to be complex. In certain situation such as proximity operations or docking the relative motion may be high and random and therefore the pose estimate needs to be very accurate and robust to avoid any collisions. Lastly, absence of ground data in space environment for testing of pose estimation algorithms also increases the challenge when it comes to developing and testing of these key systems [6].

A. Modern Solutions for Spacecraft Pose Estimation

Many approaches have been used in the past to estimate the pose of spacecraft, which include, CNNs and RNNs, SLAM algorithms, Multi-Sensor Fusion, and Particle Filtering, however, the following challenges hinder the use of these methods in space. Different CNNs and RNNs, deep learning models, the latter need to extend large labeled datasets for learning that are hard to come by in space which is diverse and unpredictable. These models also do not generalize well to new situations or new inputs to the sensors that the car may encounter in practice and were not trained on. The most prominent and reliable algorithms of SLAM, when dealing with mapping and localization perform quite effectively, but can be grasping for computational resources and deteriorate performance in conditions where feature density is low or when the environment frequently changes. The main disadvantage of multi-Sensor Fusion is that it is severely affected by the quality of data received from each of the sensors of a system and the calibration parameters of the sensors, which might introduce some issues in the final results. Particle Filtering, while being more appropriate for the non-linear and/or non-Gaussian cases, can be time consuming and may have problems such as the particle depletion, where the algorithm starts to produce wrong estimates because of the lack of variety in the particle set. Such limitations mean that for reliable and accurate computation of the spacecraft pose, higher level and sophisticated techniques must be employed.

To address the limitations of outdated spacecraft, pose estimation methods, advanced techniques like Transformer Networks and Efficient Bayesian Optimization are being explored. In order to imaginatively integrate transformer to the entire learning satellite pose estimation task, a dual-channel transformers non-cooperative spatial object pose estimating networks is constructed. The satellites' orientation & geographical translation data are effectively separated by the dual-channel network architecture. To numerous different uses, optimization is being used successfully to tune machine learning parameters. When assessments are costly, as in the case of pose estimation, Bayesian optimization is also a useful strategy. Evaluation of optimization algorithms has demonstrated the latest developments in Bayesian optimization. When compared to other nongradient techniques like particle filters and evolutionary algorithms, Bayesian Optimization uses qualified guesses rather than spontaneous mutations and sampling, which reduces the number of iterations needed. Networks and Efficient Bayesian Optimization present a promising direction for developing more adaptive, precise, and robust spacecraft pose estimation systems, enhancing the safety and success of space missions.

B. Key Contribution

Key Contributions are as follows:

- The research proposes a new Dual-Channel Transformer Model that utilizes EfficientNet for feature extraction to enhance flexibility and avoid overfitting as compared to traditional methods.
- A new approach involving 1×1 convolutions is introduced to turn the activation maps into inputs that are suitable with transformers for seamless integration convolutional features with the transformer architecture.
- The model uses two specialized subnetworks: a translation estimation subnetwork and an orientation estimation subnetwork, making use of quaternion-based activation functions to enhance pose prediction accuracy and robustness.
- Bayesian optimization using an UCB acquisition function and a Gaussian process is used in the research to optimize model parameters economically with a minimum number of evaluations.
- The method is verified with the Space Rendezvous Laboratory (SLAB) Kaggle dataset, exhibiting better pose estimation accuracy and optimization performance than existing methods, setting a new benchmark for pose estimation of spacecraft in challenging satellite imagery.

The research paper is structured to provide a clear and logical flow. It begins with an Introduction in Section I, followed by a review of existing methods in Section II. Section III defines the Problem Statement, leading into Section IV, which details the Proposed Dual-Channel Transformer Model and Bayesian Optimization Techniques. Section V presents the Results and Discussion, and the paper concludes with Section VI.

II. Related Works

Accurate and reliable 6D pose estimate is necessary for on-orbit proximity activities like as debris collection, the docking process, and space rendezvous in a variety of illumination

scenarios as well as against a highly detailed history, such as the Earth. Proença and Gao [7] explores the use of photorealistic graphics and deep learning for monocular pose estimation for previously identified uncooperative spacecraft. First, describe URSO, an Unreal Engine 4 simulator that creates tagged pictures of Earth-orbiting spacecraft that can be utilized for neural network training and evaluation. Second, suggest modelling orientation uncertainty as an amalgamation of Gaussians using a deep learning model for posture prediction centered around orientations soft categorization. The ESA pose estimate problem and URSO datasets were used to assess this methodology. Our top model placed second on the real test set and third on the simulated test set during the competition. Additionally, our findings highlight the significance of several architectural and training elements and provide a qualitative example of how models trained on URSO databases might function on real-world images. Subsequent research endeavors ought to contemplate methods such as reducing the final layer connections to substitute dense connections that compromise efficiency. Furthermore, a specific network was used to generate outcomes for each dataset within this work. Having a similar backbone could be advantageous for both effectiveness and performance.

A novel deep neural network process that uses the temporal information during the rendezvous scenario to calculate a spacecraft's related posture. It makes use of LSTM components' capability to model data sequences and handle characteristics that are retrieved by a CNN backbone. Regression- Three distinct training methods are combined to produce superior end-to-end posture estimation and feature-based learning procedures that adhere to a coarse-to-fine funnelled strategy. By combining infrared thermal data alongside red-green-blue (RGB) inputs, CNNs' capacity to automatically extract feature representations from images is utilized to reduce the impact of artifacts during visible-wavelength space object imaging. The suggested framework called ChiNet has been verified on data from experiments, and each of its contributions is shown on a synthetic dataset. The strength of the design in non-nominal illuminating situations may be the subject of future research. In relation to spacecraft pose estimations, a different possible line of inquiry would be to address the issue of domain modification. This involves training a deep network via synthetic images and testing it on genuine information, the latter of which are usually hard to come by before the mission begins, however the earlier kind might be produced in huge quantities [8].

It has been suggested that the ability to estimate the pose of problematic objects in space is a crucial component for facilitating safe close-proximity activities, including active debris clearance, in-orbit maintenance, and space rendezvous. Conventional methods for pose estimation use Deep Learning (DL) algorithms or traditional computerized vision-based approaches. In this article, a unique DL-based approach for predicting the posture of recalcitrant spacecrafts using Convolutional Neural Networks (CNNs) is explored. Unlike other methods, the suggested CNN regresses poses directly, obviating the necessity for any pre-existing 3D information. Furthermore, the spacecraft's bounding boxes using the picture are anticipated in an easy-to-understand but effective way. The tests conducted show if this work interacts with the state-of-the-art in uncooperative spacecraft position estimation, which includes work needing 3D input and work that uses complicated CNNs to anticipate boundaries. [9].

For numerous space missions, including formations flying, rendezvous, the docking process, repair, and debris from space cleanup, spacecraft posture estimation is crucial. This Approach provide based on learning that uses uncertainty predictions to determine a spacecraft's attitude from a monocular picture. Firstly, cropped out the rectangle portion of the original image wherein only spacecraft were visible using a SDN. Subsequently, 11 pre-selected important points having clear features within the clipped image were detected and ambiguity was predicted using a keypoint detection network (KDN). To autonomously choose keypoints that possess greater detection precision from all identified keypoints, that provide a key location selecting approach. Using the EPnP technique, the spacecraft's 6D posture was estimated using these chosen keypoints. Research utilized the SPEED dataset to assess our methodology. Our approach works better than heatmap- and regression-based approaches, according to the studies, and the efficient uncertain predictions can raise the pose estimation's ultimate precision [10].

A real-time spaceship pose estimate technique by fusing the least-squares approach with a model using deep learning. With automated rendezvous docking and inter-spacecraft interaction, pose estimation in orbit is essential. Since deep learning algorithms are challenging to train in space, Research demonstrated that real-world trial outcomes may be predicted by software simulations conducted on Earth. This paper used a combination of DL and NLS to accurately estimate the pose in actual time given a single spacecraft photograph. To train a deep learning model, researchers built a virtual environment that can generate synthetic images in large quantities. The research presented here suggested a technique for using just synthetic photos for developing a DL model, a real-time estimating method with a visual basis that may be used in a flight testbed was built. As a consequence, it was confirmed that software models with the identical surroundings and relative distance could accurately anticipate the hardware outcomes of experiments. This work demonstrated an adequate application of a deep learning model learned solely on artificially generated images to actual images. Therefore, our study shod that the approach developed using just artificial information was suitable in space and provided a real-time pose estimate program for autonomous docking [11].

There have been a lot of recent study on the use of deep learning algorithms for space applications. Spacecraft posture estimate is a particular field where these algorithms are becoming more and more popular. This is because it is a basic need in numerous spacecraft navigation and rendezvous procedures. However, compared to terrestrial operations, the utilization of similar algorithms in space operations presents distinct obstacles. In the latter case, servers, powerful PCs, and shared assets like cloud services enable them. These resources are constrained in the space environment and ship, though. Therefore, an efficient and low-cost on-board predicting is needed to benefit from the above methods. Deep learning techniques for use in space were the subject of extensive research in the recent past. One arena wherein these methods are

gaining traction is spacecraft posture estimate, which is essential for many spacecraft rendezvous and navigational procedures. Nonetheless, the utilization of such algorithms in space operations poses unique challenges in contrast to how they are used in terrestrial operations. In the last scenario, servers, powerful PCs, and shared assets like cloud computing enable services to be provided. However, in space conditions and spacecraft, these resources are limited. Thus, in order to take use of these gets closer, an on-board inferencing that is both economical and power-efficient is required [12].

