ISSN: 2584-0495 Vol. 1, Issue 3, pp.163- 172

International Journal of Microsystems and IoT

ISSN: (Online) Journal homepage: https://www.ijmit.org

PaPEGAN: Parallelized Pose Estimation using Generative Adversarial Networks

Prabakaran N, Aditya Deepak Joshi, Rajasekaran P

Cite as: Prabakaran N, Aditya Deepak Joshi, & Rajasekaran P. (2023). PaPEGAN: Parallelized Pose Estimation using Generative Adversarial Networks. International Journal of Microsystems and IoT, 1(3), 163-172. https://doi.org/10.5281/zenodo.8362011

Published online: 21 August 2023.

DOI: https://doi.org/10.5281/zenodo.8362011

International Journal of Microsystems and IoT
Vol. 1, Issue 3, pp. 163-172; DOI: https://doi.org/10.5281/zenodo.8362011

PaPEGAN: Parallelized Pose Estimation using Generative Adversarial Networks

Prabakaran N[1], Aditya Deepak Joshi[2], Rajasekaran P[3*]

[1] School of Computer Science & Engineering, Vellore Institute of Technology, Vellore, India.
[2] Department of Computer Science & Engineering, New York University, New York City, USA.
[3*]Department of Computing Technologies, SRM Institute of Science and Technology, Chengalpattu, India.

ABSTRACT

Space and space-exploration have always provided mankind with a sense of mystery and aura. In the pursuit of quelling this curiosity man has invented spacecrafts to look into the depths of the cosmos. In this paper the researcher presents a novel architecture that combines the technologies of Generative Adversarial Networks (GANs), process-based parallelism and Convolutional Neural Network (CNN) based Pose Estimation of various spacecraft (for instance satellites and spaceships). The GANs with its generator and discriminator provides an increased number of images to augment the training possible using current datasets. The successful implementation of PaPEGAN will significantly advance the field of spacecraft pose estimation, in terms of accuracy and efficiency. The impact of this research extends to various space exploration applications, such as satellite navigation, robotic spacecraft control, autonomous landing, and trajectory planning, ultimately enhancing the success and effectiveness of space missions. Overall, the problem is to develop and demonstrate the effectiveness of PaPEGAN as an advanced solution for spacecraft pose estimation, revolutionizing the capabilities and performance of navigation and control systems in the realm of space exploration.

Keywords

6 Degrees of Freedom Pose Estimation; Generative Adversarial Networks; Horovod Parallel Computing.

1. INTRODUCTION

Space exploration has been an enduring endeavour that continues to captivate humanity, fuelling our innate curiosity and desire to unveil the mysteries of the cosmos. Over the years, spacecraft have emerged as vital tools in our quest to delve into the depths of space, enabling us to perform crucial tasks such as satellite navigation, robotic spacecraft control, autonomous landing, and trajectory planning. One fundamental aspect in the successful operation of spacecraft is the ability to accurately estimate their poses in the six axes of motion, commonly referred to as the six Degrees of Freedom (6DOF). The spacecraft's pose, which encompasses its position and orientation, plays a pivotal role in ensuring the efficiency, safety, and success of space missions. Added to this are the catastrophic historical results of instances when lapses in calculations and measurements in space-objects occur. Examples such as the Mars Climate Orbiter (1999), Genesis Sample Return Mission (2004) and perhaps most famously the Galaxy 4 satellite (1998) which unfortunately was unable to maintain its upright orientation. These examples have time and again demonstrated to us the unforgiving nature of space, and the level of perfection demanded from any endeavours related to it.

This paper presents an innovative architecture named PaPEGAN (Parallelized Pose Estimation using Generative Adversarial Networks), which leverages cutting-edge technologies such as Generative Adversarial Networks (GANs), process-based parallelism, and Convolutional Neural Networks (CNNs) for spacecraft pose estimation. GANs have shown great promise in generating high-quality synthetic data, and when combined with CNN-based pose estimation, they offer a powerful solution to augment the training process using existing datasets.

To address these challenges and advance the field of spacecraft pose estimation, novel approaches that leverage the power of deep learning and data augmentation are necessary. Deep learning techniques have demonstrated remarkable success in various image processing applications, such as self-driving cars and factory robots, but the scarcity of hands-on experimentation in space necessitates innovative strategies. Secondly, and perhaps more importantly, GANs are changing the face of generative models by boasting capabilities like producing immaculate unmarked data that is indiscernible by the human eye. The resultant data have been potent in the creation of art, style transfer, and image-to-image translation. A vanilla GAN consists of two adversarial machine learning models aptly named the generator and the discriminator. Basis a probability distribution as well as the existing real data present in a dataset, the generator will, depending on the specified learning parameters generate synthetic data. Iteratively, this data gets incrementally better till it gets to a point where the discriminator also until this point has been learning the unseen and latent patterns in the data synthesising mathematical features that determine the real-ness and the fakeness of any given data. In this chapter we aim to provide the level of quality that

not only mislead the discriminator but also be able to be a learning material that a CNN may work on, as demonstrated by [15]. This understandably, is a continuous process of learning as, the GAN can be trained until a point of mode collapse, as referenced from [14] appending enough images to the existing dataset while training the CNN on a distributed framework of workers making the timely training of the same feasible.

2. RELATED WORKS

For any moving object in space, owing to the decreased amount of physical control, the increased entropy as well as the long list of consequences in case something goes wrong it becomes essential to estimate the pose of objects. The importance of solving this problem is emphasized by the extremely fundamental functions that are achieved by spacecrafts be they debris avoidance, docking with other spacecraft or simply orienting the spacecraft in a desired fashion. Furthermore, the certain satellites are launched with chaser satellites such as the ones titled Tango and Mango, designed and developed by the Swedish Space Corporation (SSC) [1]. This mission titled PRISMA (Prototype Research Instruments and Space Mission technology Advancement) posed a new problem to future space researchers of determining the position and orientation of the primary satellite using an image taken from the point of view of the secondary chaser satellite.

