Visvesvaraya Technological University, Belagavi

A project report on

'ARTIFICIAL INTELLIGENCE BASED CAMERA CALIBRATION'

Submitted in partial fulfillment of the requirement for the award of the degree

Bachelor of Engineering

in

Electronics and Communication Engineering

Submitted by

Niveditha (4MH18EC074)

Bhoomika D. (4MH20EC010)

Harshitha S. (4MH20EC030)

Under the guidance of

Mrs. Suma R.

Asst. Professor

Dept. of E&CE

MIT Mysore

Department of Electronics & Communication Engineering

Maharaja Institute of Technology Mysore

2023-2024

Visvesvaraya Technological University, Belagavi

Maharaja Institute of Technology Mysore

Department of Electronics and Communication Engineering

Certificate

Certified that the project work entitled **'Artificial Intelligence based Camera Calibration'** is a bonafide work carried out by **Niveditha(4MH18EC074), Bhoomika D.(4MH20EC010), Harshitha S.(4MH20EC030),** and this project report is submitted in partial fulfilment for the award of **Bachelor of Engineering** in **Electronics and Communication Engineering** of **Visvesvaraya Technological University, Belagavi** during the year 2023-2024. It is certified that all corrections/suggestions indicated for internal assessment have been incorporated in the project report and have been approved as it satisfy the academic requirements in respect of project work prescribed for the said Degree.

Signature of the Guide
Mrs. Suma R.
Asst. Professor
Dept. of E&CE
MIT Mysore

Signature of the HoD
Dr. R.V. Parimala
Prof. and Head
Dept. of E&CE
MIT Mysore

Signature of the Principal
Dr. B.G. Naresh Kumar
Principal
MIT Mysore

External Viva

Name of the examiners	Signature with date
1.	
2.	

Acknowledgement

We express our gratitude and respect to **Dr. B.G. Naresh Kumar**, Principal, MIT Mysore, for granting permission to carry out the project work.

We express our heartfelt gratitude to **Dr. R.V. Parimala,** Professor and Head, Department of Electronics and Communication Engineering, MIT Mysore for being supportive throughout the project work.

We express our deep sense of gratitude to our guide **Mrs. Suma R.,** Assistant Professor Department of Electronics and Communication Engineering, MIT Mysore for her valuable suggestions and inspiration, motivating guidance and who have been the driving force behind this work and who have constantly dedicated her precious time with timely suggestions and ideas to successfully carry out the project work.

Last but not the least, we would like to thank our **Parents and Friends** for their moral support during critical phases of our work.

Niveditha (4MH18EC074)

Bhoomika D. (4MH20EC010)

Harshitha S. (4MH20EC030)

Abstract

Camera calibration technique plays a vital role in three-dimensional computer vision systems. The aim of this technique is to calibrate the camera in order to collect more precise three-dimensional data from the images to utilize for robot navigation, three-dimensional reconstruction, biomedical, virtual reality and visual surveillance. In camera calibration one of the major-issue is to search out the set of image parameters describing the mapping between 3D images reference coordinates and 2D images reference coordinates. Currently, MATLAB toolbox and Open CV are the most popular tools used by the researchers for camera calibration. We utilize the concept of deep learning to recognize the chessboard corners. Our presented technique is a convolutional neural network (CNN) trained on a huge number of chessboard images. The network is trained on different datasets: noisy and with high lens malformation images. The proposed scheme is more accurate than the conventional MATLAB algorithm technique. The presented technique is more accurate against the different sort of ruination present in the training set. Results reaffirmed the correctness and effectiveness of our proposed CNN technique.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

CNN	Convolutional Neural Network
3D	Three-Dimensional
2D	Two-Dimensional
1D	One-Dimensional
CV	Computer Vision

Chapter 1

PREAMBLE

1.1 INTRODUCTION

Camera calibration is the process of determining the parameters that describe how a camera captures the 3D world and maps it to a 2D image. The primary goals of camera calibration are to understand and quantify how a camera system behaves and to correct for any distortions or inaccuracies in the imaging process. These goals serve several important purposes in various applications of computer vision, robotics, and computer graphics. The project aims to implement artificial intelligence techniques for camera calibration. Through machine learning algorithms, the system will analyze image data to determine intrinsic and extrinsic camera parameters. The ultimate-goal is to achieve precise spatial mapping, enhancing applications like object recognition and 3D reconstruction by ensuring accurate geometric relationships between the camera and the captured environment.

Camera calibration improve the accuracy and reliability of imaging systems by understanding and correcting for distortions and errors introduced by cameras. It supports scientific research, enhances industrial efficiency, and contributes to safety and security applications, benefiting society and industry. There are multiple techniques of camera calibration. These techniques are categorized into three classes: traditional calibration technique, self-calibration and active vision. Traditional techniques are portrayal as direct linear transformation (DLT), two-step technique and dual-plane technique. If, there is no calibration object is present then it brings a type of calibration which is called as self-calibration. Currently, Active Vision camera self-calibration techniques are used in taking account the non-linear distortion of camera. There are numerous types of calibration techniques, like calibration method based on three-dimensional calibration, camera calibration using OpenCV. The camera calibration module of OpenCV offers a good interface for user, and supports Windows and Linux platform.

Camera calibration is a necessary step in 3D computer vision in order to extract metric information from 2D images. It has been studied extensively in computer vision and photogrammetry, and even recently new techniques have been proposed. In this chapter, unknown scene points can be seen in the environment (self-calibration). The focus is on presenting these techniques within a consistent framework.

3D reference object-based calibration - Camera calibration is performed by observing a calibration object whose geometry in 3-D space is known with very good precision. Calibration can be done very efficiently. The calibration object usually consists of two or three planes orthogonal to each other. Sometimes, a plane undergoing a precisely known translation is also used, which equivalently provides 3D reference points. This approach requires an expensive calibration apparatus and an elaborate step.

2D plane-based calibration - Techniques in this category requires to observe a planar pattern shown at a few different orientations. Different from Tsai's technique, the knowledge of the plane motion is not necessary. Because almost anyone can make such a calibration pattern by him/her-self, the setup is easier for camera calibration.

1D line-based calibration - Calibration objects used in this category are composed of a set of collinear points. As will be shown, a camera can be calibrated by observing a moving line around a fixed point, such as a string of balls hanging from the ceiling.

Self-calibration - Techniques in this category do not use any calibration object, and can be considered as 0D approach because only image point correspondences are required. Just by moving a camera in a static scene, the rigidity of the scene provides in general two constraints on the cameras' internal parameters from one camera displacement by using image information alone. Therefore, if images are taken by the same camera with fixed internal parameters, correspondences between three images are sufficient to recover both the internal and external parameters which allow us to reconstruct 3-D structure up to a similarity. Although no calibration objects are necessary, a large number of parameters need to be estimated, resulting in a much harder mathematical problem.

1.2 PROBLEM STATEMENT

Calibration process involves determination of camera matrix which aims to determine depth information from two-dimensional images and-also in reconstruction of three-dimensional scenes. Camera Calibration helps to collect more precise three-dimensional data from the images to utilize for robot navigation, three-dimensional reconstruction, biomedical, virtual-reality and visual surveillance. During this project, the primary challenge lies in accurately extracting intrinsic and extrinsic camera parameters. This involves training the AI model to recognize and understand the distortions present in images due to the camera's characteristics and position.

1.3 LITERATURE SURVEY

This section describes about the work done on camera calibration using different technologies which were designed by other researchers.

“ARTIFICIAL INTELIGENCE BASED CAMERA CALIBRATION” Syed Navid Raza, Suk Gyu Lee, Hafiz Raza ur Rehman, Gyu Sang Choi. 2019 IEEE/CVF International Conference on Computer Vision (ICCV) [1].