The drawbacks of the works concerning the pose estimation of a spacecraft are identified below. Most solutions leverage synthetic images in training deep learning models and hence may not be so effective when employed in real-world settings due to domain shift. Some methods involve the use of prior 3D information or intricate image texture and thus are limited due to unavailability of the information. Besides, they introduce many parameters in complex networks, thus reducing the real-time inference capacity and the practical applicability. The necessity of having a large number of synthetic images and real images for training can be time consuming with some sort of methods may work poorly under different illumination and high detailed backgrounds like Earth. Additionally, those methods not accounting for this uncertainty in key point detection and pose estimation may result in inferior solutions. Finally, it is very common not to have robust solutions adaptable to a large number of Space Craft arrangement and the operational settings.

III. Problem Statement

Past research carried out to estimate the pose of the spacecraft has benefited significantly from deep learning and computer vision methods, however, it has left rooms for improvements in some areas such as the accuracy and the time efficiency extremely much more especially when undertaking the tests under different illumination conditions and also when dealing with uncooperative spacecraft [13] [14]. Previous approaches have issues with large computational costs, for example, it difficult for them to perform well under different lighting conditions, and pose uncertainties are not well addressed. To overcome these shortcomings, this proposed approach renders the following new approach that combines Transformer networks and Bayesian optimization. Thus, a new framework is proposed here to improve the accuracy and speed of pose estimation by applying the aptitude of Transformer models in dealing with the sequence data and in efficiently introducing the model parameters. This also aims at addressing some limitations posed by previous methods, such as weaker pose estimates, restricted applicability across various situations and until now called for poor real-time performance due to non-efficient and often unscalable solutions.

IV. Proposed Dual-Channel Transformer Model for Spacecraft Pose Estimation and Bayesian Optimization Techniques

The suggested Dual-Channel Transformer Network along with Bayesian Optimization was selected for its better capability to process complicated satellite imagery and its efficiency in learning spatial as well as rotational features. As compared to conventional techniques which are plagued by excessive computational cost, poor generalization, and vulnerability to noisy or low-contrast data, our approach is better in terms of adaptability, real-time operation, and accuracy. This renders it extremely suitable for spacecraft pose estimation in harsh space environments. This choice is also substantiated by the limitations of current approaches such as CNNs, SLAM, and Particle Filters tend to have high computational requirements, poor generalization, and are very sensitive to noise or low-contrast images. These constraints limit their performance in dynamic or uncertain space environments, and they are less reliable for accurate and robust spacecraft pose estimation. The research process is guided by a systematic workflow to provide an efficient and accurate spacecraft pose estimation model. As shown in Fig. 1, the process starts with Data Collection, where raw images are acquired from sources like TRON authentic photos and synthetic datasets. The Creation of the Synthetic Dataset is essential in complementing real-world images and improving model training. The second step is Data Pre-Processing, wherein gathered images and synthetic images are refined to be rid of noise and to provide a uniform format for input. The processed data is then passed on to the Dual-Channel Transformer Model, utilizing EfficientNet for superior feature extraction. To further enhance model precision, Bayesian Optimization is utilized to optimize parameters, minimizing errors made through estimation and enhancing generalization. This end-to-end workflow increases the stability and efficiency of the model, allowing it to process intricate satellite images efficiently and perform sophisticated data analysis operations with great accuracy.

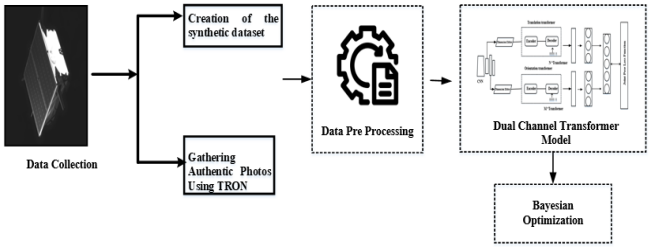

Fig. 1. Proposed figure.

A. Data Collection

Research makes use of the dataset that Space Rendezvous Laboratory (SLAB) made available on Kaggle for their Satellite Pose Estimation Challenge. The training dataset comprises 12,000 artificial satellite images together with matching ground truth pose labels. There are 300 genuine photos and 2998 artificial images in the test dataset. The purpose of the genuine photographs, which differ significantly from the artificial ones, is to assess how well the posture estimation model and algorithm work with a real-world dataset. They were taken at SLAB using a Tango satellite mockup. Distribution of Synthetic and Real Images in Training and Test Sets is given in Table I.

TABLE I. Distribution of Synthetic and Real Images in Training and Test Sets

Dataset	Synthetic	Real
Training set	12000	5
Test set	2998	300

Every image offered has a 1920 × 1080-pixel resolution and is 8 bit monochrome. Using a high-definition texturing modeling of the Tango spacecraft from the PRISMA mission and a camera model of the Point Grey Grasshopper 3 camera with a Xenoplan 1.9/17mm lens (VBS), SLAB's Optical Simulator creates the synthetic photos. To simulate camera noise and depth of field, accordingly, Gaussian blurring and white noise are applied to every image. Some of the photos simulate scenarios in which the subject is photographed against a star field by having a black background. Real photographs of the Earth either completely or in part cover the background of the remaining pictures. The subsequent set of test images consists of real images that are sourced from SLAB's TRON facility. Utilizing a real Point Grey Grasshopper 3 camera equipped with a Xenoplan 1.9/17mm lens, TRON delivers photographs of a 1:1 mockup model for the Tango spacecraft of the PRISMA mission. Keep in consideration that the OS webcam emulators program uses the exact same camera. The locations and postures of the Tango spacecraft and the camera have been captured by calibrated motion-capturing cameras, and these data are utilized to determine the Tango satellite's ground truth pose in relation to the camera. We assess every algorithm's transferability between synthetic to real images using a test set of real images [15].

1) Creation of the synthetic dataset: The Optical Stimulator's camera emulator programs are used to produce the artificial visuals on the Tango spacecraft. The software creates photo-realistic pictures of the Tango spacecraft with the necessary ground-truth postures using an OpenGL-based image rendering process. 50% of the synthetic photos have random Earth photographs from the Himawari-8 geostationary meteorological satellite4 placed into the background of the image. The artificial light used for these photos is designed to most closely resemble the background of Earth visuals. The intersecting histogram curves of the image pixel intensities from both imageries show that the synthesized imagery produced by SPEED may nearly mimic the lighting levels recorded by the real flight photography. This shows off how much SPEED's image processing process has improved and how it can produce realistic, pose-labeled photos for any chosen spacecraft. [16]

2) Gathering authentic photos using TRON: Gathering authentic photos using TRON, the Tango spacecraft's actual photos were taken with SLAB's TRON facilities. Upon creating the images, the setup comprised a one-to-one replica of the Tango spacecraft along with a robotic arm with seven degrees of freedom fixed to the ceilings that supported the camera at its tip. A xenon short-arc lamp that simulates convergent sunlight in various orbital regimes and special LED wall panels that might simulate the dispersed lighting conditions brought on by Earth albedo are also features of the center. To get the ground-truth posture labels in the real photographs, ten Vicon cameras are employed to monitor the infrared markers between the evaluation camera and the space station replica. To eliminate any errors in the predicted targets and camera references frames, the meticulous calibration procedures described are carried out. In general, the calibrated Vicon system's autonomous posture assessment yields pose labels that have degree- and centimeter-level accuracy. The present efforts are being made to simultaneously combine readings from the robot and Vicon cameras to increase the ground-truth pose's accuracy by a few orders of magnitude. It should be noted that despite the fact which the two photographs have the same ground-truth positions and the Earth's albedo in overall, there are plenty of differences in the image characteristics which may be easily noticeable, including the texture, illumination, and eclipses of particular spacecraft elements [17].

B. Data Preprocessing Steps

1) Image loading and conversion: The initial step in the data preprocessing pipeline involves the careful loading and conversion of the 8-bit monochrome images, which form the core of the dataset. These images, both synthetic and real, are stored in a format where each pixel's intensity is represented by a value ranging from 0 to 255, a typical range for 8-bit images. To begin, the images are loaded from the dataset using image processing libraries, ensuring that they are accurately read and stored in memory for further manipulation. Once loaded, the images are converted into a format that is more suitable for processing, such as NumPy arrays, which provide an efficient and flexible structure for handling large datasets in machine learning workflows. This conversion is essential as it allows for the application of various mathematical operations and transformations required during the preprocessing phase.

Among the steps of data pre-processing, the first and rather time-consuming one is loading and converting the 8-bit monochrome images on which the set is based. These images could be synthetic as well as real and are in a format in which the value of each pixel in terms of intensity can be in a scale of 0 - 255, which is a commonly employed format in ‘’8-bit’’ images. The first process in this case is the reading of the images from the dataset using image processing libraries, and the images are first preprocessed in that the images are brought into memory for further processing. After loading then they convert the images into a format that is easier to process, one of them being the NumPy arrays, which enhances the capability of large dataset input for use in most machine learning algorithms. This conversion is important because many different operations and transformations that are required at the preprocessing step are only possible with numerical data. To standardize invariant input, pixel intensity is scaled in all the images in the dataset. This is among other things done in an effort to standardize the pixel intensity values from their initial range of 0 to 255 to a range of 0 to 1. Normalization is a very crucial step for such reasons because it assists in bringing the pixel intensity into the similar ranges and in this way, not a single pixel intensity value will be overly influential during a training of the model. This way the input data is preprocessed in a way that is easier for the model to learn from the images hence improving performance and generalization that will be exhibited when the model is ran on another data set [18].

