Volume: 12 Issue: V Month of publication: May 2024

DOI: https://doi.org/10.22214/ijraset.2024.62319

Conversion of an Entire Image into a 3D Scene with Object Detection and Single 3D Models

Ashwinee Barbadekar[1], Kshitij Bhure[2], Shreyash Bele[3], Bhagyesh Bhoge[4]
Department of Electronics and Telecommunication Engineering, Vishwakarma Institute of Technology, Pune

Abstract: The paper presents a framework for the extraction of objects from 2D images using object detection techniques in open CV and then generating 3D models of the segmented objects. To achieve precise object detection, the suggested methodology blends deep learning and computer vision methods. The 3D models that are produced have fine-grained geometric and textural features, which improves the object reconstruction's accuracy. The usefulness and robustness of our technique in various settings are demonstrated by experimental findings on a variety of datasets. For use in robotics, virtual reality, and the preservation of cultural heritage, where precise 3D models of actual objects are essential, this research shows great promise.

Keywords: Image Segmentation, Object Detection, Transformers, One-2-3-45, 3D models

I. INTRODUCTION

A revolutionary era in the study of image analysis and object understanding has recently been ushered by the quick development of computer vision and deep learning. These innovative technologies have created new opportunities for use in a variety of fields, from robotics and augmented reality to the preservation of cultural heritage and object recognition. The ability to extract objects from two-dimensional (2D) images and then translate them into intricate and precise three-dimensional (3D) models is a key challenge in this environment. Researchers and business professionals alike have been fascinated by this capability because it has the potential to completely change how we interact with the digital and physical worlds.[1]

The significance of this project lies in its ambitious goal to tackle this multifaceted challenge head-on. We aim to provide a comprehensive solution by presenting a well-structured pipeline that incorporates object detection and 3D reconstruction methodologies. Object detection, a fundamental step in the process, involves the precise identification and delineation of objects within a 2D image, allowing for a granular understanding of their spatial boundaries and relationships. This segmented information then becomes the foundation for the subsequent generation of detailed 3D models.[3][4]

This project embarks on an ambitious project to build a robust image classification pipeline with the ability to accurately detect and describe many features in a given two-dimensional image The main goal is to develop sophisticated systems which can accurately categorize different components, and lay the foundation for subsequent phases. Based on segmented object data, the objective of the project is to produce more realistic 3D models for each identified object, extending the application to more detailed three-dimensional models The ultimate goal is to demonstrate how a proposed method of the technique is efficient and applicable to the spectrum of real-world scenarios and data sets To achieve the project is not only to provide practical solutions for applications that require the creation of complex 3D objects not only but will contribute significantly to ongoing computer vision techniques.

II. LITERATURE REVIEW

Deepu R, Murali S et al. -Perceiving 3D scenes is a challenge for computers as it requires obtaining 3D shape information from planar images, known as 3D reconstruction. This is done by identifying planes and constructing a representation of the scene. The captured image is perspectively distorted, which is corrected using corner point estimation. View metrology helps determine the true dimensions. A 3D model is then constructed in VRML based on these dimensions, and texture maps are applied to polygon surfaces. VRML supports walkthroughs for generating different views [1].

Era Xian-Feng Han*, Hamid Laga*e et al. - This paper presents a comprehensive review of recent developments in image-based 3D reconstruction using convolutional neural networks (CNNs). It focuses on deep learning techniques to estimate the 3D shape of common objects from one or more RGB images. The literature is organized based on shape representations, network architectures, and training mechanisms. We also look at certain classes of objects, such as human body shapes and faces. The article provides an analysis of the results, identifies open issues, and suggests future research directions.[2]

Victoria M Baretto et al.- To address the scarcity of 3D content compared to 2D, various methods have been proposed for 2D-to-3D image conversion. Human-operated methods have been successful but time-consuming and costly.