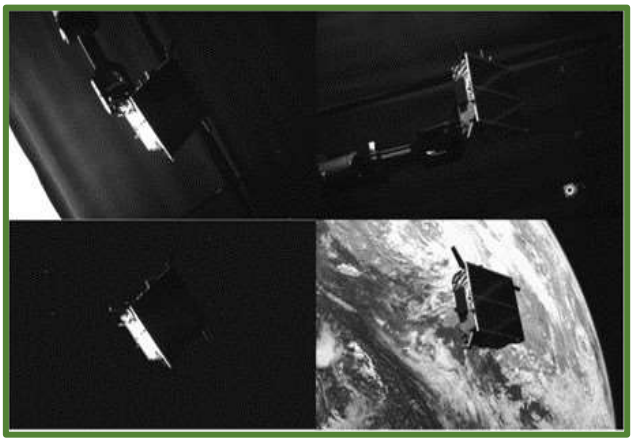

Fig. 1 Sample images from Speed+ database

Figure 1 depicts the sample images from Speed+ database. The solution to this problem is further complicated by the lack of 'which way is up' in the oblivion of space, furthermore another stark difference to terrestrial conditions is the lack of atmospheric obfuscation (a common problem faced by low earth orbit satellites) while this is an advantage that allows imaging over long distances it also massively complicates the target illumination in many situations. Secondly one is limited to a handful of information mediums in the form of images, radio waves etc. When discussing existing methods of Pose Estimation, we encounter techniques like Inertial Measurement Units (IMU), that use the accelerometers and gyroscopes on board to calculate the linear acceleration and angular velocity respectively. These however run into eventual sensor drift over time making their errors to cascade and add up over time. These traditional approaches usually involve handcrafted descriptors for various features of images. The descriptors are mapped and recognized using the various geometrical features of the images. These techniques though valid and solid in the face of certain problems, the available hardware encompasses monocular pose scenarios as well as when there is a large variation between the well-lit and poorly lit aspects of the image, which as mentioned before is a common occurrence in space.

Furthermore, without the presence of an astute verification method, it becomes difficult to cross-check the results of any given instrument measures. This can lead to a phenomenon called as error propagation, as explained in a slightly different but relevant concept the authors of [12] discuss the implications of, and the reasons for the introduction of error propagation in the orbital prediction process.

Considering the dire and serious nature of the problem there exist dedicated datasets for this purpose, Spacecraft PosE Estimation Dataset (SPEED) as well as its far expanded counterpart SPEED+. As discussed in [2,3] the former dataset consists of around 15,000 synthetic images and 300 real images. The newer and improved SPEED+ dataset consists of 60,000 synthetic images and an additional 9531 Hardware-in-Loop (HIL) test images taken with a lightbox and a sunlamp. These images are taken at a facility named the SLAB (Space Rendezvous Laboratory) at Stanford.

Initially though the size may seem large, for a deep learning task it sadly might not provide the best of results considering the criticality of applications and possible orientations, one does not have to imagine too much as to why an increased number of images with increased variations may be provided for the model to learn. Additionally, unlike the popular image processing centric deep learning applications such as self-driving cars as well as factory robots, we do not possess the ability to hands-on experiment with our test subject in its environment of operation, outer space. The solution to this problem is to augment this dataset using a Generative Adversarial Network (GAN) Architecture. The constituent parts of the GANs as introduced in [4] are generators and discriminators that act as counterfeiters generating phony images, which due to the machine learning logic of the discriminator progressively get better and closer to the real thing respectively. The generator gets an input of a noise vector from a probabilistic distribution such as a Gaussian Distribution, which it uses in tandem with the dataset provided to it, in our case the SPEED+ dataset. Using these two it picks the essential latent features of the image to vary these parameters and create an entirely new image. In our case these "phony" generated images mean non-existent images that act as newer training scenarios that the pose estimating architecture could be exposed to.

Additionally, as proposed by the authors of [14], GANs do and will hold great roles in the consumer deep learning industry, specifically in the role of supporting other critical computer vision applications that lack the possibility of collecting actual, non-synthetic data.

This further exemplified by the reality of the space industry were operating the rig for the purposes, and the scale of generating images for deep learning tasks is intractable it bodes the requirement of utilising GANs to generate these images to, in essence fill the gaps. Secondly, the authors of paper [13] also demonstrated the quality of the results given by GANs in the field of animated characters, inculcating the preliminary and fundamental qualities of GANs such as style transfer and creating synthetic data. Therefore, combining the need, the quality, and the feasibility we get the root of innovation present at the crux of this chapter.

Additionally, as proposed by the authors of [14], GANs do and will hold great roles in the consumer deep learning industry, specifically in the role of supporting other critical computer vision applications that lack the possibility of collecting actual, non-synthetic data.