This paper explores the application of convolutional neural networks (CNNs) to address the challenge of camera calibration. The CNN system is trained using real-world images, making it adaptable to a variety of camera types. One of the significant advantages of this method is its robustness in detecting chessboard corners even under severe noise conditions. The technique exhibits consistent performance, maintaining accuracy even in extremely noisy environments. However, the method does require a high-precision calibration system, which can be costly. The paper references Zhang's calibration technique, which relies on a 2-D pattern and simplifies the process by requiring multiple shots of the same calibration pattern from different angles. This method efficiently calculates both intrinsic and extrinsic camera parameters without needing to precisely identify the position and movement of the planar pattern. Additionally, the CNN-based approach has been fine-tuned to refine all training data, effectively reducing the calibration error to zero. The paper highlights the widespread use of MATLAB toolbox and OpenCV in camera calibration research, noting their popularity among researchers. The authors also discuss how different calibration errors can be mitigated and minimized using their proposed technique. Looking ahead, they plan to apply this advanced calibration method in the field of autonomous multi-robot systems, aiming to enhance the accuracy and reliability of robot navigation and interaction in complex environments.

“DEEP SINGLE IMAGE CAMERA CALIBRATION WITH RADIAL DISTORTION” Manuel Lopez-Antequera, Roger Mari, Pau Gargallo, Yubin Kuang, Javier Gonzalez-Jimenez, Gloria Haro [2].

This paper delves into a convolutional neural network (CNN)-based approach for predicting both extrinsic and intrinsic camera parameters from a single image. By leveraging proxy variables visible within the image, this method fine-tunes independent regressors to accurately determine parameters such as tilt, roll, focal length, and radial distortion coefficients (k1 and k2). The innovative training process utilizes synthesized data from a comprehensive and varied panorama dataset, ensuring the network can generalize effectively across a wide range of scenarios. A key aspect of this method is its novel parameterization strategy, which incorporates easily observable features like the horizon line and the vertical field of view to enhance the accuracy of the predictions. This approach focuses on predicting specific camera parameters rather than attempting a full recovery of the rotation matrix, which can be more complex and computationally intensive. By prioritizing tilt, roll, focal length, and radial distortion parameters, the methodology aims to simplify the calibration process while maintaining high precision. For implementation, the study likely employs advanced deep learning frameworks such as TensorFlow or PyTorch, which are well-suited for developing and training CNNs. The proposed learning-based approach for joint extrinsic and intrinsic camera parameter prediction effectively addresses the challenges of radial distortion, showcasing notable improvements over traditional geometric methods. The paper not only presents a robust framework for camera calibration but also opens the door for future research. Potential areas for further exploration include distortion calibration techniques and applications on a larger scale, such as in autonomous systems or large-scale image processing tasks. The results indicate that this CNN-based approach can significantly enhance the accuracy and efficiency of camera calibration, offering a promising alternative to conventional methods.

"DEEP LEARNING FOR CAMERA CALIBRATION AND BEYOND: A SURVEY". Kang Liao, Lang Nie, Shujuan Huang, Chunyu Lin, Jing Zhang, Yao Zhao, Fellow, IEEE, Moncef Gabbouj, Fellow, IEEE, Dacheng Tao, Fellow, IEEE [3].

This paper provides an in-depth examination of recent advancements in deep learning applied to camera calibration, highlighting significant challenges and suggesting future research directions. The researchers conduct an extensive review of over 100 scholarly articles, systematically categorizing various deep learning methods and strategies not only in camera calibration but also in broader related applications. This includes an analysis of different models and their specific objectives. The study delves into the utilization of deep neural networks, such as convolutional neural networks (CNNs) and generative adversarial networks (GANs), alongside diverse datasets. These datasets contribute to the training and validation of models, ensuring robust and generalized performance across different calibration scenarios. To support the research community, the survey also mentions an open-source repository that compiles the categorized methods and strategies, facilitating easier access and collaboration. By categorizing the methods, the paper provides a detailed analysis of various learning strategies employed in deep learning-based camera calibration. This encompasses a range of techniques from supervised learning to more complex paradigms such as semi-supervised and unsupervised learning. The discussion on challenges includes practical issues like dataset diversity, computational requirements, and the need for real-time processing capabilities. The survey offers a comprehensive overview of the current state-of-the-art in deep learning for camera calibration. It covers conventional models, explores various learning paradigms, and presents cutting-edge approaches that push the boundaries of what is currently possible. A detailed taxonomy is provided, which serves to organize and classify the different techniques and methodologies in the field. This taxonomy not only helps in understanding the landscape of deep learning in camera calibration but also identifies gaps and opportunities for future research. Overall, the paper aims to foster further advancements in the field by providing a thorough understanding of existing methods, their strengths and limitations, and potential avenues for innovation. The insights and detailed classifications presented in this survey are intended to guide researchers in developing more effective and efficient camera calibration techniques, leveraging the power of deep learning to address both current and emerging challenges.

"MACHINE-LEARNING-INSPIRED WORKFLOW FOR CAMERA CALIBRATION" Alexey Pak, Steffen Reichel, and Jan Burke. 2022 [4].

This paper emphasizes the critical importance of calibrating digital cameras, which are extensively used in applications such as optical measurements and computer vision. Despite the advanced performance capabilities of modern cameras, the precision and accuracy of their measurements are highly contingent on the quality of their calibration. The authors advocate for a more systematic and precise approach to camera calibration, inspired by established machine learning workflows and practical requirements in camera-based measurement systems. They propose a comprehensive calibration methodology that combines standard calibration tools with a specialized technique involving unique targets and patterns. These targets and patterns are designed to meticulously characterize the camera's imaging geometry, enabling the capture of high-fidelity calibration data. The suggested approach aims to achieve exceptionally accurate measurements, even when cameras are used over considerable distances. By employing specialized targets, the method can accurately determine intrinsic parameters (such as focal length, principal point, and distortion coefficients) and extrinsic parameters (such as the camera's position and orientation in space). Additionally, the paper highlights the importance of ensuring that the quality of the calibration data and the resulting parameters is rigorously controlled and well-understood throughout the calibration process. This includes careful consideration of the entire workflow, from data acquisition to parameter estimation, and the application of robust validation techniques to verify the accuracy and reliability of the calibration results. The authors propose that this enhanced calibration process can significantly improve the reliability of camera-based measurements in various applications, including industrial inspection, 3D reconstruction, and robotic vision systems. By integrating advanced calibration techniques with practical considerations, the methodology not only improves measurement accuracy but also facilitates better control and understanding of the calibration process itself. Ultimately, this paper calls for a shift towards more precise and standardized calibration practices, leveraging modern computational tools and techniques to meet the evolving demands of camera-based measurement technologies. This approach promises to enhance the overall effectiveness and applicability of digital cameras in critical measurement tasks, ensuring high accuracy and consistency across a range of real-world scenarios.

1.4 OBJECTIVES

The objectives of our project are

1. To determine intrinsic parameters of a camera such as focal length, principal point and lens distortion coefficients.
2. To determine extrinsic parameters of a camera such as position and orientation in 3D space.

1.5 Report Organization

Chapter 1: In this chapter, a brief introduction of camera calibration. Also it describes the scope of the problem statement, objectives and literature survey.

Chapter 2: This chapter focusses on complete detail of block diagram and methodology.

Chapter 3: This chapter focusses on the complete detail of hardware and software components along with the specifications.

Chapter 3: This chapter describes on the result and conclusion of our project and future scope.

Chapter 2

METHODOLOGY

2.1 BLOCK DIAGRAM

The block diagram of "Artificial Intelligence based Camera Calibration" is shown in Figure 2.1.