Automatic methods relying on deterministic 3D scene models have not achieved comparable quality due to practical violations of assumptions. Two types of methods have been developed: one based on learning a point mapping from local image attributes, and the other based on estimating the depth map of an image from a repository of 3D images using nearest-neighbour regression. These methods demonstrate effectiveness and computational efficiency but also have their drawbacks and benefits[3]

Avi Kanchan1, Tanya Mathur2 et al. **-** This paper explores various methods for converting 2D images to 3D. Due to the dominance of 2D content, there is a pressing need for such conversion. Methods are categorized as automatic or semi-automatic, depending on the involvement of human operators. Depth calculation is crucial for 3D images, and researchers have proposed various approaches to address this challenge. The paper focuses on an algorithm that utilizes learning from examples and monocular depth cues, providing an overview and evaluation within the field of conversion algorithms. The aim is to contribute to the development of novel depth cues and improved algorithms leveraging combined depth cues.[4]

Xiangrong Zhou*a, Kazuma Yamada a, Takuya Kojima a, Ryosuke Takayama a, Song Wang b, Xinxin Zhou c, Takeshi Hara a, and Hiroshi Fujita a et al:- This study evaluates and compares the performance of state-of-the-art deep learning techniques for automatic detection and segmentation of multiple life regions in 3D CT images. Deep learning approaches are applied to the challenging task of CT image segmentation in medical image analysis. The study compares two different deep learning approaches using 2D and 3D deep convolutional neural networks (CNN) with and without a preprocessing step. A traditional state-of-the-art segmentation method without deep learning is also evaluated. A dataset consisting of 240 CT images of different body parts is used, and up to 17 life regions are segmented and compared with human markers. Experimental results show that deep learning approaches achieve higher accuracy and robustness compared to the traditional method using probabilistic atlas and graph cutting techniques. The study highlights the efficiency and usefulness of deep learning for multi-organ segmentation in 3D CT images.[5]

The paper "GET3D: Generative 3D Modeling for Textured Meshes with Complex Topology" responds to the demand for scalable, high-quality 3D content creation in fields like gaming and architecture. Manual 3D asset creation is labour-intensive, and creating diverse 3D models often requires significant time and expertise. The authors introduce GET3D, a groundbreaking generative model that can directly produce detailed 3D meshes with intricate geometry and high-fidelity textures. GET3D satisfies three key requirements for practical 3D generative models: the capacity to create complex shapes with arbitrary topology, the generation of textured 3D meshes, and the utilization of 2D images for training, which are more widely available than 3D data. Comparative analysis demonstrates that existing methods fall short of meeting all these criteria, highlighting the novelty of GET3D. Leveraging advances in differentiable surface modelling, rendering, and 2D Generative Adversarial Networks, GET3D achieves state-of-the-art performance across a range of 3D categories, such as vehicles, furniture, animals, human characters, and buildings. Additionally, its adaptability for various tasks, including material generation and text-guided shape creation, positions GET3D as a versatile and pivotal tool for the 3D content creation landscape.[6]

The paper presents a new technique for single-image 3D reconstruction called One-2-3-45. One-2-3-45 combines a view-conditioned 2D diffusion model, Zero123, with a cost-volume-based 3D reconstruction technique to address the issues of long optimisation times, inconsistent 3D geometry, and inadequate geometry in previous methods. This allows for the creation of high-quality 360-degree textured meshes from a single image in a single feed-forward pass. The novel aspect of the approach is that it does not use per-shape optimisation, which allows for a dramatic 45-second reconstruction time reduction. It also enhances geometry and 3D consistency. The approach stays close to the input image by introducing training strategies and utilizing multi-view predictions. Testing on both synthetic and real data shows that One-2-3-45 is superior in terms of efficiency and quality, which represents a breakthrough in single-image 3D reconstruction.[7]

The "Zero-1-to-3" framework is presented in the paper, which uses a single RGB image to modify an object's camera viewpoint by utilizing large-scale diffusion models. This new method of view synthesis makes use of the geometric priors that these models have acquired from their extensive pre-training data. The approach demonstrates strong zero-shot generalisation capabilities to out-of-distribution datasets and even in-the-wild images, such as impressionist paintings, and learns controls for relative camera viewpoint transformations from a synthetic dataset. It can also be used for single-image 3D reconstruction. Large diffusion models that were only trained on 2D images have been shown to acquire rich 3D priors about the visual world, which makes them useful for zero-shot 3D object reconstruction as well as novel view synthesis. [8]