This further exemplified by the reality of the space industry were operating the rig for the purposes, and the scale of generating images for deep learning tasks is intractable it bodes the requirement of utilising GANs to generate these images to, in essence fill the gaps. Secondly, the authors of paper [13] also demonstrated the quality of the results given by GANs in the field of animated characters, inculcating the preliminary and fundamental qualities of GANs such as style transfer and creating synthetic data. Therefore, combining the need, the quality and the feasibility we get the root of innovation present at the crux of this chapter.

$$F(Q,T) := M_{x\sim q\ data(x)}[\log\ T(x)] + M_{z\sim p(z)}[\log\ (1 - T(q(z)))] \quad (1)$$

The above Equation 1 formulates the loss function for the GANs represented by F. The first term calculates the average of the log of the discriminator's output for data sample x, sampled from the real data distribution $q_{data}(x)$. The mean the discriminator's adeptness to classify a real image correctly. The method used to examine the difference between phony and real images is to think of the discriminator as a binary classifier. It produces a probability between 0 and 1 where values closer to 1 indicating that the input data is authentic while 0 indicates fake data. It is first being trained for 'n' epochs with a stream of real ground truth images from the dataset. Next it is fed the generated images from the generator. The quickness with which the discriminator, discriminates between the two kinds of data (real and phoney) is the penalty the generator receives. The reader is advised to keep in mind that the quickness does not refer to the time rather the difference between the generated data and the stream of real data it previously received. The discriminator's learning process occurs in an unsupervised manner. The examination of the inherent patterns and features found in the real data distribution allows generalization of these patterns to effectively characterize the entire category of data. An analogy would be to mathematically construct a generalised genre of data. Subsequently, this characterization aids in distinguishing the presence or absence of these characteristics even in synthetic images.

The second component measures the discriminator's capability to detect generated content from the generator. It involves evaluating the discriminator's output (using parameters T) when identifying the generated content (represented by q) produced from synthetic data z sampled from a noise distribution p(z) in equation 2. This objective function primarily assesses how well the discriminator can differentiate real images from fake ones. During discriminator training, the focus is on minimizing this loss, simultaneously maximizing the probability of correctly identifying real images and fake images.

$$L_G := -M_{z\sim p(z)}[\log\ (1 - T(q(z)))] \quad (2)$$

The GANs training process occurs in a start stop, staggered fashion. First the generator is trained on the real data, it is then provided a probability distribution to fit with the analyzed latent aspects of said data. The training process consists of learnable parameters annotated by Q that are iteratively refined during the training process. The training process of the generator is then momentarily paused before the training of the discriminator is resumed. It learns to map the data vector z to a high-dimensional generated object, like an image. The generator's loss is calculated as the negative of the discriminator's output, reflecting the probability of the discriminator incorrectly classifying the generated image as real.

$$\min_{Q}\max_{T} V(Q,T) \quad (3)$$

The training process then becomes an iterative adversarial game, as illustrated in equation 3. The adversaries, namely the generator and the discriminator respectively attempt to minimize parameters Q and maximize parameters T. This encourages the generator to improve its ability to generate increasingly realistic images, and the discriminator to enhance its ability to distinguish fake and real images. The process continues until a point of equilibrium, known as the Nash Equilibrium, is reached. At this stage, the discriminator makes random guesses when faced with generated data so realistic that it cannot differentiate between real and fake. While this is the ideal behaviour of GANs, practically achieving this point is highly complex and challenging. The researchers will delve further into the challenges of GANs, including asymptotic convergence and the difficulties in reaching this ideal state.

Finally, the HRNet which is a pose estimation architecture centred around human pose estimating applications [5], has been popularly shown to be especially useful in space-object pose estimation scenarios [6] and further enhanced

using a technique named Key Point Detection in [7]. The deep learning framework proposed by the authors of [7] utilizes multiple Neural Network Frameworks to not only extract the space object from the background but also determine its attitude and translation vector.

This deep learning task owing to its network complexity can be deemed a computationally complex learning task. To alleviate this in terms of runtimes, we inculcate the final piece of the puzzle Horovod [8]. As described by its makers, it is a distributed deep learning training framework that functions in conjunction with libraries like TensorFlow, Keras and Pytorch to buttress their computability by providing the ability to parallelize the learning task through data parallelism.

To understand the functioning of the Horovod technique, one must only take a look at the imagery the makers have in the form of the logo of the same. It is meant to represent a Russian Folk dance of the same name, involving the dancers' holding hands and performing the dance as a circle. This is an analogy to the technique of communication that TensorFlow process utilise for their inter-process communication [16]. In the same blog post they also mention why the need for this arose over the traditional paradigms arose, for their specific needs at Uber for the massive dataset sizes they possess and the deep learning decisions to be made forth that data. They built this technique based on the now deleted blog post made by Chinese giant Baidu, on the ring all reduce to reduce the time spent in communicating between the different computing Graphical Processing Units (GPUs) participating.

Before discussing the mechanics of the ring all reduce algorithm on which our solution's speed gains rely, we must discuss what significant change this brought about in the distribution of learning operations to multiple GPUs. To understand this let us go over a basic technique called the Data Parallel Stochastic Gradient descent (DPSGD). As with the vanilla SGD we perform batching to take on smaller chunks (or minibatches) of the dataset at hand and progress through the entire dataset in this fashion. Now, in DPSGD what we do is partition these minibatches across multiple GPUs while each GPU is assigned a copy of the neural network model being trained. In each iteration, the respective GPU's minibatch is sampled to provide the data to it on which the GPU runs a forward propagation of the network. The gradient loss is computed by subsequently running the error backpropagation. The key communication aspect comes in when the results of the previous step are to be communicated across the different GPUs to compute the average gradients to calculate the weights for the following iteration. This is a problem that can be understood by a very polite airline crew. Imagine the airplane in question is to be filled before it takes to the skies and all but two passengers have boarded the plane. In this situation none of the timely arrived passengers can take advantage of their timeliness and must wait their turn until the last passengers arrive. This elegant analogy explains the problem known as the 'communication problem' where the GPUs are configured to compute in lockstep. This is logistically clear as well since when the average gradient is to be computed it cannot be computed in the absence of the various data members.