2) Gaussian blurring and noise addition: Gaussian blurring and noise addition are two major operations intending to improve the quality of synthetic images in the way that the synthetic imagery has faults that real data does not. The

Gaussian blurring is done by convolving each image with a Gaussian kernel and the standard deviation was set to 1 to smoothen the image but discard as well much of the high frequency noise. This blurring technique is fully realistic because it mimics factors such as depth of field and smoothed out of focus blurring that may be found inside actual camera systems to take off harsh edges and give synthetically produced images a more natural look. Further, to mimic the noise patterns of actually camera sensors, Gaussian white noise for enhancing the electrode signal to noise ratio is incorporated to the images. This noise that has a zero mean and a variance (σ^2) equal to 0 is defined as follows: 0022, adds additional small variations in pixel intensity that look like the phenomenon of shot noise, the kind of noise that arises from the nature of light. These adjustments are then used subtly to recreate a form of realism making the manufactured synthetic images to mimic natural response of real-world images which in effect enhances the model performances while on the testing phase [19].

3) Background segmentation: It is also necessary to define and divide the background of the images which can be background (black or Earth background) this operation is very important in order to separate the satellite from the background, which results in pose estimation improvement. Specifically for images with the Earth background, one needs to consider that the segmentation algorithm should be able to work with variation of the Earth's appearance and illumination

4) Resize and cropping: Before feeding them to the model, down sample the pictures to a more standardized size if that is required to minimize computational strain on the algorithm. When resizing make sure that the aspect ratio of the satellite is retained to avoid stretching of the satellite. Trim the pictures to the satellite, erasing everything else that might be around them or surrounding the satellite.

5) Histogram matching: This is done in order to align the intensity features of the two images and minimize the variations of illumination and contrast. This step normalises intensity distribution of synthetic images to that of real images, which is helpful when one uses transfer learning. This is particularly important since the histograms of curves of the synthetic images and the real images are nearly the same implying that they were both under the same illumination conditions. Fig. 1 Pre Processing steps are described in Fig. 2.

C. Dual-Channel Transformer Model

The batch size has been set as B provided the satellite picture M∈R^(C×H×W). Following the EfficientNet extraction of features network, 2 layer of features P (t) and P (r), that have various rate sizes are chosen at random and allocated to each of the regressive sub networks. For converting activation maps into input that are suitable with transformers, must first convert is converted into P∈R^ (B×C×H×W) to (P)¨∈R^ (B×X×Y) accordingly, using 1×1 convolution in a dimension editor that follows its processing rules. The transformer's process stream is comprised of an encoding and a device for decoding.

Processed P∈R^(B×C×H×W) to (P)¨∈R^(B×X×Y)in order to translate activation maps into transformer-compatible input. The activating maps are flattened by the dimension editors using 1×1 convolution in accordance with their processing rules; P∈R^(B×C×H×W)is processed into P∈ R^(B×X_R×Y_R)accordingly. The encoder and decoder that make up a transformer's working flow is given in Eq. (1)

$$Z' = Decoder(Encoder(Z^{1-1}) \quad (1)$$

where Z^1 is the result of processing with numerous transformers, where Z^1 is handled as a one-dimensional sequenced feature S via the flattening layer after being produced by multiple-transformer analysis. In order to generate the pose information, next input S into the completely linked layer. The function that activates in the oriented regress networks is quaternion SoftMax-like. Dual-channel transformer model is shown in Fig. 3.

D. EfficientNet Backbone Network

The new architecture of EfficientNet originated from the MBConv that integrated the SE system's attention mechanism. The SE module that was originally placed after deep convolutional layers describes how to refine feature responses using point-wise convolutional for the improvement of features., MBConv incorporates this idea at an earlier level where point-wise convolutions are applied to transform the dimensions of features before going deep convolution; thus, improving feature extraction with reduced computation. EfficientNet is therefore computationally efficient in feature extraction and high in performance from images. This is done sequentially, meaning that the model is trained progressively in terms of depth, width and resolution not exceeding a level that demands more computations which would slow down the system. The model's architecture also helps in getting faster training sessions owing to the lower computational demands of this network as opposed to other feature extraction networks. This efficiency is vital when working with large datasets of image features, for example, satellite images in which both the quality and the speed of the recognition are to be achieved [20]. Efficient Net Architecture is shown in Fig. 4.

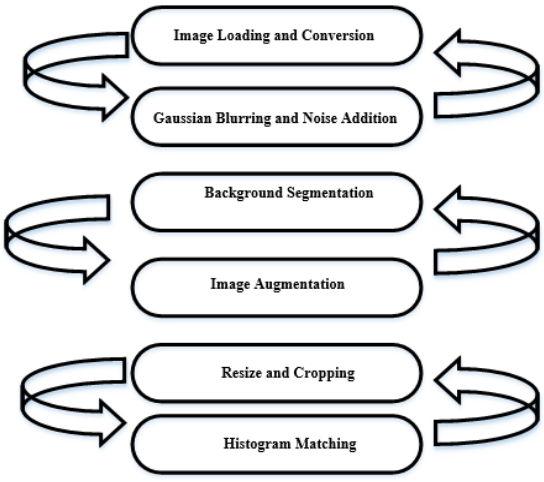

Fig. 2. Pre-processing steps.

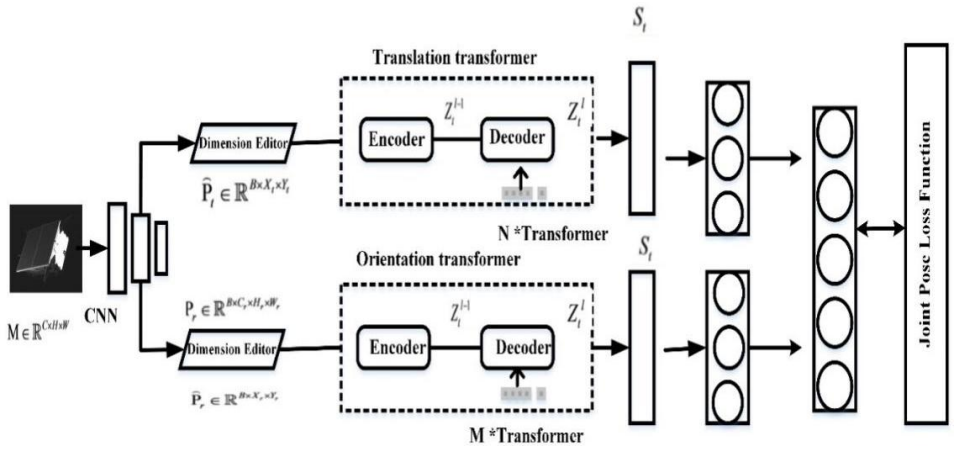

Fig. 3. Dual-channel transformer model.

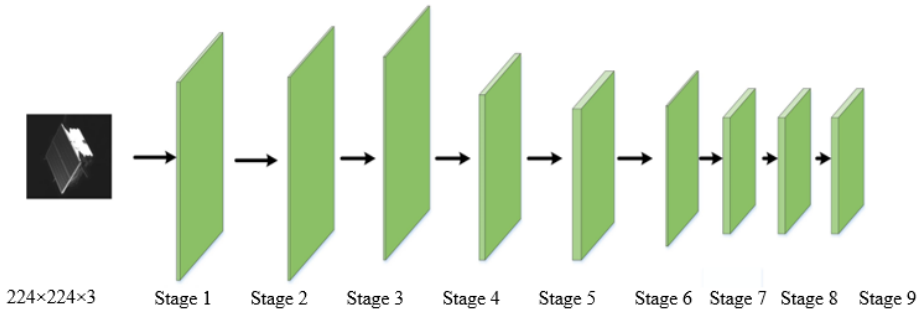

Fig. 4. Efficient net architecture.

Initially, having 32 convolutional layers of 3 × 3 × 3 with an initial phase size of 2 × 2, a feature map containing an input size of 224 × 224 × 3 is processed to yield 112 × 112 × 32 following normalizing and Swish function activation analysis. Following the initial processing, the features go through 16 distinct MBConv layers before being ultimately sized at 7 × 7 × 1280 Two feature layers are selected at random & fed into the translation transformers and the perspective transformers, two estimation of poses subnetworks, in the dual-channel transformers design.

1) Feature layers and dimension editing: The Feature Layers and Dimension Editing, thus, introduce the features extracted by the EfficientNet model into deformation ready for feeding to the Transformer. In particular, two feature maps named P_t and P_r are chosen from EfficientNet's output laying base for the next stage. These layers indicate different abstraction level in feature hierarchy, which means they have different sizes, and thus represent different resolution of the input data and are rich sets of inputs from which it is possible to extract features. They have to be transformed to be implemented within.

2) Transformer model architecture: A transformer is made up of multiple network blocks and an encoding and decoding unit. Positional encoding (PE), self-attention (SA), feed-forward network (FFN), residue relationship, normalization of layers (LN) blocks (Add & Norm), and multi-headed attention (MHA) are among its components. SA is the fundamental block of MHA. Transformers makes use of Add and Norm for enhanced model fitting, FFN to facilitate modelling learning, and MHA to connect diverse characteristics. Schematic diagram of the transformer structure is shown in Fig. 5.

3) PE: Preserving the spatial location data among each of the input image blocks is the primary goal of positional encoding. The features' positional is encoding is given in Eq. (2).

$$\begin{cases} PE_{(pos,2i)}=\sin(pos/10000^{2i/d}) \\ PE_{(pos,2i+1)}=\cos(pos/10000^{2i/d}) \end{cases} \quad (2)$$

In the given scenario, wherein PE is a matrix with two dimensions, the parameters sin and cos are positioned in its both even and odd terms, respectively. The a two-dimensional matrices is formed from variables such as sin and cos, and z^(l-1)has been encoded in positional.