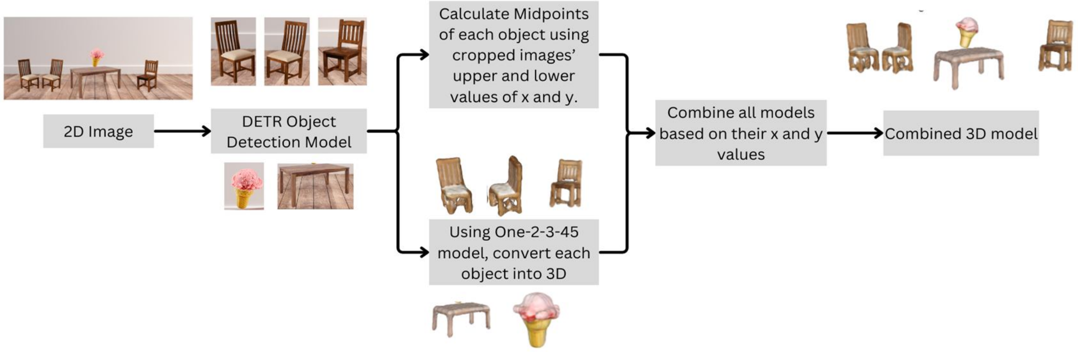

2D Image
DETR Object Detection Model
Calculate Midpoints of each object using cropped images' upper and lower values of x and y.
Using One-2-3-45 model, convert each object into 3D
Combine all models based on their x and y values
Combined 3D model

Fig 1: 2D to 3D conversion Algorithm

III. METHODOLOGY

Fig 1 shows step-by-step process of the model. The algorithm shows that the first step is to use object detection and detect various objects in the image. However, the previous approach was to use an image segmentation model and segment different parts in the image and convert the segmented parts into 3d models. The issue we encountered is inconsistent results on most of the images being used for the input. We used popular image segmentation models from HuggingSpace[9].

Image Segmentation Models:

To develop a state-of-the-art image-to-3D-model system, we thoroughly investigated three different segmentation models: CLIPSeg, SegFormer (b0-size), and MaskFormer. With their distinct features and promises, each of these models has the potential to completely transform the way that objects are extracted from uploaded images and seamlessly transformed into detailed 3D models.

An innovative approach to picture segmentation was taken by CLIPSeg, an extension of the CLIP model, which allowed segmentation maps to be generated based on random prompts during test time. This was a change from conventional techniques that called for predetermined sets of object classes to be trained. The model supported tasks like referencing expression segmentation, zero-shot segmentation, and one-shot segmentation by utilizing a transformer-based decoder to provide dense prediction. Although CLIPSeg's design is quite versatile, the segmented object results were not up to par for our project's particular needs. The expected accuracy and precision in segmentation outcomes did not materialize from the promise of a single model trained for several segmentation tasks.[10]

We used SegFormer (b0-size), a semantic segmentation system designed to combine lightweight multilayer perceptron (MLP) decoders with Transformers to overcome these constraints. To mitigate potential performance decrease when testing resolutions differed from training resolutions, this model avoided the use of positional encoding and instead included a hierarchically structured Transformer encoder that produced multiscale features. A unique MLP decoder integrated into the lightweight architecture combined local and global attention for robust representation by aggregating data from many layers. A range of models, from SegFormer-B0 to SegFormer-B5, were used to illustrate scalability; the latter produced some outstanding results, including an 84.0% mIoU on the Cityscapes validation set.[11]

In an attempt to find the best segmentation solution, MaskFormer, the third model, was presented. Using the same model, loss function, and training process for both semantic and instance-level segmentation tasks, MaskFormer tackled semantic segmentation as a mask classification task. Even though MaskFormer was successful in streamlining segmentation tasks and surpassing per-pixel classification baselines in scenarios with a high number of classes, it still failed to meet the project's specific requirements.[12]

It was clear from extensive testing that none of the three segmentation models produced the required segmentation results for the image-to-3D-model conversion. This led to a change in strategy, with the investigation now focusing on object detection rather than segmentation. This change in strategy recognised the shortcomings of the segmentation-focused approaches and sought to investigate alternative approaches to accomplishing the project's goals.

A. Object Detection

The computer vision technique, object detection, locate objects in images or videos. The object detection algorithm uses machine learning or deep learning to produce meaningful results.