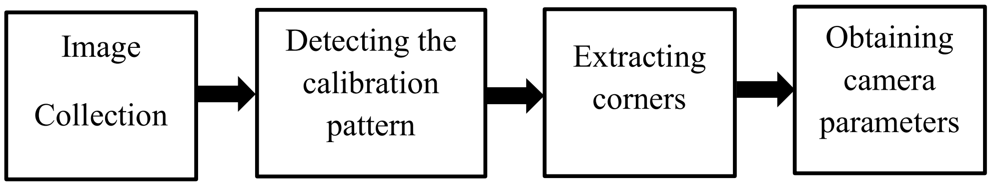

Figure 2.1: Block diagram of Artificial Intelligence based Camera Calibration

Cameras obtain the geometric data of the 3D object. The calibration of the camera has become most significant computer vision component. This allows us to acquire the camera intrinsic and extrinsic parameters. The image pixel coordinates are connected with the respective coordinates in the reference frame of camera by intrinsic parameters. Extrinsic parameters describe the camera reference frame's place and alignment in relation to a well-known reference frame of the world. Currently, calibration of camera is constantly crucial part of photogrammetric extent, with a fundamental and regularly implemented procedure of self-calibration.

2.2 METHODOLOGY

This project typically begins with a dataset creation. Dataset creation involves capturing a set of images with different scenes and angles using the camera that needs calibration and adding Gaussian noise to some of the images. This model is designed to process images and extract meaningful features that can aid in the subsequent calibration process. The dimensions of the chessboard pattern are specified. This information is crucial for accurately detecting the corners of the chessboard in the images.

Then initializes arrays to store object points and image points from all the chessboard images. These points are used in the calibration process to establish correspondences between the 3D coordinates of the chessboard corners and their 2D projections in the images. Here self-Calibration takes place. Techniques in this category do not use any calibration object, and can be considered as 0D approach because only image point correspondences are required. Just by moving a camera in a static scene, the rigidity of the scene provides in general two constraints on the cameras' internal parameters from one camera displacement by using image information alone. Therefore, if images are taken by the same camera with fixed internal parameters, correspondences between three images are sufficient to recover both the internal and external parameters which allow us to reconstruct 3-D structure up to a similarity. Although no calibration objects are necessary, a large number of parameters need to be estimated, resulting in a much harder mathematical problem.

For each image in the dataset, the code loads the image, converts it to grayscale, and resizes it to fit the input shape of the CNN model. Features are extracted from the resized image using the CNN model. If one uses a generic corner detector, such as Harris corner detector, to detect the corners in the check pattern image, the result is usually not good because the detector corners have poor accuracy (about one pixel). A better solution is to leverage the known pattern structure by first estimating a line for each side of the square and then computing the corners by intersecting the fitted lines. There are two common techniques to estimate the lines. The first is to first detect edges, and then fit a line to the edges on each side of the square. The second technique is to directly fit a line to each side of a square in the image such that the gradient on the line is maximized. One possibility is to represent the line by an elongated Gaussian, and estimate the parameters of the elongated Gaussian by maximizing the total gradient covered by the Gaussian. We should note that if the lens distortion is not severe, a better solution is to fit just one single line to all the collinear sides. This will leads a much more accurate estimation of the position of the checker corners. Then detects the corners of the chessboard pattern in the grayscale image. If corners are found, the object points (representing the 3D coordinates of the chessboard corners) and image points (the detected 2D corner locations) are appended to their respective arrays.

Once object points and image points have been collected from all the images, the camera is calibrated using calibrate Camera function. This function estimates the intrinsic and extrinsic parameters of the camera, including the camera matrix, distortion coefficients, rotation vectors, and translation vectors. These parameters provide valuable insights into the characteristics and orientation of the camera, enabling accurate geometric transformations between the 3D world and the 2D image plane.

Camera Matrix:

$$\begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

f x and fy represent the focal lengths in the x and y directions.

Cx and Cy denote the optical centres in the x and y directions.

Distortion Coefficient:

[k1 k2 p1 p2 k3**]**

K1, k2, and k3 are radial distortion coefficients.

P1 and p2 are tangential distortion coefficients.

Rotation Vector:

$$\begin{bmatrix} r_x \\ r_y \\ r_z \end{bmatrix}$$

r x, r y, and rz are the rotation angles around the x, y, and z axes, respectively.

Translation Vector:

$$\begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix}$$

t x, ty, and t z represent the translation along the x, y, and z axes, respectively.

These matrices and vectors provide insights into the camera's intrinsic properties (focal length, optical center, distortion) and extrinsic parameters (position and orientation relative to the scene), essential for geometric transformations in computer vision tasks.

Compared with traditional methods, camera self-calibration approach only requires the establishment of the correspondence between the image .it is flexible, but the calibration process of the self-calibration method is complex, it does not apply to the in-time Updating occasion, and is only applicable to the situation that calls for less precision ,such as virtual reality, owing to the non-linear calibration of self-calibration method, relying on good initial estimates and lacking of robustness.

Traditional method of camera calibration is known as using a calibration piece, which of known structure and high precision processing, as space reference point and through the correspondence between the space points and the image points to establish the constraints of the camera model parameters, then to obtain these parameters by optimization algorithm. The traditional methods can use any camera model and have high precision of calibration, so that when applications required highly accuracy, often use this approach. The typical representatives are as follows: the direct linear transformation method (DLT method), nonlinear optimization method, two-step method, planar template method, dual-plane method and so on.

1. Direct linear transformation method – It is first proposed by Abdel-Aziz and Karara in 1971. In this method, a set of intermediate parameters are defined, and then the intrinsic and extrinsic parameters of camera model could be solved by establishing and solving linear equations without iterative calculations, which is the attraction of the direct linear transformation method. Because of this approach involved few parameters and calculated easily, it is relatively easy to be adopted. However, this approach did not consider the non-linear distortion problems during the camera works. In-order-to increase accuracy, the direct linear transformation method can be expanded to include these non-linear factors and used non-linear means to solve them.
2. Nonlinear optimization technique - Considered the distortion of the camera, nonlinear optimization technique use a large number of unknowns and large-scale nonlinear optimize, which makes the computational cost become larger with the increased accuracy of the nonlinear model. Nonlinear optimization technique's precision is high, but its algorithm is fussy and slow, and the iterative nature of algorithm needs a good initial estimate. If the iterative process of design is not appropriate, especially in high-distortion conditions, the optimization process may be unstable, resulting in instability and even the results of the error, so its validity

is not high. The advantages and disadvantages of the distortion calibration method are proposed based on a simple and rapid calibration method of the lens distortion parameters. This method makes use of the perspective projection of the cross-ratio invariance, in the distortion model is a first-order radial distortion of the cases, only needs the space to a total line of the image coordinates of four points and their cross-ratio, for example, the establishment of a quadratic equation distortion parameters can be calibrated. This method algorithm is simple and easy to implement. Literature in its image on the basis of a correction, using linear calibration completed the calibration, to avoid other nonlinear optimization methods may encounter instability.

3. The direct linear transformation method didn't considered the distortion of the camera lens, while the non-linear model methodology could take the non-linear factors into account, but it made the calculations much more complicated, and the result even may not be exact solutions. Tsai studied and summarized the traditional calibration method before 1987; there was radial distortion factor of the camera model, on the basis of which a practical two-step calibration algorithm was proposed. First, the perspective matrix transform method was used to solve the camera parameters of the linear systems, and then used the achieved parameters as the initial value, given the distortion factor, and use optimization methods again to improve the calibration accuracy. The advantage of this method was that the model assumed the camera lens distortion was radial and the vector remained unchanged from the image center to the point direction of the image, regardless of changes in distortion. Because of its high calibration accuracy, it significantly reduced the space dimension of parameters, so it was suitable for precise measurements. The disadvantage is that the calibration of the equipment requirements is relatively high, not suitable to use on a simplified calibration.