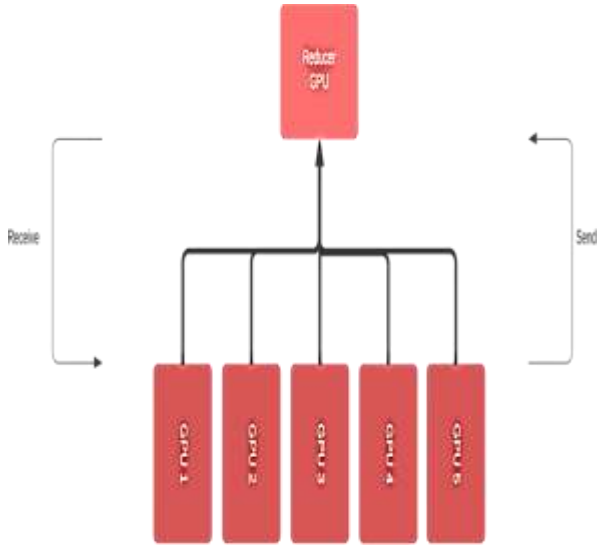

Fig. 2. Data Transfer to single reducer GPU

The above figure, as Gibiansky explains in [17] is the simplest possible means of explaining the blaring issue of time consumption. Furthermore, this was under the assumption that the time required to transmit the data was independent of the amount of the data, while in reality it is not. From the concept of data transmission in networks we are well aware that the higher the number of data packets the higher the transmission time. As the number of GPUs increases, the chances of early completing GPUs having to wait increase, simultaneously so do the data transmission costs. As explained in his write up he explains that the communication costs add up to each iteration and at even a few seconds per GPU, this linear growth in time creates a hard ceiling for the number of GPUs that can be employed. To understand this problem let us take simple network architectures, to understand this problem at a smaller scale and compare these to the scale of a Large Language Model (LLM) like ChatGPT.

In a configuration like in Figur 2, let us assume we possess a total of N GPUs including the reducer GPU. Let us also assume that the results of one iteration of learning are S in size. Therefore, stemming from the N-1 GPUs and their respective N-1 learning processes, the reducer GPU receives D_r data in equation 4.

$$D_r := (N-1) * S \quad (4)$$

Now simplistically speaking, this was the receive phase, now post the averaging we must also transmit the data consisting of the learnings back to the GPUs as well i.e., the total data transmitted D_t becomes as in equation 5.

$$D_t := 2 * D_r \quad (5)$$

$$:= 2 * (N-1) * S \quad (6)$$

The above equation 6 is total data transmitted. The last variable is that of the bandwidth of the communication medium between the various GPU. Let us assume that to be B. The time taken to transmit this data T_t in equation 7

is proportional to the factor N and inversely proportional to the factor B. The latter will become immensely important when considering the numerical example below.

$$T_t := \frac{(2*(N-1)*S)}{B} \quad (7)$$

Currently within the field of research, there is probably no one who is unaware of ChatGPT. It is one of the most powerful LLMs out there with 175 billion trainable parameters, with each parameter requiring approximately 4 bytes. Simple multiplication would say the data to be sent from each GPU becomes 700 billion bytes. Let us assume the standard of bandwidths, in our network architecture, one capable of supporting one gigabyte per second. Therefore, for each iteration the time to transmit this data becomes 700 billion bytes/1,073,741,824 bytes per second. The time taken approximately per send phase becomes 651 seconds per iteration for a single GPU and single reducer GPU communicating. This scale is itself extremely staggering and when distributed to even 5 GPUs adds a factor of almost 4.7 to each iteration's waiting time. Imagine the amount of compute time wasted if this were applied to weeks or months' worth of learning operations. The reader could think of a simpler alternative to this being to eliminate the synchronicity as identified in the dissertation [18] and a solution for which exists in [19]. But as we see, creating the solutions for these is extremely complicated in terms of the batching strategy of the data. This is not to say the SGD paradigm would break down in this scenario, in fact Asynchronous Gradient Descent (ASGD) methods do exist. However, doing so has the possibility to create new race conditions based on the timing of the model parameters being read and updated simultaneously. This newly introduced instability would also result in a subpar network performance in terms of achieving convergence. Therefore now, synchronicity must be maintained while reducing the linear dependence of time on the number of GPUs.

Enter the ring all reduce algorithm which is dependent only on the slowest GPU-GPU link in the computing network. This means this slowest connection is a constant and hence the transmission time becomes a constant. That leaves us with bandwidth being the control variable that determines the transmission time. This gives us a much-required amount of determinism in terms of what to improve to improve the transmission time. The authors of [20] definitively prove ring all reduce to be an optimal communication algorithm when bandwidth is considered as a metric and a sufficiently large buffer is present. Simply put like in Figur 3, we distribute the responsibility of the reduction to the logical hexagonal ring of GPUs. At the risk of oversimplifying, we basically perform a scatter-reduce to ensure every GPU gets a batch or a part of the result. The following all gather step transmits these chunks through the ring such that all the GPUs end up with the complete result. We essentially offload the responsibility of performing the compilation-like reduction to the entire ring and replace few large data transmissions with multiple smaller data transmissions performed an increased number of times.