4) SA: A key element of transformer is self-attention. By directing focus via a mathematical method, it replicates the properties that biological observing targets and collects features of particular important locations. The self-attention mechanism offers benefits in parallel processing, enhanced localized attention, and distant learning. The self-attention technique is primarily accomplished utilizing scaling dot-product focus, is given in Eq. (3).

$$Attention(Q,K,V) = softmax(\frac{QK^T}{\sqrt{d}})\,V, \quad (3)$$

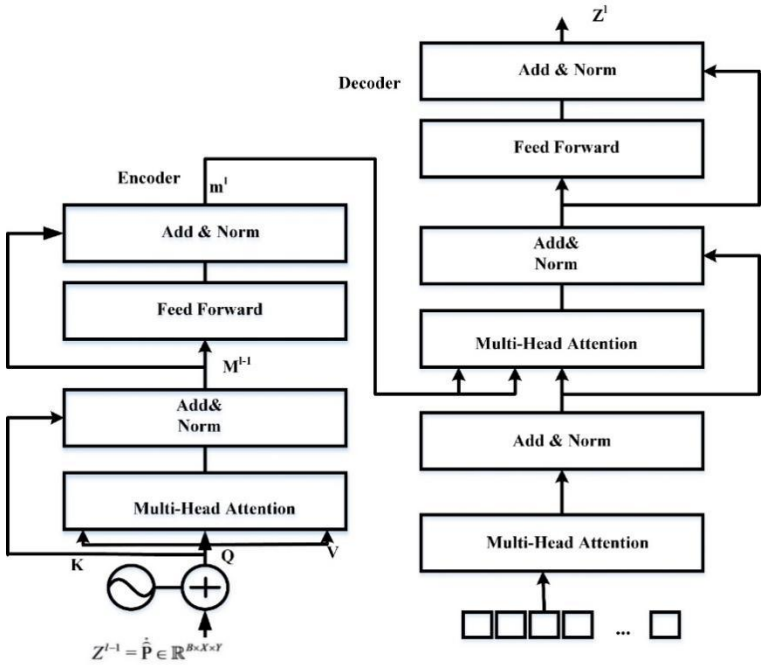

Fig. 5. Schematic diagram of the transformer structure.

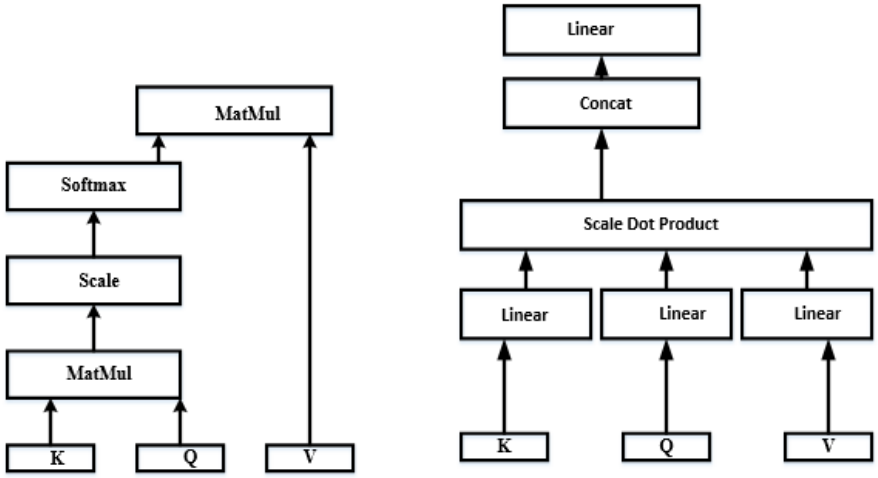

Fig. 6. Structure of a) SA module, b) MHA module.

where Q, K, and V represent the query matrix, key matrix, and value matrix, respectively, and d represents the input feature's dimensions. These are created by multiplying the matrix with the feature by 3 randomised weighting matrices. Structure of SA module and MHA module is shown in Fig. 6.

5) MHA: MHA is employed in a variety of projected areas to determine various projection information. The input matrix is then projected in various directions, and the resultant matrix is pieced together. SA is executed concurrently by MHA for every forecast outcome this is given in Eq. (4)

$$head_i = \text{Attention}\,(QW_i^Q, KW_i^Q, VW_i^V) \quad (4)$$

With $d_k = d_v = \frac{d_{model}}{h}$, denotes the total amount of heads are arranged, and d_{model} denotes the total length of the given input feature. $d_k = d_v = d_{\frac{model}{h}}\ and$ indicates no of heads.

Concatenated the projected computation outcomes of numerous heads, is given in Eq. (5).

$$\text{MHA (Q, K, V)} = Concat\ (head_1, head_2, \ldots\ldots\ldots head_h)W^0 \quad (5)$$

Where in W^0∈R^(⟦hd⟧ _(v×d_model)). Multi-head technology allows for the more detailed extraction of distinct heads' attributes. The feature extraction impact is better whenever the overall computation volume is equal to the value of a single head.

6) FFN: FFN maps features after a mapping from the high-dimensional space to the low-dimensional space. Incorporating various forms of information, improving the model's ability to solve problems, and removing low-resolution features by lowering the dimensionality are the objectives of mapping features to high-dimensional spaces. The method is derived in Eq. (6).

$$FFN(x)=\max(0,W_1 x+b_1)\,W_2+b_2 \quad (6)$$

where $W_1 \in R^{d_{model}}$and are the learnable weights, and b1 ∈ R4dmodel and b2 ∈ R4dmodel are the learnable biases.

Add & Norm: LN blocks and residual connections are contained in Add & Norm. The network depth's processing capability can be enhanced by the residual relationship, which can also successfully stop gradient expansion and the disappearance. LN accelerates the point of convergence of the mathematical framework by stabilizing the data feature distributions this is given in Eq. (7) and Eq. (8)

$$F(X) = LN(m^1 + m^{l-1}) \quad (7)$$

$$LN(x_i) = \alpha \times \frac{x_i - E(X)}{\sqrt{Var(x)+\epsilon}} + \beta \quad (8)$$

where α and β are the parameters that can be learned, and if their variance is zero, ε is used to avoid mistakes in calculation.

In the pose estimation subnetworks of the dual-channel Transformer model, two distinct regression subnetworks are employed to derive comprehensive pose information from feature maps. The first subnetwork is dedicated to estimating translation, or the position of objects, by analyzing spatial features extracted from the image. This involves regressing feature maps to predict the object's location. The second subnetwork focuses on estimating orientation, which involves predicting the object's rotation. For this task, a quaternion-based activation function is utilized, often resembling a SoftMax function but tailored to handle quaternion representations of rotation, providing a robust way to encode 3D orientations. Following the Transformer's processing, which enhances the feature representations through attention mechanisms and encoding-decoding processes, the resulting multi-dimensional output is flattened into a one-dimensional sequence. This flattened sequence is then fed into fully connected layers, which aggregate the information to produce the final pose estimates, including both translation and orientation of the object within the image. This structured approach allows the model to effectively combine and utilize the spatial and rotational data extracted from the satellite images [20].

7) Bayesian optimization: For the purpose of spacecraft pose estimation, Bayesian Optimization is used to fine-tune model learning rates, dropout rates, and the depth of Transformer layers to decrease pose estimation errors. The process starts with the formulation of objective function which measures the error involved in pose estimation which is the goal of the optimization process. Gaussian Process (GP) is employed to map the behavior of the objective function given a limited number of evaluations and yield a probabilistic estimate of the function mean and variance at locations in the design space yet unobserved. The Upper Confidence Bound (UCB) acquisition function then dictates which new set of parameters should be sampled in the next iteration with the intent of balancing exploration, where new parameters with a high level of uncertainty are chosen, and exploitation where parameters with a higher predicted reward is chosen. By indicating this iterative approach, it is possible to quickly navigate the parameter space and adaptively fine-tune the model's accuracy in terms of estimating a spacecraft's attitude and position with as few evaluations as possible to achieve the best result.

Bayesian Optimization uses qualifying guesses rather than spontaneous mutations and sampling, which reduces the number of repetitions needed. Utilizing a surrogate model that is fitted to every one of the prior specimens, the subsequent one to be evaluated is chosen in Bayesian optimization. Given the parameters, random uniform sampling is used to create an amount of starting samples. The surrogate model that is being utilized is a Gaussian process that is a non-parametric approach that builds a framework using all of the prior samples. A prediction's likelihood is also provided by the Gaussian process. As an acquisition function, the widely recognized Upper Confidence Bound (UCB) is utilized. As the name suggests, the subsequent parameter setting that is investigated is chosen based on the confidence bound above its present max. Bayesian Optimization is applied to enhance the pose estimation system by optimizing key parameters, specifically the feature radius and normal radius, which are fundamental to feature matching methods.

Fig. 7 depicts a parameter optimization procedure, where the process is initiated by the selection of 'p' training scenes, every scene comprising of 'm' objects that results in 'm*p' object detections. First, a parameter set is used for recognizing the objects (A) and then the result is assessed by scoring function (B). Afterwards, a Gaussian Process models the distribution of these performance results, which is denoted as C. According to this model the decision-making process chooses new sets of parameter for test (D). This is followed by the creation of the Gaussian Process with a selection of pre-specified number of parameter values and then applying Bayesian Optimization for 'n' steps. Lastly, all the 13 parameters and their assigned scores are employed to fit an extra Gaussian Process in order to determine expected optimum parameter set (E). This last stage supplements the identifications made on the best parameters by utilizing the acquired data to anticipate and determine the most profitable parameter setting.