Chapter 3

PROJECT IMPLEMENTATION

3.1 HARDWARE

3.1.1 CAMERA

AI-based camera calibration for mobile cameras involves leveraging specialized hardware components and advanced algorithms to optimize image quality and enhance user experience. These systems utilize dedicated processing units, such as AI Processing Units (APUs) or specialized image processing chips, to perform real-time analysis of captured images.

Working in tandem with the camera's Image Signal Processor (ISP) and sensor fusion technology, these components process image data efficiently. Advanced algorithms, including computer vision techniques and signal processing methods, are employed to analyze the scene and dynamically adjust camera settings.

Tasks such as autofocus optimization, exposure control, white balance adjustment, and scene recognition are performed using these algorithms without relying on machine learning techniques.

By integrating these hardware components and algorithms, mobile cameras can deliver superior image quality, improved low-light performance, and optimized shooting experiences for users.

3.2 SOFTWARE

3.2.1 Anaconda Navigator

Anaconda Navigator is a graphical user interface (GUI) included with Anaconda, a popular Python distribution for data science and machine learning. While Anaconda itself doesn't directly relate to camera calibration or artificial intelligence (AI), it provides a platform where we can manage various Python packages and environments, including those used for AI and computer vision tasks like camera calibration. For camera calibration in the context of AI, we might use libraries like OpenCV or TensorFlow, which can be easily managed and installed using Anaconda. Anaconda Navigator simplifies the process of creating and managing different environments for different projects, making it convenient for AI developers to work on camera calibration tasks alongside other AI projects.

TensorFlow is one of the most popular deep learning frameworks, and Anaconda Navigator makes it easy to work with TensorFlow within our Python environment. Here's how Anaconda Navigator integrates with TensorFlow:

1. **Installation:** Anaconda Navigator simplifies the installation of TensorFlow. we can search for TensorFlow in the Navigator interface and install it with just a few clicks. Additionally, we can choose the version of TensorFlow you want to install, making it convenient for experimenting with different releases or maintaining compatibility with existing projects.
2. **Environment Management:** Anaconda Navigator allows you to create separate environments for your TensorFlow projects. This is particularly useful when working with TensorFlow because we may need different versions or configurations for different projects. With Anaconda Navigator, you can manage these environments effortlessly, ensuring that TensorFlow installations remain isolated and consistent.
3. **Package Management:** Once TensorFlow is installed, Anaconda Navigator provides tools for updating, removing, and managing TensorFlow and its dependencies within our environments. This ensures that we can keep our TensorFlow installations up-to-date and compatible with your projects' requirements.
4. **Integration with Jupyter Notebooks:** Anaconda Navigator seamlessly integrates with Jupyter Notebooks, which is a popular tool for developing and experimenting with TensorFlow models. we can launch Jupyter Notebooks from within Anaconda Navigator and start coding TensorFlow applications.

Overall, Anaconda Navigator streamlines the process of working with TensorFlow by providing a user-friendly interface for installation, environment management, package management, and integration with tools like Jupyter Notebooks. This makes it easier for both beginners and experienced deep learning practitioners to leverage the power of TensorFlow for their artificial intelligence and machine learning projects.

TensorFlow is an open-source machine learning framework developed by Google, widely used for various artificial intelligence tasks, including camera calibration. In the context of camera calibration, TensorFlow can be employed for tasks such as:

1. **Intrinsic Parameter Estimation:** TensorFlow can be used to build neural network models that estimate the intrinsic parameters of a camera, such as focal length, principal point, and distortion coefficients. These parameters are crucial for correcting image distortions caused by the camera lens.
2. **Extrinsic Parameter Estimation:** TensorFlow can also be used to estimate the extrinsic parameters of a camera, which define its position and orientation in 3D space relative to the scene being captured. This is essential for tasks like 3D reconstruction and augmented reality.
3. **Distortion Correction:** TensorFlow models can be trained to correct various types of distortions present in images captured by cameras, such as radial and tangential distortions. This is particularly important for computer vision tasks where accurate geometric information is required.
4. **Camera Pose Estimation:** TensorFlow can be used to estimate the pose (position and orientation) of a camera relative to the scene being captured. This information is useful for applications like robot navigation, object tracking, and augmented reality.

TensorFlow provides a powerful platform for developing and deploying machine learning models for camera calibration and related tasks in the field of artificial intelligence and computer vision. Its flexibility, scalability, and extensive library of pre-built modules make it a popular choice among researchers and developers working in this domain.

Keras related to artificial intelligence based on camera calibration:

1. **High-level API:** Keras provides a user-friendly interface for building and training neural networks, making it easier to develop models for camera calibration tasks.
2. **Integration with TensorFlow:** Keras seamlessly integrates with TensorFlow, enabling efficient computation for camera calibration tasks while leveraging TensorFlow's capabilities as a backend.
3. **Transfer Learning:** Keras supports transfer learning, allowing users to adapt pre-trained models for camera calibration by fine-tuning them with specific data.
4. **Customization:** Users can define custom loss functions and metrics tailored to camera calibration objectives, optimizing models for accuracy and performance.
5. **Data Augmentation:** Keras offers tools for data augmentation, essential for training robust camera calibration models by increasing the diversity of training data.

Overall, Keras provides a convenient platform for developing and training neural network models tailored to camera calibration tasks, with features like integration, customization, and transfer learning enhancing its utility in artificial intelligence applications.

CNN Architecture:

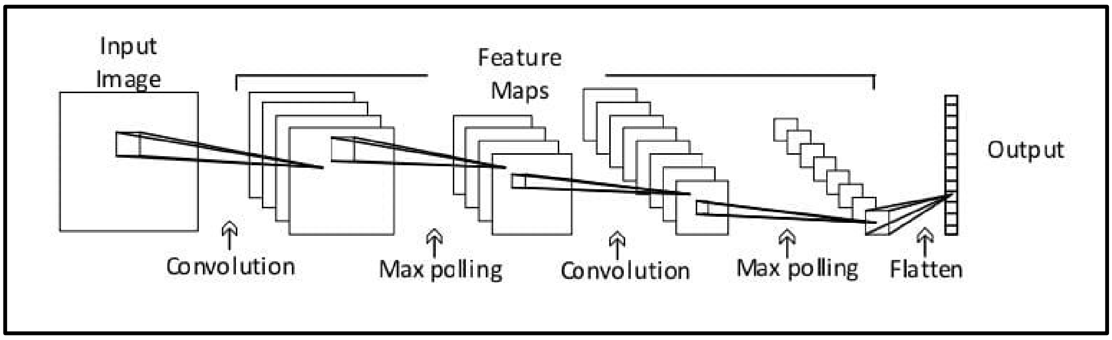

Figure 3.1: Generic CNN model

Artificial intelligence-based camera calibration leverages Convolutional Neural Network (CNN) architecture for feature extraction from images. The CNN is trained to identify relevant patterns and features in images, which are then used for the camera calibration process.

Convolutional neural network is a multi-layer system. It consists of multiple layers like convolutional layers, max-pooling and flatten layer. The generic model of a convolutional neural network is, as shown in Figure 3.1.