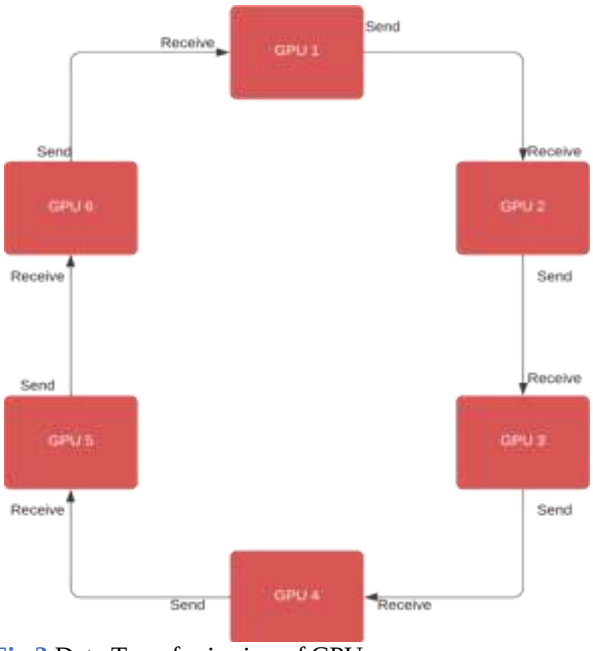

Fig.3 Data Transfer in ring of GPUs

The experimental results found by the Baidu team for a 300 million parameter language model, spread across 40 GPUs synchronously is an excellent sign of performance depicted in [17]. In the Figur.3 we see that each GPU communicates with only its neighbour peers. Each node possesses a buffer to construct the value buffer from the received data from the remaining GPUs.

Now, coming to the pythonic library Horovod. Its research impact has been to substantially increase the access and ease of utilising technologies like MPI to discover GPU workers and integrate the reduce functionality into deep learning research without going into the depths of High-Performance Computing (HPC). This open-source library has been shown to have great success in the field of image segmentation as well as image classification in [9-11] respectively. The research conclusions of these align with our theoretical expectations of Horovod in our use case.

Traditional methods for pose estimation often struggle to provide accurate and efficient results, limiting the effectiveness of spacecraft mission planning, navigation, and control systems. To overcome these limitations, the researcher proposes the use of PaPEGAN an innovative architecture that leverages the power of Generative Adversarial Networks (GANs) and parallel processing techniques to enhance the accuracy and efficiency of spacecraft pose estimation. The objective is to develop an approach that can handle the complexities of pose estimation in a parallelized manner, enabling real-time or near real-time estimation of spacecraft position and orientation. The key challenges to address include training the GAN model to generate high-quality and diverse pose estimations [28] varying parameters of illumination ground truth orientation etc. thereby augmenting and appending the dataset substantially, improving the HRNet performance.

3. METHODOLOGY

In continuation to the problem statement the researcher proposes the pipeline in Fig.4. The process flow begins with the Generative Adversarial Network [29], in our case an architecture based on StyleGAN2. As highlighted in the figure it has a sequential set of Fully Connected (FC) layers, in essence convolutional layers. The first layer takes a noise vector of the size latent size, essentially a representation of a random latent tensor. Using this noise vector, the Generator represented in green, generates a phony image. In the initial epochs this generated image is a series of splotches of black and white on a background as the generator progressively learns the difference between what a space object is, then what its complex and simple backgrounds are then on varying it from a dark to an illuminated image not necessarily in that order. Owing to this unsupervised nature of the architecture we are only able to control so much using the parameters hence we give it ample learning time to create a capable Discriminator Generator pair. This is a balancing act between the compute time as well as the quality of results as for instance the training of 21 epochs takes in excess of 19 hours while producing better results than obviously one trained for 10 epochs over a period of 6 hours.

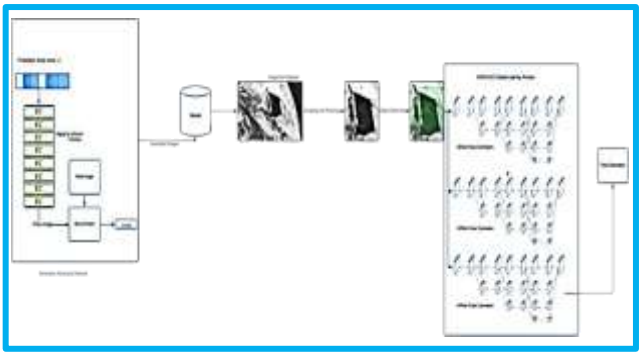

Fig. 4 Architecture Diagram of Pipeline

Following this we append and combine the results with the SPEED+ dataset. Now, before we are able to, we focus on the conversion of the image to a suitable form to be usable to the HRNet architecture. This process also known as preprocessing involves going over image by image the union of datasets we have created and respectively, cropping, resizing, and centring of images to provide the best possible compatibility to the HRNet architecture. As highlighted by the representative green image, we centre the image such that the space object is in the centre of the frame reducing the presence of the background [21-27]. The reader at this point would pose the question, what was the point of generating the synthetic images if we are going to trim the backgrounds from the images? Secondly why are we performing this lengthy process after spending precious compute time generating the complex planetary backgrounds? The answers to these questions are respectively, this step is done to improve the ability of the landmarking process. To determine the key points on the space object we need to refocus the image to the points on the cuboidal surface that make the most significance. The answer to the second question lies in the first one, to improve the ability to centre the image we must supply the HRNet with an increased number of images.

Finally, the Horovod comes into the picture. We use it to distribute the deep learning Residual neural Network (ResNet) and the Convolutional Neural Network (CNN) across 3 separate computing units over the internet. These possess synthetic NVIDIA Pascal GPUs to partition our learning task we partition the newly pre-processed dataset into three parts and provide this to the HRNet. This will allow us to simultaneously train these networks on the partitioned datasets, in essence drastically reducing the time per epoch drastically across the three workers, in our case the three GPU virtual computers. The Horovod approach as referenced in the literature has immense success in the image segmentation domain, this will allow us to, successively improve the training results over time. The result of this is below in Figure.5, given the grainy results the computed orientation is remarkably precise.