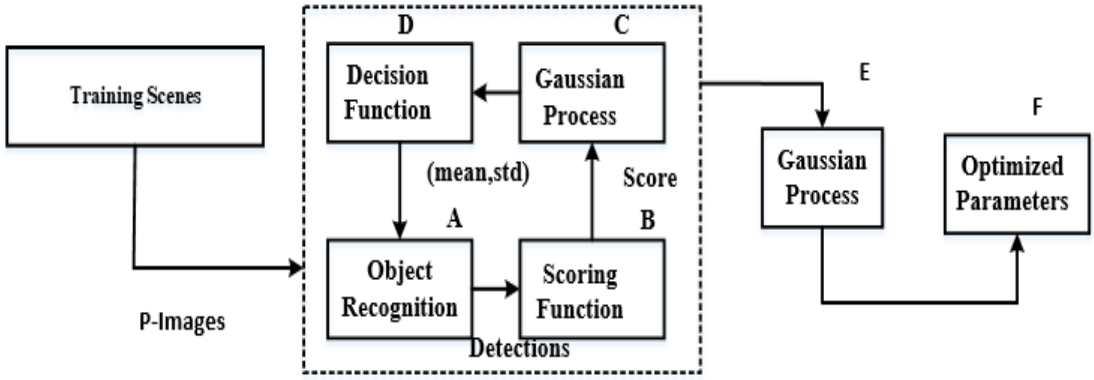

Fig. 7. Bayesian optimization for hyperparameter tuning in object recognition.

8) Scoring the detections: The outcome of the detection method's score to generate an additional score towards the optimization framework in order to maximize efficiency for reliable identifications. The system's overall score into TPs and FPs, or right and wrong findings, to obtain KDET P and KDEFP, accordingly. Any scoring mechanism may theoretically be employed in this situation, however the KDE is the result of the kernel density score during a pose given by the fundamental pose voting technique utilized for the estimation. Research employs the TP/FP ratio to calculate the score, rewarding high scores for accurate findings and penalizing higher scores for incorrect findings. For numerical causes, the score function is log-transformed since it produces more reliable results when the optimizing process is used in Eq. (9)

$$\text{score(KDE)} = \begin{Bmatrix} \log(\sum \frac{KDE_{TP}}{KDE_{FP}}) & i \sum KDE_{TP} \geq \sum KDE_{FP} \\ o, & \end{Bmatrix} \quad (9)$$

9) Gaussian process regression for mode finding: There is a chance of overfitting parameters for just the specific set of training scenes observed over training because just a tiny training set is utilized. This also utilize a Gaussian Process for regressing across the completed group of evaluations in order to reduce the possibility of incorrect parameter selection. This process makes the ideal parameter set prediction more accurate and smooth. To prevent overfitting of sparse training sets by Bayesian optimization techniques, alternative methods were additionally proposed. A term that penalizes steep peaks has been added to the newly acquired function. A Gaussian Process is then fitted to all examined sites in order to identify a stable maximum. The matrix made up of the variables X and the final score y can be used to represent the total amount of investigated points, or n this is given in Eq. (10)

$$x, y = \left\{ \left(x_i f(x_i) \right) \ | i = 1, \ldots \ldots n \right\} \quad (10)$$

A distribution is necessary in order to use a Gaussian Process for predicting the predicted result at new values for parameters. In this case, ˆx represents a brand-new, unproven parameter set this is given in Eq. (11).

$$\begin{bmatrix} y \\ x \end{bmatrix} \sim \begin{bmatrix} K & K_x^T \\ K_x & K_{xx} \end{bmatrix} \quad (11)$$

When K is the covariance matrix provided by a chosen kernel, k(x1, x2), and each index is determined by the interaction of a pair of parameters. Thus $K_{nxn,}$ $K_{nx1,}$) is obtained. to determine the new parameter's predicted value, which is determined by the difference between the variance and the mean.

By using the mean and the range to represent the degree of uncertainty, determine the anticipated amount of the newly added parameter this is given in Eq. (12) and Eq. (13)

$$E(x) = K_x K^{-1} \text{y} \quad (12)$$

$$var(x) = K_{xx-} K_x K^{-1} K_x^T \quad (13)$$

In this case, the kernel function K requires a distance d as inputs, integrating the Matern covariance C_V and the diagonal noise terms N. This adds an additional term into the covariance functioning, which when combined with the Bayesian Optimization yields a Matern-kernel as well as a White Noise kernel, making the Gaussian process less susceptible to noises this is given in Eq. (14).

$$K(d) = C_V(d) + N(d) \quad (14)$$

The function is represented by J, and the gamma function is denoted by Γ. Since the entire dataset is not utilized, the white noise increases the evaluation's uncertainty this is given in Eq.. (15) and Eq. (16).

$$C_V(\text{d}) = \sigma^2 + \frac{2^{l-v}}{\tau(v)} \left(\sqrt{(2v\frac{d}{\rho})^v} j_v((2v\frac{d}{\rho}) \right) \quad (15)$$

$$N(d) = \begin{Bmatrix} \sigma, & if\ d = 0 \\ 0, & otherwise \end{Bmatrix} \quad (16)$$

The equation provided seems to define a kernel function, $N(d)$ where d represents some distance measure, and the kernel takes the value σ sigma when d=0 otherwise. This kernel is used to construct the covariance matrix for a Bayesian optimization process. The parameters must be established prior the prediction may prove computed, even though this kernel is utilized to produce the covariance matrix. is used to do a minimization procedure which fixes the v value, or the amount that distant points interacts with the projected result, whereas fitting the parameter values to the known score y. That will

provide a more accurate parameter forecast. These variables allow for the calculation of the kernels and the creation of an additional durable function given the expected parameter space. Numerous samples have been obtained and the space of parameters is investigated utilizing Bayesian optimization utilizing the training information and the scoring system [21]. Flowchart for Bayesian optimization is shown in Fig. 8.

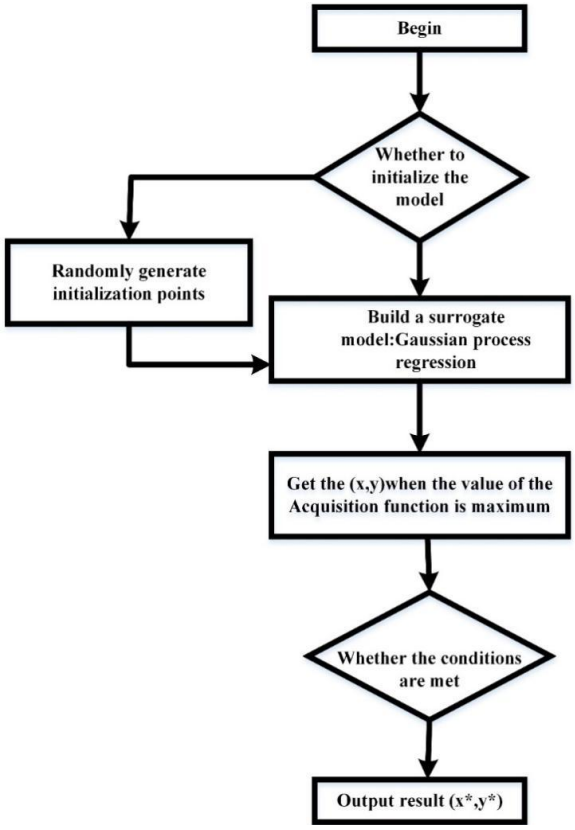

Fig. 8. Flowchart for Bayesian optimization.

```
# Pseudocode for Bayesian Optimization with Gaussian Process Regression
Start: Initialize the problem
Define the objective function f(x) to be optimized
Define the parameter space X (e.g., feature radius, normal radius)
Initialize Gaussian Process with a chosen kernel (e.g., Matern kernel)
Initialize acquisition function (e.g., Upper Confidence Bound - UCB)
n_initial_samples = 10
X_initial = RandomUniformSampling(X, n_initial_samples)
y_initial = EvaluateObjectiveFunction(f, X_initial)
GP = FitGaussianProcess(X_initial, y_initial)
n_iterations = 100
for i in range(n_iterations):
X_next = SelectNextSample(GP, X, acquisition_function="UCB")
y_next = EvaluateObjectiveFunction(f, X_next)
X_initial.append(X_next)
 y_initial.append(y_next)
 GP = FitGaussianProcess(X_initial, y_initial)
Log or print the best result so far
   BestX, BestY = GetBestResult(X_initial, y_initial)
   print(f"Iteration {i+1}: Best X = {BestX}, Best Y = {BestY}")
Output the final optimal parameters and corresponding score
OptimalX, OptimalY = GetBestResult(X_initial, y_initial)
print(f"Optimal Parameters: {OptimalX}, with score: {OptimalY}")
End
```

V. Result and Discussions

This research presents a novel breakthrough in spacecraft pose estimation by combining deep Transformer networks with Bayesian Optimization algorithms. The suggested Dual-Channel Transformer Model, augmented with EfficientNet-derived feature layers, is shown to exhibit higher accuracy in pose estimation than traditional approaches. Through the use of Bayesian Optimization, the model efficiently optimizes essential parameters like learning rates and network depths, making use of Gaussian Process Regression and Upper Confidence Bound (UCB) in order to reduce pose estimation error. The performance of the model is strictly verified using the Space Rendezvous Laboratory (SLAB) dataset, achieving significant improvements in both translational and rotational accuracy. The findings showcase a notable decrease in position and attitude errors under different distances, enhanced reliability in actual spacecraft pose estimation applications, and optimized parameters for the model that increase both efficiency and accuracy. This novel method sets a new standard for spacecraft pose estimation, proving effective in processing complicated satellite imagery and enhancing overall model performance.