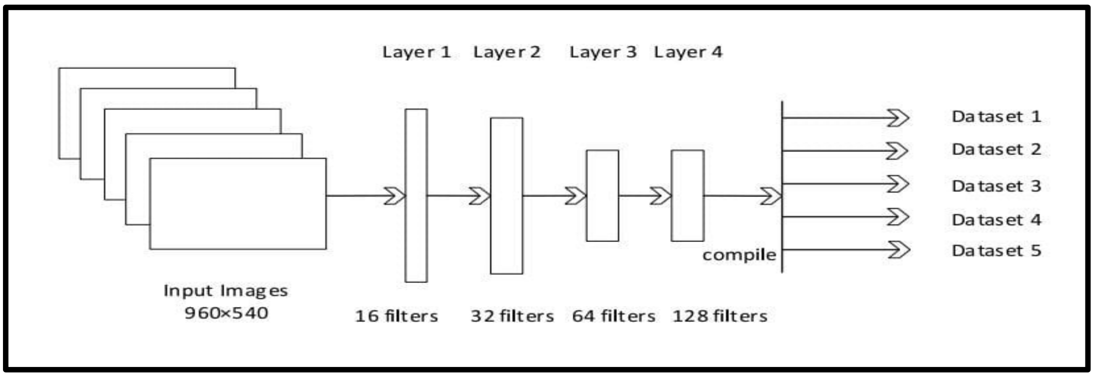

Figure 3.2: The proposed CNN system

The proposed network is comprises of four convolutional layers, two dense layers and one flatten layer as shown in Figure. 3.2. First layers extract the features from the pixels, while the descendant layers combine these feature into a chessboard corner weight as shown in Figure 3.2.

CNNs are adept at learning hierarchical representations of features from images. In camera calibration, these features can include edges, corners, and other distinctive patterns present in calibration targets like chessboard patterns or grids. Once trained, the CNN can detect specific patterns or features in images that are indicative of the calibration target, such as corners of a chessboard pattern. This detection is crucial for accurately identifying corresponding points in the 2D image plane and the 3D world coordinates. Prior to feeding images into the CNN, preprocessing steps such as resizing, normalization, and color space conversion may be applied to ensure compatibility with the network's input requirements and enhance feature extraction. Extracted features from the CNN are integrated into the camera calibration pipeline. These features serve as input for algorithms that estimate the intrinsic and extrinsic parameters of the camera, such as the camera matrix, distortion coefficients, rotation, and translation vectors. CNNs may be trained using annotated datasets containing images of calibration targets and corresponding ground truth parameters. Fine-tuning techniques may also be employed to adapt pre-trained CNN models to specific calibration tasks or imaging conditions.

In general, the use of artificial intelligence, particularly Convolutional Neural Networks (CNNs), in camera calibration offers several advantages:

1. **Automated Feature Extraction:** CNNs excel at automatically extracting relevant features from images, eliminating the need for manual feature selection or extraction methods. This automation streamlines the calibration process and reduces human intervention.
2. **Robustness to Variations:** CNNs can learn and adapt to a wide range of patterns, textures, and lighting conditions present in images. This robustness enables accurate calibration even in challenging environments with varying imaging conditions.
3. **Flexibility and Adaptability:** CNN architectures can be tailored and fine-tuned to specific calibration tasks, camera models, or calibration targets. This flexibility allows for customization and optimization of the calibration process based on the application requirements.
4. **Scalability:** With the availability of large-scale datasets and computational resources, CNN-based calibration approaches can scale effectively to handle large volumes of calibration data and diverse camera setups.
5. **Reduced Human Effort:** By automating the feature extraction and pattern detection processes, CNN-based calibration reduces the manual effort required for data

preprocessing and analysis, making the calibration process more efficient and less labor-intensive.

6. **Improved Accuracy:** Leveraging the power of deep learning, CNN-based calibration methods can achieve higher levels of accuracy and precision compared to traditional calibration techniques. This improved accuracy leads to more reliable camera parameter estimates and better performance in downstream computer vision tasks.

Overall, the integration of artificial intelligence, specifically CNNs, in camera calibration enhances the efficiency, accuracy, and scalability of the calibration process, paving the way for advanced applications in fields such as robotics, augmented reality, and autonomous driving.
The first layer of the network has a convolutional filter with a large number of neurons. The activation function used for the first layer is ReLU (Rectified Linear Unit). The spatial size of the filter is large to avoid the blur effect of the images. There are 16, 32, 64, and 128 neurons. Stride, max-pooling is 2 and 3*3 convolutions in four convolutional layers respectively. The activation function in each layer is ReLUs.

Initially we captured three sets of data using a camera. For each image, the first step is to read the image file, which involves loading the pixel data into memory. The image is then converted to grayscale to simplify further processing, after converting to grayscale, the image is resized to a standard size to ensure consistency across all images. The next step involves using a Convolutional Neural Network (CNN) to extract features from the image. CNNs are particularly effective at identifying patterns and details in images, which is essential for the accurate detection of chessboard corners. These corners are crucial for camera calibration as they provide known reference points. If the chessboard corners are successfully found, then records the object points (the actual 3D coordinates of the corners in the real world) and the image points (the 2D coordinates of the corners in the image). These points are stored for later use in the calibration process. draws the corners on the image and displays it. The program pauses, waiting for a key press before proceeding to the next image. This process is repeated for each image in the set, accumulating a comprehensive set of object and image points. Once all images have been processed, the code uses the collected points to calibrate the camera. Calibration involves computing the camera’s internal parameters, such as focal length, Principal point and lens distortion coefficients. These parameters are essential for correcting distortions in images captured by the camera, extrinsic parameters, which describe the camera's position and orientation relative to the chessboard for each image. This detailed information is invaluable for understanding the camera's behavior and improving the accuracy of captured images. The flowchart for Artificial Intelligence based camera calibration is shown in Figure 3.3.

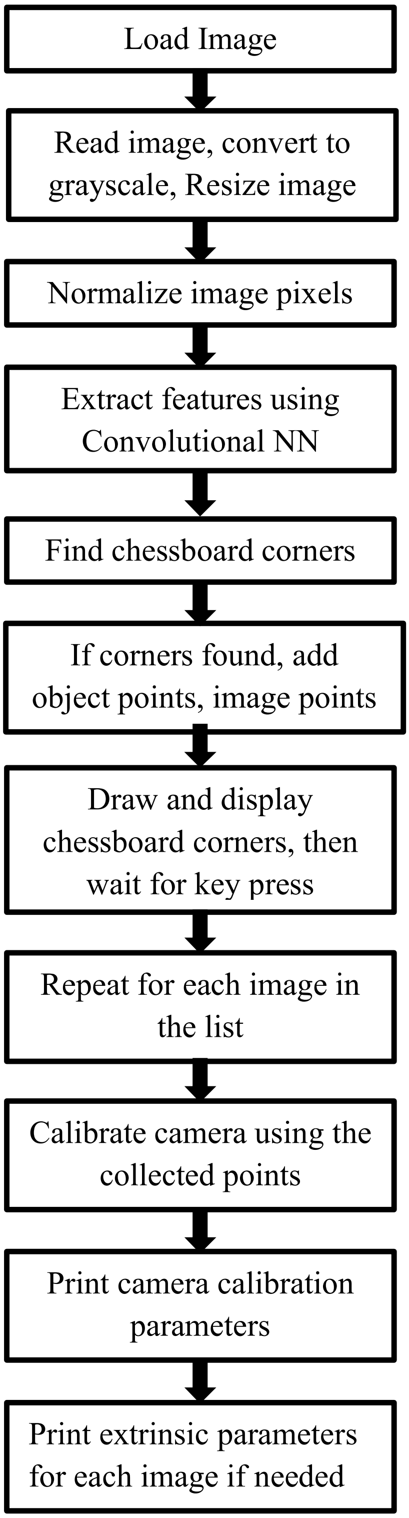

Figure. 3.3: Flowchart for Artificial Intelligence based Camera Calibration

CHAPTER 4

RESULTS AND DISCUSSION

4.1 RESULT

The input image 1 is shown in Figure 4.1 and the output image and parameters for input image 1 are shown in Figure 4.2 and Figure 4.3.