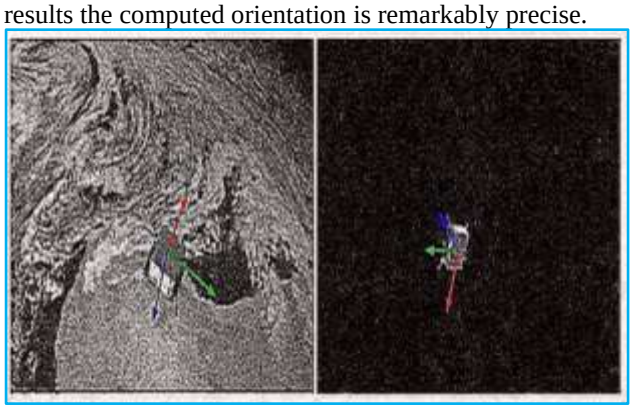

Fig.5 Final Pose Estimate of two space objects with (a)complex and (b)simple backgrounds

4. RESULTS

The first graph in Figure4, emphasizes the need we have for Horovod when distributed across 3 workers. We see the runtimes per epoch of the parallelized workers in the training process are not just approximately 3 times faster but also averaging the runtimes consistently perform better so. These results are tabulated in Table 1.

Table. 1 Epochs of single core and multi core time of 3 workers

Epochs	Single Core Time	Multi Core Time (3 workers)
1	1211.2000	399.6612
2	1242.6998	403.5486
3	1295.5000	413.6831
4	1254.6998	436.3776
5	1258.2007	564.3470
6	1297.6004	269.1826

7	1276.8996	617.7330
8	1305.7993	111.9592
9	1292.3001	685.0201
10	1288.5985	179.9809
11	1256.3001	484.9999
12	1261.3996	189.7941
13	1298.5985	557.1765
14	1315.1022	544.5404
15	1274.2007	957.9896
16	1271.3978	485.9975
17	1245.4015	605.2488
18	1275.7993	855.9455
19	1277.0000	191.5956
20	1271.4015	330.2006

Table 2 gives the epochs between real scores and fake scores.

Table. 2 epochs between real scores and fake scores.

Epochs	Real Scores	Fake Scores
1	0.8263	0.2548
2	0.7175	0.2701
3	0.7483	0.1527
4	0.8068	0.1455
5	0.6547	0.071
6	0.5877	0.0136
7	0.6952	0.033
8	0.9228	0.085
9	0.988	0.1046
10	0.991	0.038
11	0.9508	0.0593
12	0.9276	0.0048
13	0.9276	0.1095
14	0.95	0.1955
15	0.6859	0.0171
16	0.8061	0.0359
17	0.9725	0.2269
18	0.9618	0.0445
19	0.9543	0.0991
20	0.9724	0.0646
21	0.8402	0.0771

In Figure. 5, we see, the higher real scores and lower fake scores indicate improved discrimination between real and fake samples, the consistently high real scores represent its ability to identify real examples. This is a good sign as the discriminator has and maintains its ability to correctly identify real images as real and phony images as phony.

Secondly other than the initial spike of the Generator we see a consistent rise after epoch 6 in Fake scores as well. Figure.6 represents a positive trend as well, a decreasing discriminator loss starting epoch 2 which again demonstrates its improving capability to determine real examples from incorrect ones.

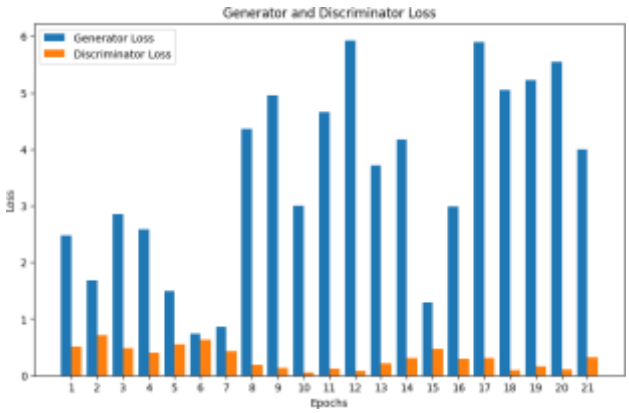

Fig.6 Generator and Discriminator Loss of GAN

Table 3 gives the details about generator and discriminator Loss of GAN.

Table. 3 Generator and Discriminator Loss of GAN

Epochs	Generator Loss	Discriminator Loss
1	2.4771	0.5069
2	1.6762	0.71
3	2.8544	0.4779
4	2.5777	0.4021
5	1.4855	0.5436
6	0.7361	0.6304
7	0.8619	0.4315
8	4.3599	0.1878
9	4.959	0.127
10	3.0053	0.0486
11	4.6555	0.1173
12	5.9277	0.0826
13	3.7191	0.2123
14	4.1788	0.3012
15	1.2851	0.463
16	2.983	0.2947
17	5.8916	0.3077
18	5.0542	0.0889
19	5.2179	0.1625
20	5.5496	0.1005
21	3.9941	0.3141

Finally, the last image is 4 new generated images, these images are to be refocused, illuminated, and zoomed to be then passed onto the Horovod architecture. In these we clearly see Fig7 has a dark background but is more complex. These space objects being complex shapes is a good sign in terms of realism. Fig 7 is the final output.