For our project, object detection is the first need to identify different objects that are present in the image, separate them and save them as separate images. By directly predicting sets of objects without the need for heuristics or post-processing procedures, DETR transforms object detection. DETR ensures unique and accurate matches between ground truth objects and predicted objects through a combination of transformer-based architectures with parallel decoding and a bipartite matching loss derived from the Hungarian algorithm. By predicting absolute box positions relative to the input image, this methodology streamlines the detection process and does away with the need for conventional anchor- or proposal-based detection methods. DETR represents a major shift towards direct set predictions in the field by providing a new and effective method of performing end-to-end object detection by utilizing global computations and memory within transformers.[9]

Along with saving the different objects, the x and y centre coordinates of each object are calculated simultaneously.

```
xmin = image['xmin']
ymin = image['ymin']
xmax = image['xmax']
ymax = image['ymax']
center_x = (xmin + xmax) / 2
center_y = (ymin + ymax) / 2
```

The above formula is used for calculation of x and y centre of each object.

B. Create 3D Models

For creating 3D models from an image, we decided to use a Generative AI model. There have been recent developments in the field of converting a 3D model from an image. But there’s a catch. These AI generators convert only a single object image into a 3D model and are incompatible with creating a 3D model of a big image. We explored 2 Generative AI models, One-2-3-45 and Nvidia Get3D. The Get3D model didn’t have an API that could provide direct results and required coding the entire program which would drain all the GPU resources available. One-2-3-45 provided an API on the HuggingFace platform.[18] With the help of the API, the individual images extracted from the object detection module are converted into 3D models. The model is trained on previous work - Zero-1-2-3. This model predicts the views of a single input image from various angles. Based on this, One-2-3-45 creates meshes of different sides of the predicted views. This model works by altering 2 parameters namely, diffusion guidance scale and a number of diffusion inference steps determining diversity from the original image and the number of diffusion steps applied for generation respectively. This step is manual as the user has to adjust these parameters for different kinds of images according to his satisfaction with the 3D model being generated. The created 3D models are exported in the format of a .glb file, which is a 3D file extension.

C. Combining the 3D Models

This was the most difficult process in the entire project. Combining 3D model meshes along with their texture is a very difficult task. We had to try a lot of methods to achieve good results. There are various Python libraries for working with 3D models. We used libraries like pygltflib, vtk, trimesh, PyWavefront, pyrenderer, assimp and pyglet. While all these libraries provided the function to combine various 3D models together, not all were useful. Most of the libraries had compatibility issues with the generated 3D models due to several reasons like texture, mesh type etc while others failed to stitch the models together because of importing issues. There are various 3D file extensions like .fbx, .obj, .stl, .glb etc. For our first attempt, we used the .obj extension. We converted the generated 3D models into .obj files and tried to stitch those 3D models together successfully using trimesh library. The acquired output model was satisfactory but had one major flaw, the textures of the 3D models were not imported. This can be seen in Fig 2.

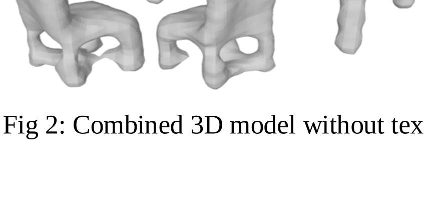


Fig 2: Combined 3D model without textures

This problem arises because .obj and .stl files save textures separately in another file while .glb file embeds the textures along with its mesh. So, we decided to use the .glb file format for stitching all the 3D models together. The 3D models were successfully stitched together along with their textures using the trimesh library. In the object detection module, we saved the x and y centre coordinates of the individual objects. These coordinates are used while stitching the 3D models. The obtained output can be seen in Fig 4.

Test Case	Running Time
Test Case 1(3 Models)	4 min 40 sec
Test Case 2 (4 Models)	3 min 29 sec
Test Case 3 (4 Models)	5 min 15 sec