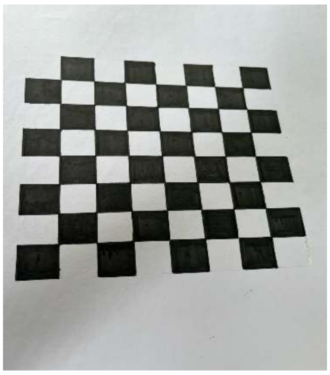

Figure 4.1: Input Image 1

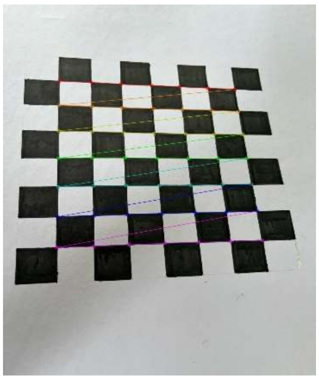

Figure 4.2: Output Image 1

```
Camera Matrix:
[[1.23262783e+03 0.00000000e+00 4.98146020e+02]
 [0.00000000e+00 1.19581967e+03 5.03413200e+02]
 [0.00000000e+00 0.00000000e+00 1.00000000e+00]]

Distortion Coefficients:
[[-5.02383653e-01  6.85568868e+00 -1.14204562e-02  1.67785306e-02
  -2.47335716e+01]]

Extrinsic parameters for image 1:
Rotation Vector:
[[-0.36419812]
 [-0.23963361]
 [-0.03420629]]
Translation Vector:
[[-3.40663997]
 [-2.54624561]
 [12.65956932]]
```

Figure 4.3: Parameters for Input image 1

The input image 2 is shown in Figure 4.4 and the output image and parameters for input image 2 are shown in Figure 4.5 and Figure 4.6.

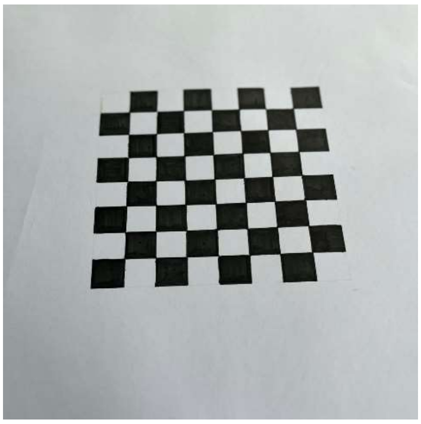

Figure 4.4: Input Image 2

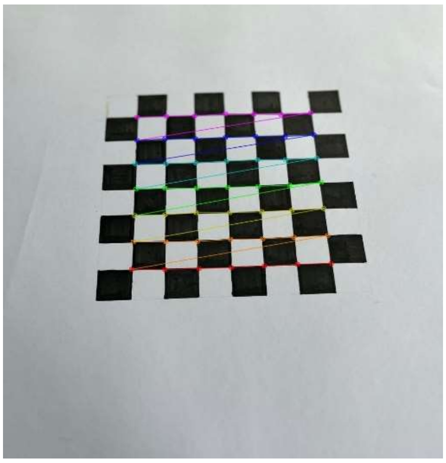

Figure 4.5: Output Image 2

```
Camera Matrix:
[[1.53983960e+03 0.00000000e+00 5.57985675e+02]
 [0.00000000e+00 1.80443650e+03 6.38433810e+02]
 [0.00000000e+00 0.00000000e+00 1.00000000e+00]]

Distortion Coefficients:
[[ 1.72396494e-01 -1.74601867e+01 -9.45688890e-02 -4.93666798e-03
   1.59676110e+02]]

Extrinsic parameters for image 1:
Rotation Vector:
[[-0.05328182]
 [ 0.97545488]
 [ 2.93026267]]
Translation Vector:
[[ 2.17393947]
 [-0.30240305]
 [21.48962338]]
```

Figure 4.6: Parameters for Input image 2

The input image 3 is shown in Figure 4.7 and the output image and parameters for input image 3 are shown in Figure 4.8 and Figure 4.9.

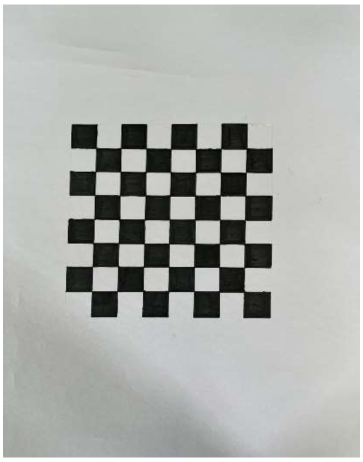

Figure 4.7: Input Image 3

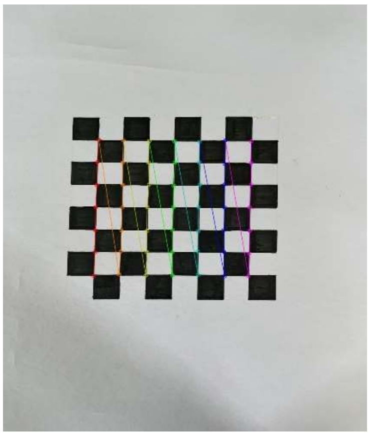

Figure 4.8: Output Image 3

```
Camera Matrix:
[[8.01753559e+03 0.00000000e+00 7.14858527e+02]
 [0.00000000e+00 8.11970653e+03 6.30400723e+02]
 [0.00000000e+00 0.00000000e+00 1.00000000e+00]]

Distortion Coefficients:
[[-2.50811687e+00  8.66468289e+02  8.89856754e-02  4.38304431e-02
   3.17241412e+00]]

Extrinsic parameters for image 1:
Rotation Vector:
[[-0.13714336]
 [-0.17197808]
 [-1.55690692]]
Translation Vector:
[[ -6.95312599]
 [  2.56428385]
 [117.0531451 ]]
```

Figure 4.9: Parameters for Input image 3

The input image 4 is shown in Figure 4.10 and the output image and parameters for input image 3 are shown in Figure 4.11 and Figure 4.112.

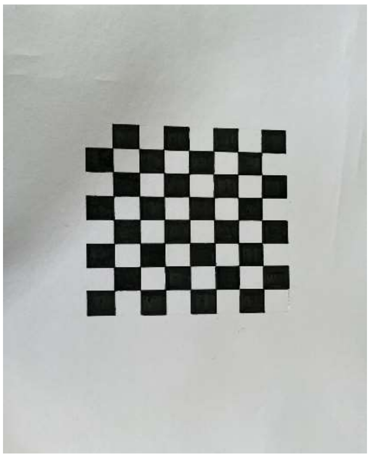

Figure 4.10: Input Image 4

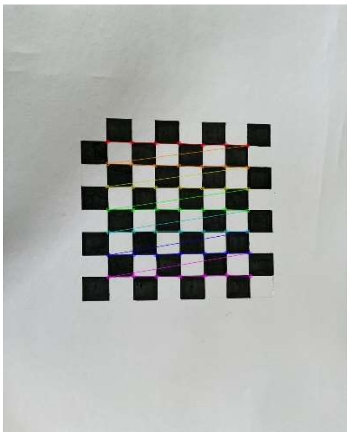

Figure 4.11: Output Image 4

```
Camera Matrix:
[[4.56848600e+03 0.00000000e+00 3.83556806e+02]
 [0.00000000e+00 3.78690155e+03 5.95305617e+02]
 [0.00000000e+00 0.00000000e+00 1.00000000e+00]]

Distortion Coefficients:
[[-3.81217726e+00  7.32750321e+02 -4.46908080e-03  1.45787110e-02
   1.01481101e+01]]

Extrinsic parameters for image 1:
Rotation Vector:
[[-2.97705747e-02]
 [-5.93672343e-01]
 [-6.24253212e-05]]
Translation Vector:
[[-1.13488053]
 [-2.73543239]
 [55.67564638]]
```

Figure 4.12: Parameters for Input image 4

The input image 5 is shown in Figure 4.13 and the output image and parameters for input image 3 are shown in Figure 4.14 and Figure 4.15.