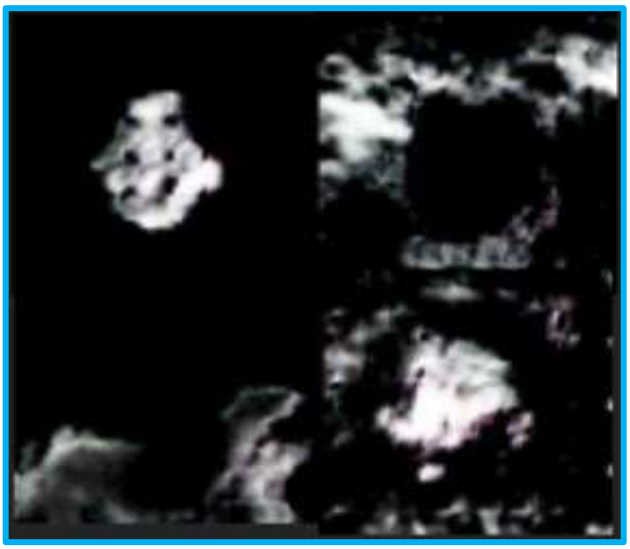

Fig.7 Sample Generated Images from GAN

5. CONCLUSION AND FUTURE SCOPE

After the completion of this comprehensive research, we see the final results to be comprehensive and extremely encouraging. As with all research we can expect improved results with increased compute time and number of workers beyond the already available consumer servers online. Additionally, another avenue to be explored is that of video clips of space objects. In our example we have considered a singular approach across the domain of satellite like objects, asteroid like objects, space junks etc. This may not be the best approach as there are other approaches like Star Trackers and Sun Sensors which can be used in combination with this vision-based approach. To elaborate the previous point, we need to also consider situation where the image clarity may not be up to par and may need to upscale the image while maintaining its integrity in the space background scenario. Finally, we encourage future researchers to explore different combinations of data points given initially, as in our case we are considering visible light, while in other scenarios we may possess other forms of electromagnetic radiation as shadow images of the space objects. These situations may involve alternate forms of data and signal processing however the data generation using GAN may remain largely similar to our current approach.

REFERENCES

1. Faller, Ralf and Ohndorf, Andreas and Schlepp, Benjamin and Eberle, Sabrina (2012) Preparation, Handover, and Conduction of PRISMA Mission Operations at GSOC,63rd International Astronautical Congress, IAC2012, Naples, Italy https://elib.dlr.de/78565
2. Park, T. H., Märtens, M., Lecuyer, G., Izzo, D., D'Amico, S. SPEED+(2022): Next-Generation Dataset for Spacecraft Pose Estimation across Domain Gap,IEEE Aerospace Conference (AERO), pp. 1-15, https://doi.org/10.1109/AERO53065.2022.9843439.
3. Park, T. H., Bosse, J., D'Amico, S. (2021) Robotic Testbed for Rendezvous and Optical Navigation: Multi-Source Calibration and Machine Learning Use Cases, 2021 AAS/AIAA Astrodynamics Specialist Conference, Big Sky, Virtual, https://arxiv.org/abs/2108.05529.
4. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y. (2020), Generative adversarial networks,Communications of the ACM, *63*(11), 139-144, https://arxiv.org/abs/1406.2661.
5. Wang, J., Sun, K., Cheng, T., Jiang, B., Deng, C., Zhao, Y., ... & Xiao, B. (2020), Deep high-resolution representation learning for visual recognition,IEEE transactions on pattern analysis and machine intelligence, *43*(10), 3349-3364, https://doi.org/10.1109/tpami.2020.2983686.
6. Chen, B., Cao. J, Parra.A, Chin.T, (2019) Satellite Pose Estimation with Deep Landmark Regression and Nonlinear Pose Refinement,In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Korea, https://doi.org/10.48550/arXiv.1908.11542.
7. Shang.H, Song.B, Yang.X & Nan.X(2020), An Improved Deep Keypoint Detection Network for Space Targets Pose Estimation,Remote Sensing, 12(23),3857,https://doi.org/10.3390/rs12233857.
8. Sergeev.A & Del Balso.M(2018), Horovod: fast and easy distributed deep learning in TensorFlow,arXiv preprint arXiv:1802.05799.
9. Anthony.Q, Awan.A. A, Jain.A., Subramoni.H & Panda.D.K.D(2020),Efficient training of semantic image segmentation on summit using horovod and mvapich2-gdr,In 2020 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)(pp. 1015-1023)IEEE.
10. Rakshith.R. M, Lokur. V, Hongal. P, Janamatti. V & Chickerur.S (2022), Performance Analysis of Distributed Deep Learning using Horovod for Image Classification,In 2022 6th International Conference on Intelligent Computing and Control Systems (ICICCS)(pp.1393-1398)IEEE, https://ieeexplore.ieee.org/document/9788247.
11. Saxena.S, Sharma.S & Sharma.N (2016), Parallel Image Processing Techniques, Benefits and Limitations,Research Journal of Applied Sciences, Engineering and Technology, 12(2), 223–238, https://doi.org/10.19026/rjaset.12.2324.
12. Chen.L, Bai. X, Liang. Y & Li. K (2016), Orbital Prediction Error Propagation of Space Objects, In Orbital Data Applications for Space Objects (pp. 23–75), https://doi.org/10.1007/978-981-10-2963-9_2.
13. Prabakaran. N, Bhattacharyay. R, Joshi. A. D & Rajasekaran. P (2023), Generating Complex Animated Characters of Various Art Styles with Optimal Beauty Scores Using Deep Generative Adversarial Networks, In P. Swarnalatha & S. Prabu (Eds.), Handbook of Research on Deep Learning

Techniques for Cloud-Based Industrial IoT (pp. 236-254), IGI Global, https://doi.org/10.4018/978-1-6684-8098-4.ch014.