A. Performance Evaluation

1) Localization Error (Translational Error): This measures the Euclidean distance among the foretold and ground-truth positions in 3D space (X, Y, Z). The formula for localization error is given in Eq. (17).

$$\sqrt{(x_{pred-}x_{true}{}^{2}) + (y_{pred-}y_{true}{}^{2}) +) + (z_{pred-}z_{true}{}^{2})} \quad (17)$$

Where $x_{pred,}$ $y_{pred,}z_{pred,}$ are the predicted coordinates, and $x_{true,}$ $y_{true,}z_{true,}$ are the ground-truth coordinates.

2) Orientation Error (Rotational Error)**:** This measures the angular difference between the predicted and ground-truth orientations, typically represented by quaternions or Euler angles. The rotational error in degrees can be computed in Eq. (18).

$$Orientation\ Error = \cos -1(2(\ q_{pred}.q_{true})^{2} - 1 \quad (18)$$

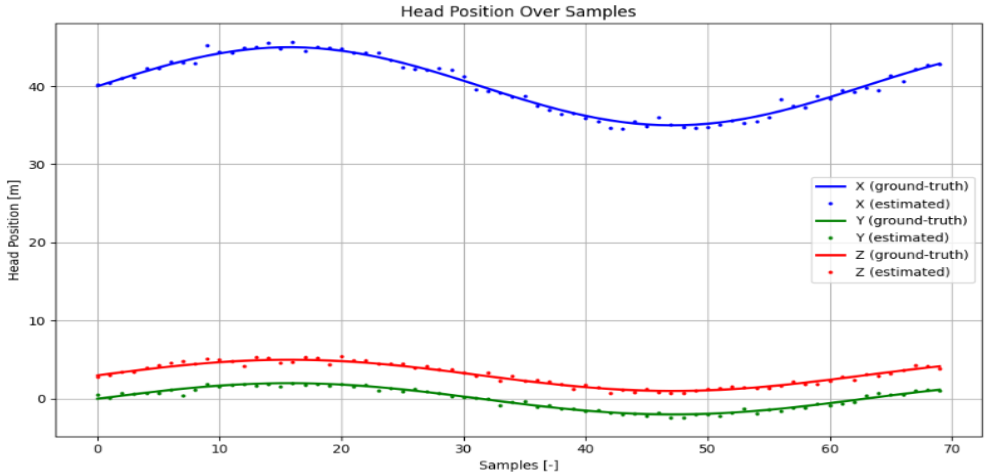

Fig. 9. Head position over samples.

Fig. 9 illustrates the comparison between ground-truth and estimated head positions across X, Y, and Z coordinates over several samples, with solid lines depicting ground-truth and dashed lines showing estimated positions. The blue, green, and red lines correspond to the X, Y, and Z coordinates, respectively. This visualization is key for assessing the accuracy of head position estimation algorithms, particularly in motion tracking and virtual reality applications. A close alignment between the ground-truth and estimated lines suggests that the estimation algorithm performs with high accuracy, effectively mirroring the true head movements across all three dimensions.

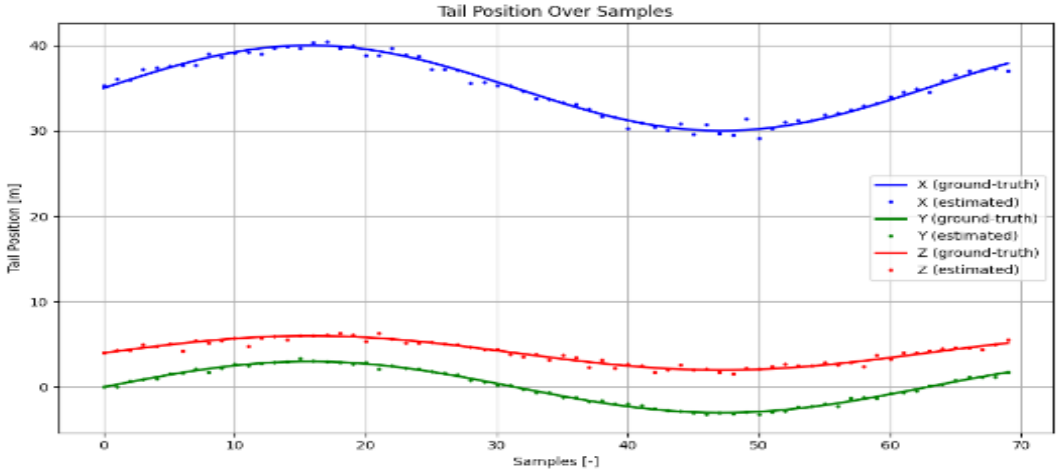

Fig. 10. Tail position over samples.

Fig. 10 shows the comparison between the ground truth and estimated tail positions in three dimensions (X, Y, Z) over a series of samples. The x-axis represents the sample number, ranging from 0 to 70, while the y-axis represents the tail position in meters, ranging from -20 to 40. The blue, green, and red dots indicate the actual measured positions for X, Y, and Z respectively, while the smooth curves in corresponding colors represent the estimated positions. t is probable that this graph is used to assess the error of tracking or predicting algorithm where one would plot the estimated position against the time and the plotted position against the actual measured position against time.

Fig. 11 shows the position error of the spacecraft's center of mass (CoM) over time across three axes: X, Y, and Z. The position error indicates how much the spacecraft's actual position deviates from its position. The x-axis signifies the number of samples (time), while the y-axis shows the position error in meters. The blue, green, and red lines correspond to the X, Y, and Z axes, respectively. The y-axis showing the error in meters (ranging from -2 to 3 meters) and the x-axis showing the number of samples (from 0 to 70).

Fig. 12 represents the change of the attitude error of a spacecraft's center of mass over time. Attitude in spacecraft pose estimation means the orientation of the spacecraft in space. The attitude error shows the difference of the actual orientation with the required degrees of orientation. The value on the horizontal axis is that of sample number with the values ranging from 1 to a figure slightly below 70. The y-axis represents the ˜attitude error' in degrees scale with the range of roughly 2 to 8 degrees. On the graph presented below, the changes in the attitude error can be observed with clear elevation and decline periods. This variability implies fluctuations of the stability or the control system performance of an object.

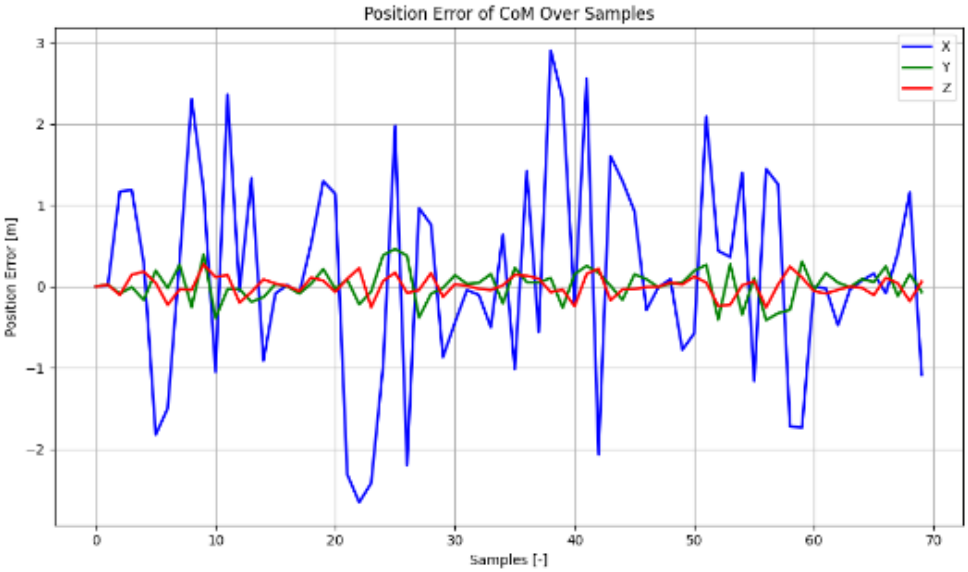

Fig. 11. Position error of CoM.

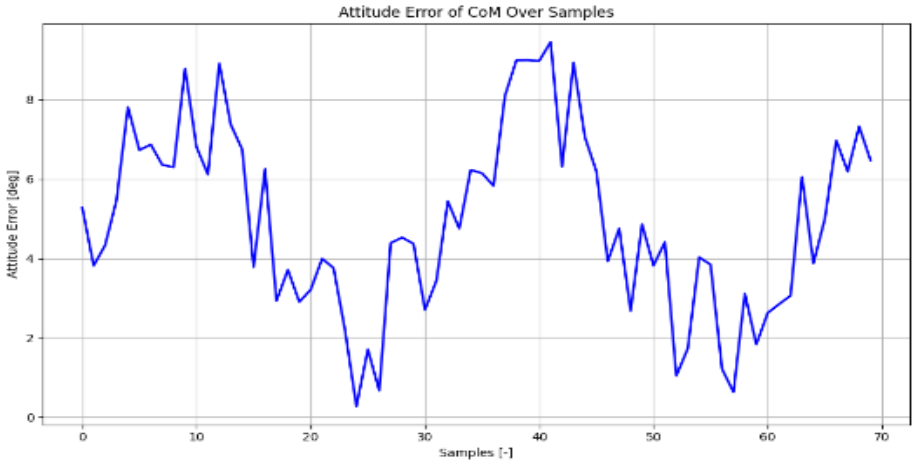

Fig. 12. Attitude error of CoM over samples.