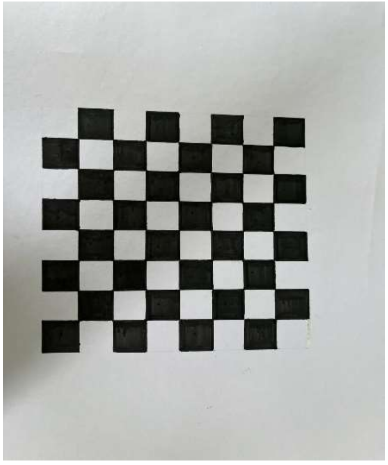

Figure 4.13: Input Image 5

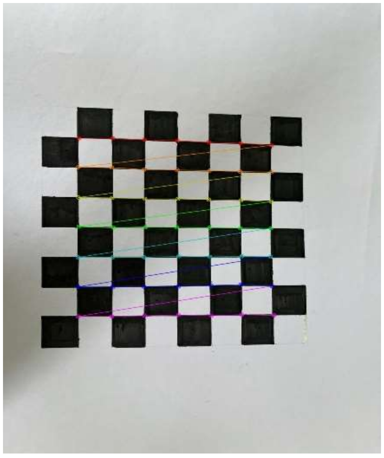

Figure 4.14: Output Image 5

```
Camera Matrix:
[[3.07062198e+03 0.00000000e+00 4.76971184e+02]
 [0.00000000e+00 2.89858887e+03 6.34858655e+02]
 [0.00000000e+00 0.00000000e+00 1.00000000e+00]]

Distortion Coefficients:
[[-1.27550461e+00  3.21997893e+02  1.51030650e-02  2.55990727e-02
  -1.42843439e+04]]

Extrinsic parameters for image 1:
Rotation Vector:
[[-0.04604565]
 [-0.36998984]
 [ 0.00321997]]
Translation Vector:
[[-3.20424321]
 [-2.99342853]
 [34.22165568]]
```

Figure 4.15: Parameters for Input image 5

The intrinsic parameters and extrinsic parameters for image 1 is shown in Table 4.1.

Input Image 1	Intrinsic parameters	Focal length: $f_x = 1.23262783e{+}03$ $f_y = 1.19581967e{+}03$
		Principal points: $p_x = 4.98146020e{+}02$ $p_y = 5.03413200e{+}02$
		Distortion coefficients: $k_1 = -5.02383653e{-}01$ $k_2 = 6.85568868e{+}00$ $p_1 = -1.14204562e{-}02$ $p_2 = 1.67785306e{-}02$ $k_3 = -2.47335716e{+}01$
	Extrinsic parameters	Rotation Vector: $r_x = -0.36419812$ $r_y = -0.23963361$ $r_z = -0.03430629$
		Translation Vector: $t_x = -3.40663997$ $t_y = -2.54624561$ $t_z = 12.65956932$

Table 4.1: Intrinsic and Extrinsic parameters for Image 1

The Table 4.1 consists of Intrinsic and Extrinsic parameters for input image 1. The Intrinsic parameters consists of focal length, principal points and distortion coefficients. The focal length can be represented as f_x, f_y, principal points can be represented as p_x, p_y and distortion coefficients can be represented as k_1, k_2, p_1, p_2, k_3. Here, k_1, k_2, and k_3 are the radial distortion coefficients and p_1 and p_2 are the tangential distortion coefficients. The Extrinsic parameters consists of rotation vector and translation vector. The rotation vector can be represented as r_x, r_y, r_z and translation vector can be represented as t_x, t_y, t_z.

The intrinsic parameters and extrinsic parameters for image 2 is shown in Table 4.2.

Input Image 2	Intrinsic parameters	Focal length: f_x = 1.53983960e+03 f_y = 1.80443640e+03
		Principal points: p_x = 5.57985675e+02 p_y = 6.38433810e+02
		Distortion coefficients: k_1 = 1.72396494e-01 k_2 = -1.74601867e+01 p_1 = -9.45688890e-02 p_2 = -4.93666798e-03 k_3 = 1.59676110e+02
	Extrinsic parameters	Rotation Vector: r_x = -0.05328182 r_y = 0.97545488 r_z = 2.93026267
		Translation Vector: t_x = 2.17393947 t_y = -0.30240305 t_z = 21.48962338

Table 4.2: Intrinsic and Extrinsic parameters for Image 2

The Table 4.2 consists of Intrinsic and Extrinsic parameters for input image 2. The Intrinsic parameters consists of focal length, principal points and distortion coefficients. The focal length can be represented as f_x, f_y, principal points can be represented as p_x, p_y and distortion coefficients can be represented as k_1, k_2, p_1, p_2, k_3. Here, k_1, k_2, and k_3 are the radial distortion coefficients and p_1 and p_2 are the tangential distortion coefficients. The Extrinsic parameters consists of rotation vector and translation vector. The rotation vector can be represented as r_x, r_y, r_z and translation vector can be represented as t_x, t_y, t_z.

The intrinsic parameters and extrinsic parameters for image 3 is shown in Table 4.3.

Input Image 3	Intrinsic parameters	Focal length: $f_x = 8.01753559e+03$ $f_y = 8.11970653e+03$
		Principal points: $p_x = 7.14858527e+02$ $p_y = 6.30400723e+02$
		Distortion coefficients: $k_1 = -2.50811687e+00$ $k_2 = 8.66468289e+02$ $p_1 = 8.89856754e-02$ $p_2 = 4.38304431e-02$ $k_3 = 3.17241412e+00$
	Extrinsic parameters	Rotation Vector: $r_x = -0.13714336$ $r_y = -0.17197808$ $r_z = -1.55690692$
		Translation Vector: $t_x = -6.95312599$ $t_y = 2.56428385$ $t_z = 117.0531451$

Table 4.3: Intrinsic and Extrinsic parameters for Image 3

The Table 4.3 consists of Intrinsic and Extrinsic parameters for input image 3. The Intrinsic parameters consists of focal length, principal points and distortion coefficients. The focal length can be represented as f_x, f_y, principal points can be represented as p_x, p_y and distortion coefficients can be represented as k_1, k_2, p_1, p_2, k_3. Here, k_1, k_2, and k_3 are the radial distortion coefficients and p_1 and p_2 are the tangential distortion coefficients. The Extrinsic parameters consists of rotation vector and translation vector. The rotation vector can be represented as r_x, r_y, r_z and translation vector can be represented as t_x, t_y, t_z.

The intrinsic parameters and extrinsic parameters for image 4 is shown in Table 4.4.