Fig 3. Running time of different test cases

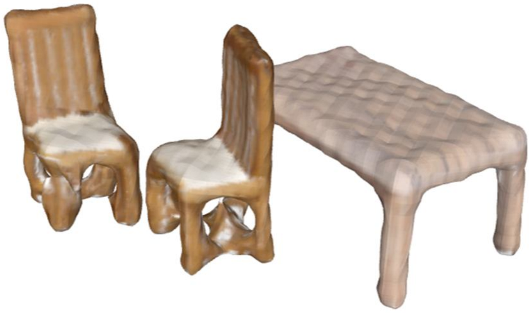


Fig 4: Combined 3D model with textures

IV. RESULTS

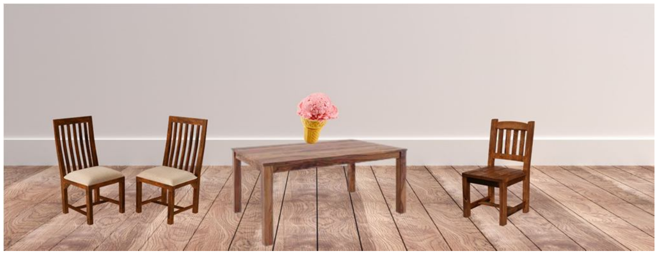


Fig 5: Input image with multiple objects

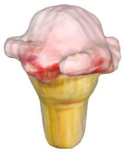


Fig 6: 3D model of Ice cream

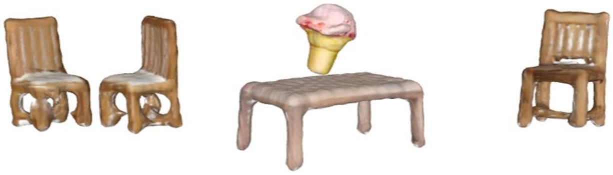


Fig 7: 3D Output gained from the input image

Fig 5 shows that multiple objects are detected individually. Fig 6 shows the 3D model of ice cream that is generated from the detected image. In Fig 7, you can see the combined 3D model of all the objects in the input image. The position and rotation are also retained from the original image.
In Fig 3, a table is shown denoting the running time of different test cases. Through various test cases, we figured out that the more the 3D models are to be generated, the more time is taken. After running the entire model on test cases with the same number of 3D models which was 4, the average running time was about 4 minutes 36 seconds.

V. LIMITATIONS

This project has several shortcomings. Its high computational resource requirements hamper large-scale or real-time applications. The lack of high GPU hardware resulted in low-quality 3D models which need refining before practical use. Results are heavily influenced by the quality of the input images, and segmentation mistakes of different objects can cause errors in the 3D models. It can be difficult to capture complex scenes with lots of objects, occlusions, and reflections. An image with a clean and simple setting seems to give a good result whereas with images with lots of details and objects, the object detection model fails to detect every single detail. It's still difficult to accurately depict materials, textures, and reflectance qualities. It is not easy to generalize models to new object classes or scenes. Privacy and possible misuse are significant ethical concerns. Practical challenges also include handling large 3D models, accessibility, and user-friendliness.

VI. CONCLUSION

In summary, this study introduces a flexible framework for feature extraction from 2D models, followed by complex 3D models Our method uses imaging computer vision techniques established together with state-of-the-art deep learning-based object detection. Ensuring segmentation and optimization of 3D models the resulting 3D models exhibit finer geometric and textural details, increasing the overall accuracy of object reconstruction. The robustness and flexibility of our method can be seen in the test results on different data sets and environments. These findings have broader implications in a variety of fields, including robotics, virtual reality, and cultural heritage preservation. In robotics, our approach can significantly improve the tasks of object manipulation and detection. Additionally, it enables the creation of more authentic, immersive experiences in virtual reality. Additionally, our approach is a valuable tool for the restoration of historic objects and sites, contributing to the preservation of cultural heritage. For professional uses, the obtained 3D model can be modified in various 3D modelling software like Maya and Blender3D and will help in adding more details.

VII. FUTURE SCOPE

Future plans for the image-to-3D model project include investigating sophisticated segmentation models and possibly combining hybrid techniques or object detection components for increased accuracy. Data augmentation techniques can be used to diversify the training dataset, and fine-tuning and transfer learning approaches can be used to improve the adaptability of the models. The system's performance could be improved by incorporating user feedback mechanisms to improve semantic understanding and segmentation results.
Crucial elements include investigating 3D reconstruction methods, optimizing for real-time processing, and enhancing the user interface for improved interaction. Further consideration of modalities, like depth data, could improve input data even more. Together, these approaches seek to overcome existing shortcomings and guarantee an image-to-3D-model system that is more precise, flexible, and easy to use while satisfying a wide range of application requirements.