14. Prabakaran. N, Joshi. A. D, Bhattacharyay. R, Kannadasan. R & Anakath. A. S (2023), Generative Adversarial Networks the Future of Consumer Deep Learning: A Comprehensive Study, In P. Sivaram, S. Senthilkumar, L. Gupta, & N. Lokesh (Eds.), Perspectives on Social Welfare Applications' Optimization and Enhanced Computer Applications (pp. 181-198), IGI Global, https://doi.org/10.4018/978-1-6684-8306-0.ch012.
15. Phisannupawong. T, Kamsing. P, Torteeka. P, Channumsin. S, Sawangwit. U, Hematulin. W, Jarawan. T, Somjit. T, Yooyen. S, Delahaye. D & Boonsrimuang. P (2020), Vision-Based spacecraft pose estimation via a deep convolutional neural network for noncooperative docking operations, Aerospace, *7*(9), 126, https://doi.org/10.3390/aerospace7090126.
16. Meet Horovod: Uber's open source distributed deep learning framework for TensorFlow | Uber Blog, (2017), Uber Blog. https://www.uber.com/en-IN/blog/horovod/.
17. Gibiansky. A, Bringing HPC techniques to Deep Learning Andrew Gibiansky, https://andrew.gibiansky.com/blog/machine-learning/baidu-allreduce/.
18. LeBeane. M (2018), Optimizing communication for clusters of GPUs, https://doi.org/10.15781/t2qn5zx92.
19. Coquelin. D, Debus. C, Götz. M, Von Der Lehr. F, Kahn. J, Siggel. M & Streit. A (2022), Accelerating neural network training with Distributed Asynchronous and Selective Optimization (DASO), Journal of Big Data, 9(1), https://doi.org/10.1186/s40537-021-00556-1.
20. Patarasuk. Pitch and Xin Yuan,(2009) "Bandwidth optimal all-reduce algorithms for clusters of workstations" Journal of Parallel and Distributed Computing 69.2: 117-124, https://api.semanticscholar.org/CorpusID:7433454 .
21. Hannun, Awni et al. (2014) "Deep speech: Scaling up end-to-end speech recognition" arXiv preprint arXiv:1412.5567.
22. Kurth. T, Smorkalov. M, Mendygral. P, Sridharan. S & Mathuriya. A (2019), TensorFlow at Scale: Performance and productivity analysis of distributed training with Horovod, MLSL and Cray PE ML, Concurrency and Computation: Practice and Experience, 31(16), 4989, https://doi.org/10.1002/cpe.
23. Wu. X, Taylor. V, Wozniak. J. M, Stevens. R, Brettin. T & Xia. F (2018), Performance, power and scalability analysis of the horovod implementation of the candle nt3 benchmark on the cray xc40 theta, In SC18 Workshop on Python for High-Performance and Scientific Computing, Dallas, USA.
24. Kunde. S, Pandit. A & Singhal. R (2020), Benchmarking performance of RaySGD and Horovod for big data applications, In 2020 IEEE International Conference on Big Data (Big Data) (pp. 2757-2762), http://dx.doi.org/10.1109/BigData50022.2020.9378470.
25. Awan. A. A, Bédorf. J, Chu. C. H, Subramoni. H & Panda. D. K (2019), Scalable distributed DNN training using Tensorflow and cuda-aware Mpi: Characterization, designs and performance evaluation, In 2019 19th IEEE/ACM International Symposium on Cluster, Cloud and GRID computing (CCGRID) (pp. 498-507), https://api.semanticscholar.org/CorpusID:53079193.
26. Sørlie. L, (2022), Preparing the CERN machine-learned particle-flow model for Exascale using Horovod: Experience and performance studies on the Flatiron and Jülich supercomputers (Doctoral dissertation, Norwegian University of Science and Technology (NTNU)(NO)).
27. Wang. Y, Dong. D, Xu. Y, Ouyang. S & Liao. X, (2021), FastHorovod: Expediting Parallel Message-Passing Schedule for Distributed DNN Training, IEEE Symposium on Computers and Communications (ISCC) (pp. 1-7).
28. Zehua Wang, Suyin Zhou, Ping Yin, Aijin Xu, Junhua Ye(2023), GANPose: Pose estimation of grouped pigs using a generative adversarial network, Computers and Electronics in Agriculture, Vol 212, https://doi.org/10.1016/j.compag.2023.108119.
29. Wu et al., (2023)"Video-Based Fall Detection Using Human Pose and Constrained Generative Adversarial Network," in IEEE Transactions on Circuits and Systems for Video Technology, doi: 10.1109/TCSVT.2023.3303258.

Prabakaran.N is currently working as an Associate Professor in the School of Computer Science and Engineering, Vellore Institute of Technology (VIT), Vellore, Tamil Nadu and India under the Department of Data Science. He received his BE and M.E in Computer Science and Engineering from Anna University, Chennai, India in 2009 and 2011 respectively. He has completed his Ph.D.from Computer Science and Engineering at Vellore Institute of Technology (VIT), Vellore, India in 2017. His current research interest includes Massive mining Data, Machine learning Algorithms, Deep learning models and Pervasive Computing.

Corresponding Author E-mail: dhoni.praba@gmail.com

Aditya Deepak Joshi is a Postgraduate student of Department of Computer Science and Engineering from New York university, USA. He received his B.Tech degree from Vellore Institute of Technology (VIT), Vellore, India. He is passionate about exploring the applications of Machine Learning

algorithms, cluster analysis, and Deep Learning techniques in various domains such as natural language processing, computer vision, and social network analysis.

E-mail: aj4128@nyu.edu

Rajasekaran.P received his B.E. degree in Electronics and Communication Engineering from Francis Xavier Engineering College, Tirunelveli, Tamil Nādu, India in 2009, M.E. degree in Pervasive Computing Technologies from Anna University, Tiruchirappalli, Tamil Nādu, India in 2011 and Ph.D. degree in Information and Communication Engineering from Anna University, Chennai, Tamil Nādu, India in 2023.His areas of interest are network security, pervasive computing and dynamic power management.

E-mail: p.rajasekaran537@gmail.com