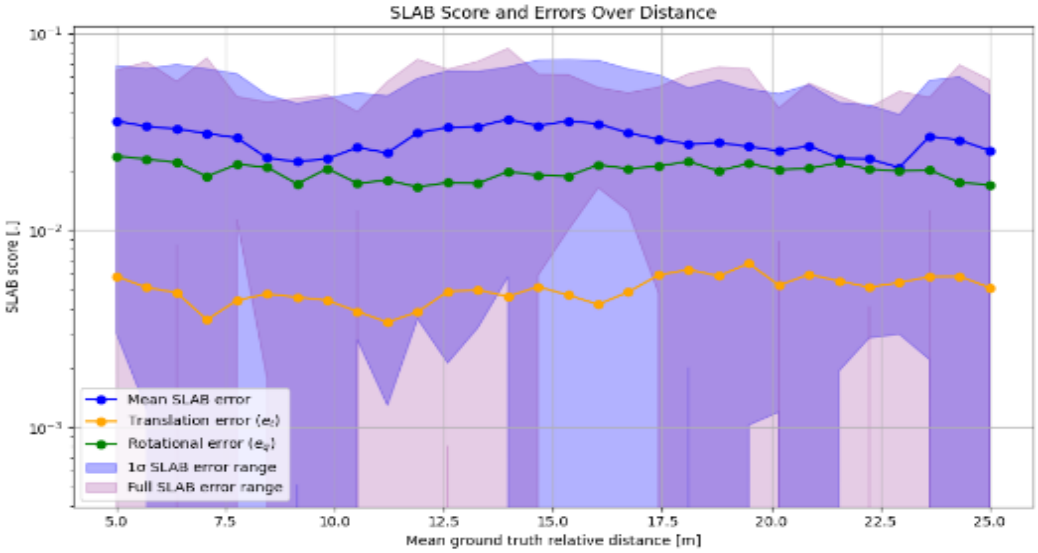

Fig. 13. SLAB Score and errors over distance.

Fig. 13 shows how distances affect pose estimation errors of spacecrafts the range of distances is shown on the horizontal axis, while the vertical axis describes SLAB scores in logarithmic degrees. The blue line represents the Mean SLAB Error, meaning that it presents the mean pose estimation errors. The green line represents Translational Error which is exclusively related to the errors of spacecraft's movement. The dark blue line with circle markers represents Rotational Error. The orange line gives the Full of SLAB Error Range to get the overall idea of errors. Grey-shaded areas in light purple and orange represent error variability, including the full range of SLAB error range, as well as 1σ SLAB error bars. This graph is important as it depicts the manner in which the accuracy and reliability of pose estimate are impacted by the distance from the spacecraft to its object.

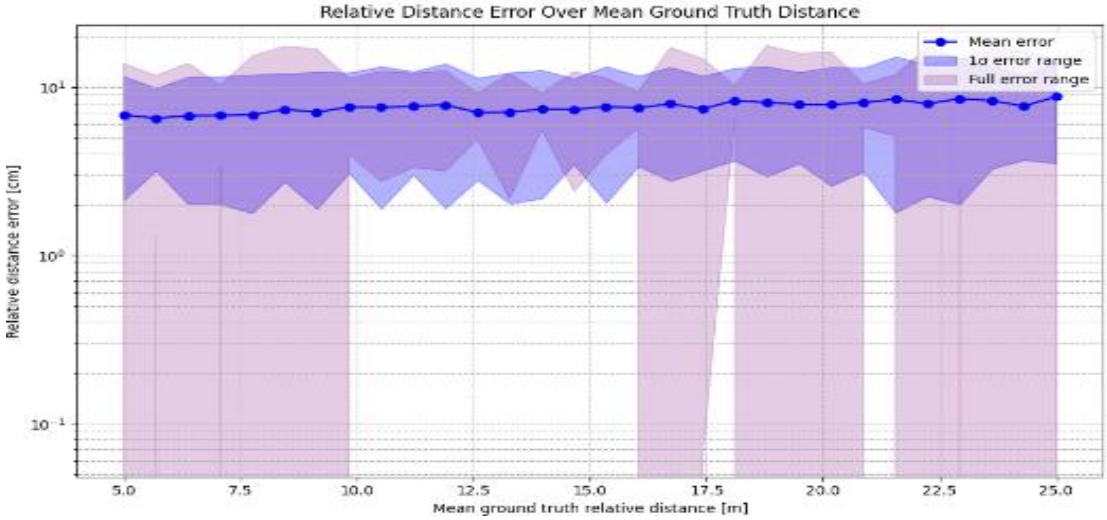

Fig. 14. Relative distance error over mean ground truth distance.

Fig. 14 shows the accuracy rates of the estimation of the position of a spacecraft at different distances. The horizontal axis depicts the average ground truth distance in meters (0 – 25 meters) while the vertical axis depicts the relative distance error in log-log scale in centimeters where values ranges from 0. 01 centimeters to 10 centimeters. The solid blue curve presents the average error and despite the increase of distance this value does not change dramatically. The region between this line and the light purple colored area is the distribution of data within one standard deviation (1σ Error Range) and the darker colored area represents the range of errors observed which is the maximum and minimum. This graph is needed for determining the accuracy of the spacecraft pose estimation, especially during essential operations such as docking or landing, as the nature of error dependence on distance can be observed from this graph.

Fig. 15 shows how accurately a spacecraft's orientation can be determined over varying distances. The mean error line shows the average deviation from the true pose, while the 1σ error range and full error range illustrate the variability and extremes of these errors. This helps in assessing the reliability and precision of the pose estimation system, which is vital for navigation, docking, and other critical operations in space missions. By analyzing the Attitude Error Analysis Based on Ground Truth Distance pose of a spacecraft, the Relative Attitude Error Over Mean Ground Truth Distance is important to know how well orientation measurements of a spacecraft can be from far and near. The mean error line was used to indicate the average distance off true pose, while the 1σ to demonstrate the spread of these errors and full error range shown the overall high/low of these errors.

Table II evaluates four object detection methods based on their Intersection over Union (IOU) scores, orientation errors $\xi_R$and localization errors ξ_TThe SPN method exhibits moderate IOU scores but shows relatively high orientation and localization errors, indicating less accuracy in detecting object positions and orientations. HRNet+PE excels with the highest IOU scores and the lowest errors in both orientation and localization, reflecting superior precision and accuracy. URSONet presents lower IOU scores and significant errors in orientation and localization, suggesting lower overall performance. The Proposed method combines high IOU scores with competitive orientation and localization errors, indicating a well-balanced approach with effective accuracy and precision in object detection.

B. Discussions

The present research offers a revolutionary method to spacecraft pose estimation through the implementation of a Dual-Channel Transformer Network coupled with Bayesian Optimization, providing a crucial improvement over other conventional methods such as CNNs, SLAM, and Particle Filters. With the use of EfficientNet to achieve stable feature extraction and splitting translation and orientation predictions using specific subnetworks, the model is able to capture both the spatial and rotational features of spacecraft from challenging satellite images [25]. Employment of Bayesian Optimization in tandem with Gaussian Process Regression and UCB acquisition fine-tunes the model through optimized hyperparameters via low-order evaluation, raising the bar of both accuracy and computation. Demonstrated on the SLAB data, the solution worked with far superior generalizability and precision in a multitude of scenarios and exemplified potential deployment in actual operational autonomous space settings. In comparison to

current research, this work not only obtains better estimation performance but also proposes a more scalable and flexible method. Outcomes bridge gaps in literature by offering solutions to the most critical challenges of domain shift, computational expense, and sensor noise sensitivity, building a strong platform for future development in deep learning-based space navigation and robotics.

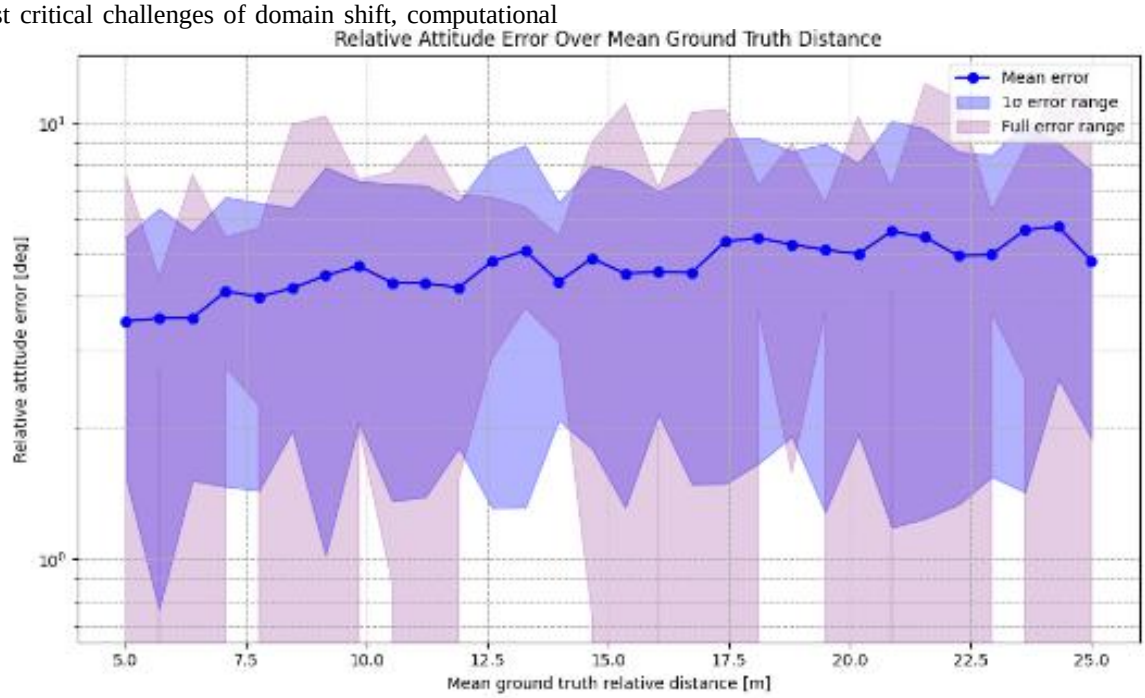

Fig. 15. Relative attitude error analysis based on ground truth distance.

TABLE II. PERFORMANCE COMPARISON OF OBJECT DETECTION METHODS: LOCALIZATION AND ORIENTATION ERRORS

Method	Mean IOU	Median IOU	Mean ξ_R (degree)	Median ξ_R (degree)	Mean ξ_T (m)	Median ξ_T (m)
SPN [22]	0.8582	0.8908	8.4254	7.0689	0.2937	0.1803
HRNet+PE [23]	0.9534	0.9634	0.7277	0.5214	0.0359	0.0147
URSONet [24]			3.1036	2.6205	2.1890	1.2718
Proposed	0.9610	0.9727	0.6812	0.5027	0.0320	0.0144

VI. CONCLUSION AND FUTURE WORK

This is a novel research and innovation in the field of spacecraft pose estimation which utilizes Dual-Channel Transformer Model with EfficientNet is as feature extractor and Bayesian Optimization is used for hyperparameters tuning. The proposed method has shown a clear advantage in terms of translational and rotational accuracy over traditional methods. EfficientNet made the model able to comprehend complex spatial and rotation characteristics of the spacecraft, while dual subnetworks focusing on translation and orientation contributed to improved pose estimation accuracy. Bayesian Optimization as an optimization algorithm using Gaussian Processes with Upper Confidence Bound (UCB) acquisition function enabled adequate hyperparameter tuning, with a reduced number of function evaluations. Results obtained by validating this new approach on SLAB dataset shows significant improvement regarding position and attitude estimation for different distances, confirming advantages presented by this innovative concept in real-world applications. Even though the progress achieved, several directions should be further explored. First, more diverse data should be used to establish a model with stronger generality. Data from different types of spacecrafts under various environmental conditions need to be included in the training and testing datasets to improve the generality of the proposed method and verify its effectiveness under more general settings. Real-time learning can also be integrated into the model so that onboard or native spacecraft climate data can be continuously accumulated to update (train) the current deep learning models during missions. This would make it feasible for the long-duration application of a deep-learning-based model in varying space environments. Hybrid optimization algorithms such as coupling Bayesian optimization with genetic algorithms or reinforcement learning could potentially enhance both computational efficiency and modeling accuracy. Furthermore, expanding this work from attitude estimation to other applications, including velocity estimation or fuel efficiency optimization and control, will significantly increase our capability in exploring state-of-the-art technologies using attitude as well as other critical information in modern autonomous navigation tasks of docking, rendezvous and landing. This work establishes a new state-of-the-art for spacecraft pose estimation, but continued advancements in adaptability, real-time learning, and more extensive parameter estimation should allow even higher levels of accuracy and efficiency for spaceflight missions.

REFERENCES

[1] C. Vela, G. Fasano, and R. Opromolla, "Pose determination of passively cooperative spacecraft in close proximity using a monocular camera and AruCo markers," Acta Astronautica, vol. 201, pp. 22–38, 2022.

[2] T. H. Park et al., "Satellite pose estimation competition 2021: Results and analyses," Acta Astronautica, vol. 204, pp. 640–665, 2023.

[3] L. Pauly, W. Rharbaoui, C. Shneider, A. Rathinam, V. Gaudillière, and D. Aouada, "A survey on deep learning-based monocular spacecraft pose estimation: Current state, limitations and prospects," Acta Astronautica, vol. 212, pp. 339–360, 2023.

[4] A. M. Heintz and M. Peck, "Spacecraft state estimation using neural radiance fields," Journal of Guidance, Control, and Dynamics, vol. 46, no. 8, pp. 1596–1609, 2023.

[5] A. Lotti, D. Modenini, P. Tortora, M. Saponara, and M. A. Perino, "Deep learning for real-time satellite pose estimation on tensor processing units," Journal of Spacecraft and Rockets, vol. 60, no. 3, pp. 1034–1038, 2023.

[6] S. Kaki, J. Deutsch, K. Black, A. Cura-Portillo, B. A. Jones, and M. R. Akella, "Real-time image-based relative pose estimation and filtering for spacecraft applications," Journal of Aerospace Information Systems, vol. 20, no. 6, pp. 290–307, 2023.

[7] P. F. Proença and Y. Gao, "Deep learning for spacecraft pose estimation from photorealistic rendering," in 2020 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2020, pp. 6007–6013.

[8] D. Rondao, N. Aouf, and M. A. Richardson, "ChiNet: Deep recurrent convolutional learning for multimodal spacecraft pose estimation," IEEE Transactions on Aerospace and Electronic Systems, vol. 59, no. 2, pp. 937–949, 2022.

[9] A. Garcia et al., "Lspnet: A 2d localization-oriented spacecraft pose estimation neural network," in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2021, pp. 2048–2056.

[10] K. Li, H. Zhang, and C. Hu, "Learning-based pose estimation of non-cooperative spacecrafts with uncertainty prediction," Aerospace, vol. 9, no. 10, p. 592, 2022.

[11] S. Moon, S.-Y. Park, S. Jeon, and D.-E. Kang, "Design and verification of spacecraft pose estimation algorithm using deep learning," Journal of Astronomy and Space Sciences, vol. 41, no. 2, pp. 61–78, 2024.

[12] K. Cosmas and A. Kenichi, "Utilization of FPGA for onboard inference of landmark localization in CNN-based spacecraft pose estimation," Aerospace, vol. 7, no. 11, p. 159, 2020.

[13] H. Viggh, S. Loughran, Y. Rachlin, R. Allen, and J. Ruprecht, "Training deep learning spacecraft component detection algorithms using synthetic image data," in 2023 IEEE Aerospace Conference, IEEE, 2023, pp. 1–13.

[14] L. Yingxiao, H. Ju, M. Ping, and others, "Target localization method of non-cooperative spacecraft on on-orbit service," Chinese Journal of Aeronautics, vol. 35, no. 11, pp. 336–348, 2022.

[15] M. Bechini, P. Lunghi, M. Lavagna, and others, "Spacecraft pose estimation via monocular image processing: Dataset generation and validation," in 9th European Conference for Aerospace Sciences (EUCASS 2022), 2022, pp. 1–15.

[16] S. Sharma, C. Beierle, and S. D'Amico, "Pose estimation for non-cooperative spacecraft rendezvous using convolutional neural networks," in 2018 IEEE Aerospace Conference, IEEE, 2018, pp. 1–12.

[17] T. H. Park et al., "Satellite Pose Estimation Competition 2021: Results and Analyses," Acta Astronautica, vol. 204, pp. 640–665, Mar. 2023, doi: 10.1016/j.actaastro.2023.01.002.

[18] M. Salvi, U. R. Acharya, F. Molinari, and K. M. Meiburger, "The impact of pre- and post-image processing techniques on deep learning frameworks: A comprehensive review for digital pathology image analysis," Computers in Biology and Medicine, vol. 128, p. 104129, Jan. 2021, doi: 10.1016/j.compbiomed.2020.104129.

[19] K. Maharana, S. Mondal, and B. Nemade, "A review: Data pre-processing and data augmentation techniques," Global Transitions Proceedings, vol. 3, no. 1, pp. 91–99, Jun. 2022, doi: 10.1016/j.gltp.2022.04.020.

[20] N. B. Le Duy Huynh, "A u-net++ with pre-trained efficientnet backbone for segmentation of diseases and artifacts in endoscopy images and videos," in CEUR Workshop Proceedings, 2020, pp. 13–17.

[21] F. Hagelskjær, N. Krüger, and A. G. Buch, "Bayesian optimization of 3d feature parameters for 6d pose estimation," in 14th International Conference on Computer Vision Theory and Applications, SCITEPRESS Digital Library, 2019, pp. 135–142.

[22] S. Sharma and S. D'Amico, "Pose Estimation for Non-Cooperative Rendezvous Using Neural Networks," 2019, arXiv. doi: 10.48550/ARXIV.1906.09868.

[23] B. Chen, J. Cao, A. Parra, and T.-J. Chin, "Satellite pose estimation with deep landmark regression and nonlinear pose refinement," in Proceedings of the IEEE/CVF international conference on computer vision workshops, 2019, pp. 0–0.

[24] P. F. Proença and Y. Gao, "Deep learning for spacecraft pose estimation from photorealistic rendering," in 2020 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2020, pp. 6007–6013.

[25] P. F. Proença and Y. Gao, "Deep learning for spacecraft pose estimation from photorealistic rendering," in 2020 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2020, pp. 6007–6013