VIII. ACKNOWLEDGEMENT

Our sincere gratitude extends to Dr Ashwini Barbadekar from the Vishwakarma Institute of Technology in Pune for his invaluable guidance and support throughout the course of this study. His expert advice and assistance were instrumental not only in the development of our research but also in navigating the publication process of this article. Dr. Barbadekar's commitment to our project's success was evident in his recommendation to maintain a structured approach, prompting us to provide weekly updates for effective progress tracking. His insightful suggestions, shared during the mid-assessment review, have significantly contributed to the refinement and enhancement of our project. We are truly thankful for his mentorship, which has played a pivotal role in the advancement of our research endeavours.

REFERENCES

[1] "3D Reconstruction from Single 2D Image”, Deepu R, Murali S, Department of Computer Science & Engineering, Maharaja Research Foundation, Maharaja Institute of Technology Mysore, India.

[2] Image-based 3D Object Reconstruction: State-of-the-Art and Trends in the Deep Learning Era Xian-Feng Han*, Hamid Laga*, Mohammed Bennamoun Senior Member, IEEE." by Kazuo Ohzeki, Max Geigis,Stefan Alexander Schneider.

[3] Automatic Learning based 2D-to-3D Image Conversion Victoria M Baretto Dept. of Computer Science and Engineering Alva’s Institute of Engineering and Technology (AIET).

[4] Automatic Learning based 2D-to-3D Image Conversion Victoria M BarettoRECENT TRENDS IN 2D TO 3D IMAGE CONVERSION: Algorithms at a glance Avi Kanchan1, Tanya Mathur2 Dept. of Computer Science and Engineering Alva’s Institute of Engineering and Technology (AIET)

[5] Performance evaluation of 2D and 3D deep learning approaches for automatic segmentation of multiple organs on CT images Xiangrong Zhou*a , Kazuma Yamada a , Takuya Kojima a , Ryosuke Takayama a , Song Wang b , Xinxin Zhou c , Takeshi Hara a , and Hiroshi Fujita a

[6] GET3D: A Generative Model of High Quality 3D Textured Shapes Learned from Images Jun Gao1,2,3 Tianchang Shen1,2,3 Zian Wang1,2,3 Wenzheng Chen1,2,3 Kangxue Yin1 Daiqing Li1 Or Litany1 Zan Gojcic1 Sanja Fidler1,2,3 NVIDIA1 University of Toronto2 Vector Institute3 {jung, frshen, zianw, wenzchen, kangxuey, daiqingl, olitany, zgojcic, sfidler}@nvidia.com

[7] One-2-3-45: Any Single Image to 3D Mesh in 45 Seconds without Per-Shape Optimization Minghua Liu1∗ Chao Xu2∗ Haian Jin3,4∗ Linghao Chen1,4∗ Mukund Varma T5 Zexiang Xu6 Hao Su1 1 UC San Diego 2 UCLA 3 Cornell University 4 Zhejiang University 5 IIT Madras 6 Adobe

[8] Zero-1-to-3: Zero-shot One Image to 3D Object Ruoshi Liu1 Rundi Wu1 Basile Van Hoorick1 Pavel Tokmakov2 Sergey Zakharov2 Carl Vondrick1 1 Columbia University 2 Toyota Research Institute

[9] Carion, Nicolas, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. "End-to-end object detection with transformers." In *European conference on computer vision*, pp. 213-229. Cham: Springer International Publishing, 2020.

[10] Lüddecke, Timo, and Alexander Ecker. "Image segmentation using text and image prompts." In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7086-7096. 2022.