Input Image 4	Intrinsic parameters	Focal length: $f_x = 4.56848600e+03$ $f_y = 3.78690155e+03$
		Principal points: $p_x = 3.83556806e+02$ $p_y = 5.95305617e+02$
		Distortion coefficients: $k_1 = -3.81217726e+00$ $k_2 = 7.32750321e+02$ $p_1 = -4.46908080e-03$ $p_2 = 1.45787110e-02$ $k_3 = 1.01481101e+01$
	Extrinsic parameters	Rotation Vector: $r_x = -2.97705747e-02$ $r_y = -5.93672343e-01$ $r_z = -6.24253212e-05$
		Translation Vector: $t_x = -1.13488053$ $t_y = -2.73543239$ $t_z = 55.67564638$

Table 4.4: Intrinsic and Extrinsic parameters for Image 4

The Table 4.4 consists of Intrinsic and Extrinsic parameters for input image 4. The Intrinsic parameters consists of focal length, principal points and distortion coefficients. The focal length can be represented as f_x, f_y, principal points can be represented as p_x, p_y and distortion coefficients can be represented as k_1, k_2, p_1, p_2, k_3. Here, k_1, k_2, and k_3 are the radial distortion coefficients and p_1 and p_2 are the tangential distortion coefficients. The Extrinsic parameters consists of rotation vector and translation vector. The rotation vector can be represented as r_x, r_y, r_z and translation vector can be represented as t_x, t_y, t_z.

The intrinsic parameters and extrinsic parameters for image 5 is shown in Table 4.5.

Input Image 5	Intrinsic parameters	Focal length: $f_x = 3.07062198e+03$ $f_y = 2.89858887e+03$
		Principal points: $p_x = 4.76971184e+02$ $p_y = 6.34858655e+02$
		Distortion coefficients: $k_1 = -1.27550461e+00$ $k_2 = 3.21997893e+02$ $p_1 = 1.51030650e-02$ $p_2 = 2.55990727e-02$ $k_3 = -1.42843439e+04$
	Extrinsic parameters	Rotation Vector: $r_x = -0.04604565$ $r_y = -0.36998984$ $r_z = 0.00321997$
		Translation Vector: $t_x = -3.20424321$ $t_y = -2.99342853$ $t_z = 34.22165568$

Table 4.5: Intrinsic and Extrinsic parameters for Image 5

The Table 4.5 consists of Intrinsic and Extrinsic parameters for input image 5. The Intrinsic parameters consists of focal length, principal points and distortion coefficients. The focal length can be represented as f_x, f_y, principal points can be represented as p_x, p_y and distortion coefficients can be represented as k_1, k_2, p_1, p_2, k_3. Here, k_1, k_2, and k_3 are the radial distortion coefficients and p_1 and p_2 are the tangential distortion coefficients. The Extrinsic parameters consists of rotation vector and translation vector. The rotation vector can be represented as r_x, r_y, r_z and translation vector can be represented as t_x, t_y, t_z.

4.2 CONCLUSION

The process for camera calibration using artificial intelligence. By leveraging AI techniques, such as convolutional neural networks (CNNs), the calibration process becomes more automated, efficient, and accurate. The process begins with loading images and then iterates through each image, performing essential preprocessing steps like grayscale conversion, resizing, and normalization. Feature extraction using CNNs helps identify key points, such as chessboard corners, necessary for calibration. Once corners are detected, object points and image points are collected for calibration purposes. The camera is then calibrated using these points, resulting in the determination of camera calibration parameters. Finally, the calibration parameters are printed, providing valuable information for subsequent tasks. Additionally, extrinsic parameters can be optionally printed for each image if needed, enhancing the understanding of the camera's position and orientation relative to the scene. Overall, this illustrates a systematic and efficient approach to camera calibration, enabled by artificial intelligence, which promises improved performance and reliability in various applications.

4.3 FUTURE SCOPE

Artificial intelligence (AI)-based camera calibration holds immense potential for enhancing imaging systems across various industries.

1. **Automated Calibration:** AI algorithms can automate the calibration process, reducing manual intervention and human error. By analyzing captured images and adjusting camera parameters such as focus, exposure, and white balance, AI can optimize image quality and consistency.
2. **Adaptive Calibration:** AI can enable cameras to adapt to changing environmental conditions in real-time. For example, in outdoor surveillance systems, AI algorithms can dynamically adjust camera settings to compensate for varying light conditions, ensuring optimal image quality throughout the day.
3. **Calibration in Autonomous Vehicles:** In the realm of autonomous vehicles, AI-based camera calibration is crucial for accurate perception of the surrounding environment. By continuously calibrating onboard cameras, AI systems can improve object detection, lane tracking, and obstacle avoidance, enhancing overall safety and reliability.

4. **Medical Imaging:** AI-powered camera calibration has significant applications in medical imaging, where precise measurements are critical for diagnosis and treatment. AI algorithms can calibrate medical imaging devices such as MRI and CT scanners, ensuring accurate spatial mapping and improved diagnostic accuracy.
5. **Augmented Reality (AR) and Virtual Reality (VR):** AI-driven camera calibration plays a vital role in AR and VR applications, where virtual objects need to be accurately overlaid onto the real-world environment. By calibrating cameras in AR/VR headsets and devices, AI ensures seamless integration of virtual content with the user's surroundings, enhancing the immersive experience.
6. **Remote Sensing and Earth Observation:** AI-enabled camera calibration can improve the accuracy of satellite imagery and remote sensing data. By automatically calibrating satellite cameras, AI algorithms can correct for atmospheric distortions and sensor noise, providing more reliable information for applications such as environmental monitoring, agriculture, and urban planning.
7. **Industrial Inspection and Robotics:** In manufacturing and robotics, AI-driven camera calibration enhances quality control and automation processes. By calibrating vision systems and robotic cameras, AI can improve defect detection, object recognition, and robotic manipulation tasks, leading to greater efficiency and productivity on the factory

Overall, the future scope of AI-based camera calibration is vast and holds the promise of revolutionizing imaging systems across diverse fields, from autonomous vehicles and healthcare to entertainment and industrial automation.

Compared with traditional methods, camera self-calibration approach only requires the establishment of the correspondence between the image .it is flexible, but the calibration process of the self-calibration method is complex, it does not apply to the in-time Updating occasion, and is only applicable to the situation that calls for less precision ,such as virtual reality, owing to the non-linear calibration of self-calibration method, relying on good initial estimates and lacking of robustness

REFERENCES

[1]. Artificial Intelligence based Camera Calibration. Syed Navid Raza, Suk Gyu Lee, Hafiz Raza ur Rehman, Gyu Sang Choi. 2019 IEEE/CVF International Conference on Computer Vision (ICCV).

[2]. Deep Single Image Camera Calibration with Radial Distortion. Manuel Lopez-Antequera, Roger Mari, Pau Gargallo, Yubin Kuang, Javier Gonzalez-Jimenez, Gloria Haro**.**

[3]. Deep Learning for Camera Calibration and Beyond: A Survey Kang Liao, Lang Nie, Shujuan Huang, Chunyu Lin, Jing Zhang, Yao Zhao, Fellow, IEEE, Moncef Gabbouj, Fellow, IEEE, Dacheng Tao, Fellow, IEEE.

[4]. Machine-Learning-Inspired Workflow for Camera Calibration. Alexey Pak, Steffen Reichel, and Jan Burke. 2022.

[5]. "Camera Calibration Using Neural Network" Márcio Mendonça, Ivan N. da Silva, José E. C. Castanho.

[6]. "CCDN: Checkerboard Corner Detection Network for Robust Camera Calibration" Ben Chen, Caihua Xiong, Qi Zhang, State Key Laboratory of Digital Manufacturing Equipment and Technology, Huazhong University of Science and Technology, Wuhan, Hubei 430074, China

[7]. "Accurate Detection and Localization of Checkerboard Corners for Calibration". Alexander Duda Udo Frese Kraken Robotik GmbH German Research Center for Artificial Intelligence, Cyber-Physical Systems Bremen, Germany 2013.

[8]. Qi, Wang, Fu Li, and Liu Zhen Zhong "Review on camera calibration." Control and Decision Conference (CCDC), 2010 Chinese. IEEE, 2010.