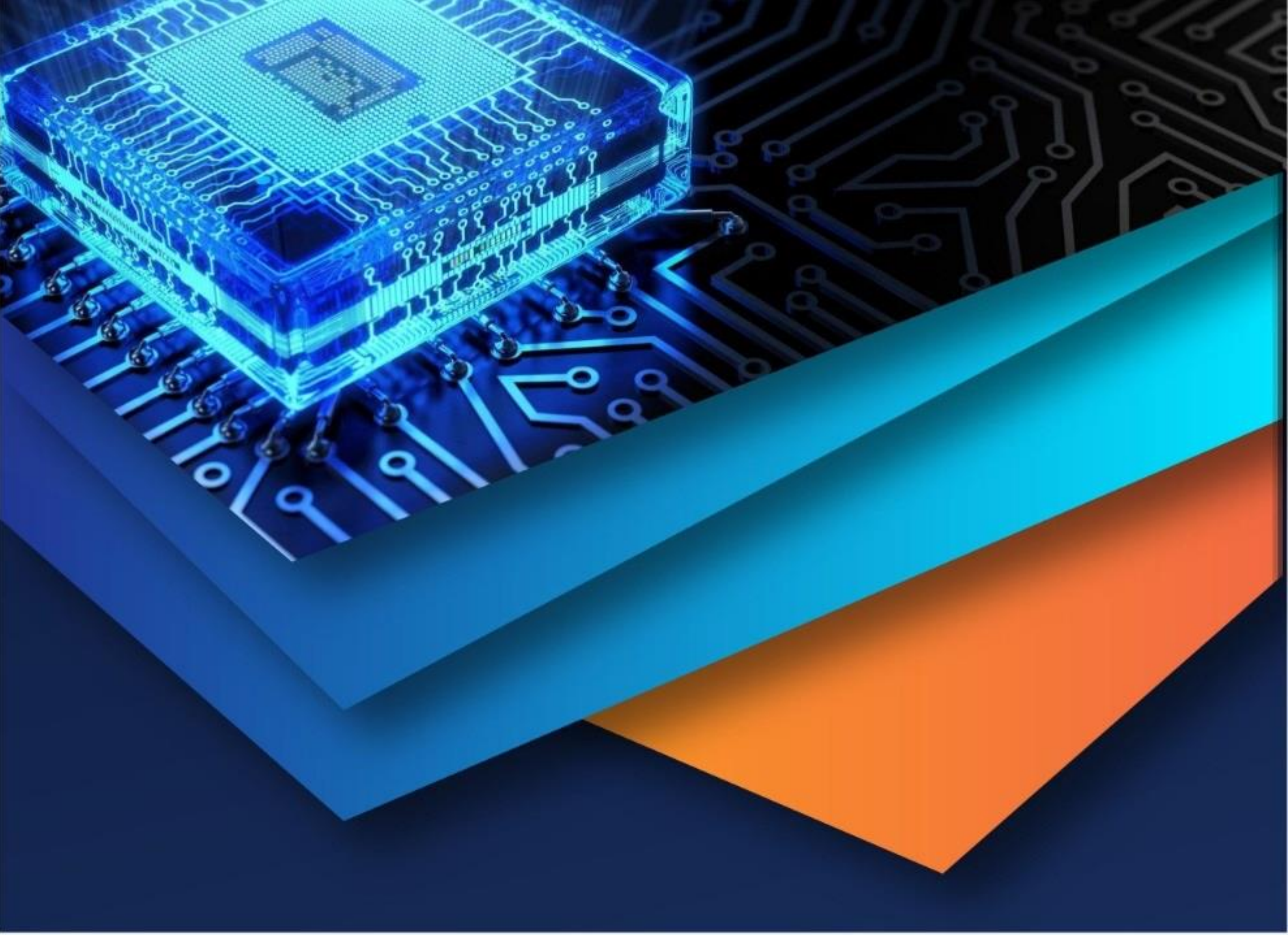
[11] Xie, Enze, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M. Alvarez, and Ping Luo. "SegFormer: Simple and efficient design for semantic segmentation with transformers." *Advances in Neural Information Processing Systems* 34 (2021): 12077-12090.

[12] Cheng, Bowen, Alex Schwing, and Alexander Kirillov. "Per-pixel classification is not all you need for semantic segmentation." *Advances in Neural Information Processing Systems* 34 (2021): 17864-17875.

[13] Cheng, Bowen, Alex Schwing, and Alexander Kirillov. "Per-pixel classification is not all you need for semantic segmentation." Advances in Neural Information Processing Systems 34 (2021): 17864-17875.

[14] Lüddecke, Timo, and Alexander Ecker. "Image segmentation using text and image prompts." In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7086-7096. 2022.

[15] Xie, Enze, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M. Alvarez, and Ping Luo. "SegFormer: Simple and efficient design for semantic segmentation with transformers." Advances in Neural Information Processing Systems 34 (2021): 12077-12090.



10.22214/IJRASET

INDEX COPERNICUS
45.98

IMPACT FACTOR:
7.129

TOGETHER WE REACH THE GOAL
IMPACT FACTOR:
7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY