Master Thesis

Master of Science (M.sc)

Department of Tech and Software

Major: Data Science (DS)

Topic: Computer Vision in Effective Education

Author: Pankaj Balotra

Matrikel-Number: 87034917

First supervisor: Prof. Dr Raja Hashim Ali

Second supervisor: Prof. Dr Iftikhar Ahmed

Submitted on: 28.02.2024

Statutory Declaration:

I hereby declare that I have developed and written the enclosed Master Thesis completely by myself and have not used sources or means without declaration in the text. I clearly marked and separately listed all the literature and all the other sources which I employed when producing this academic work, either literally or in content. I am aware that the violation of this regulation will lead to the failure ofthe thesis.

Berlin, ...

Place, Date Signature

Abstract

The integration of computer vision and education represents a transformative paradigm, offering unprecedented opportunities to reshape conventional learning methodologies. This research endeavors to pioneer a distinctive approach to education by leveraging the capabilities of computer vision, particularly contour detection, sorting, and measurement based on pixel distances.

The first phase of our investigation involves developing a sophisticated Python code for extracting object features using advanced computer vision techniques. This code excels in capturing nuanced object characteristics, including lengths, widths, heights, and other pertinent parameters, through contour detection and pixel distance measurements.

Concurrently, our research explores integrating this extracted data with Blender, a versatile 3D modeling and animation platform. This fusion of computer vision and 3D animation holds immense potential for creating dynamic and engaging educational content. Mathematical equations derived from the extracted object features serve as the foundation for crafting immersive 3D animations. This amalgamation not only introduces a novel dimension to education but also addresses the diverse learning styles of students through visually stimulating and interactive content.

The outcomes of this research illuminate the transformative potential of computer vision in effective education. By presenting a novel pipeline from object feature extraction to 3D animation integration, this study contributes valuable insights to both the computer vision and education communities. The innovative methodologies and practical applications explored herein represent a significant stride toward realizing a more dynamic, personalized, and engaging educational landscape. As we navigate the intersection of computer vision and education, this research paves the way for future endeavors seeking to redefine learning methodologies through technological innovation.

Table of Contents:

List of Abbreviations

OCR	Optical Character Recognition
HCR	Handwritten Character Recognition
ICR	Intelligent Character Recognition
HTR	Handwritten Text Recognition
BPA	Business Process Automation
RPA	Robotic Process Automation
CTC	Connectionist Temporal Classification
LSTM	Long short-term Memory
RNN	Recurrent neural networks
HMM	hidden Markov model
CNN	Convolutional neural networks
SVM	Support Vector Machine
NLP	Natural Language Processing
LM	Language Model
MLP	Multilayer Perceptron
BPTT	backpropagation through time algorithm
BRNN	Bidirectional recurrent neural networks
GRU	Gated Recurrent units
MDLSTM	Multidimensional LSTM
HMER	Handwritten Mathematical Expression Recognition
DCNN	Deep Convolutional Neural Networks
ReLu	Rectified Linear Unit
YOLO	You look only once
FCN	Fully convolution neural network
VAN	*Vertical Attention* Network
DAN	*Decoupled Attention* mechanism
Bi-LSTM	Bi-directional LSTM
GAN	Generative Adversarial Network

List of Tables:

List of Figures:

Chapter 1

Introduction

1.1 Background

A turning point in pedagogy's development has been reached with the introduction of computer vision into education, which gives a plethora of chances to reimagine the educational process. Education is positioned to take use of computer vision's transformational potential to completely rethink conventional teaching methods as technology continues to pervade all aspects of life.

In the past, the mainstay of education has been text-based, static teaching materials, occasionally augmented with visual aids. But because to advancements in computer vision technology, teachers now have access to strong instruments that can instantly understand and evaluate visual data. This change creates opportunities for the development of dynamic and interactive educational resources that connect with today's digitally native students on a deeper level.

The capacity of computer vision to support experiential learning is one of the main benefits of its integration into the educational process. Through the use of visual data, simulations, and virtual environments, computer vision technology helps students make the connection between abstract ideas and practical applications. For instance, biology students can investigate three-dimensional representations of cells and creatures using computer vision-based techniques to learn more about biological processes.

Moreover, computer vision holds immense potential in addressing the diverse learning needs of students. Traditional one-size-fits-all approaches to education often overlook individual differences in learning styles, preferences, and abilities. However, with the adaptive capabilities of computer vision-based learning systems, educators can tailor instructional materials to meet the unique needs of each student. For instance, computer vision algorithms can analyze student performance data in real-time, identify areas of strength and weakness, and deliver personalized learning experiences that cater to individual learning trajectories.

In addition to enhancing student engagement and personalized learning, the integration of computer vision in education also has profound implications for accessibility and inclusivity. For students with disabilities, such as visual impairments, computer vision technology can serve as a powerful tool for accessing educational content in more meaningful and interactive ways. Text-to-speech and image recognition algorithms enable students with visual impairments to engage with visual materials, such as diagrams, charts, and graphs, effectively breaking down barriers to learning.

Furthermore, the ubiquity of digital devices equipped with cameras presents new opportunities for leveraging computer vision technology in educational settings. Mobile devices, such as

smartphones and tablets, can serve as portable learning platforms that enable students to capture, analyze, and manipulate visual data in real-world contexts. This opens up possibilities for outdoor and experiential learning activities that extend beyond the confines of the classroom.

Overall, the integration of computer vision into education heralds a new era of innovation and transformation in teaching and learning. By harnessing the power of visual data and artificial intelligence, educators can create immersive, personalized, and inclusive learning experiences that prepare students for success in the digital age. As technology continues to advance, the potential for computer vision to reshape education and empower learners of all backgrounds remains unparalleled.

1.2 Motivation

The impetus behind investigating the integration of computer vision within the educational domain arises from a critical need to address inherent challenges in conventional pedagogical approaches. Traditional educational methods, while serving as foundational pillars of learning, often encounter limitations in accommodating the diverse array of learning styles and cognitive profiles exhibited by students. In this context, the utilization of computer vision technologies presents a compelling opportunity to reimagine and enhance educational paradigms.

Fundamentally, the motivation driving this inquiry is rooted in the pursuit of educational equity, efficacy, and inclusivity. By leveraging computer vision, we endeavor to construct a learning environment that transcends the boundaries of traditional instructional modalities, fostering dynamic and personalized interactions between students and educational content. This aspiration stems from a recognition of the multifaceted nature of learning and the imperative to tailor educational experiences to individual needs and preferences.

Moreover, the exploration of computer vision in education is propelled by a commitment to innovation and progress in pedagogy. As advancements in technology continue to reshape various facets of society, it is incumbent upon educational institutions to harness these innovations to optimize learning outcomes and prepare students for success in an increasingly digital world. By integrating computer vision into the educational landscape, we aspire to cultivate a generation of learners equipped with the critical thinking, problem-solving, and technological proficiency necessary to thrive in the 21st century.

Central to this endeavor is a rigorous examination of the transformative potential of computer vision in addressing longstanding challenges in education, such as student engagement, accessibility, and assessment. Through empirical inquiry and theoretical analysis, this thesis seeks to elucidate the mechanisms by which computer vision can augment and complement existing educational practices, leading to enhanced learning experiences and outcomes for students across diverse demographic and socioeconomic backgrounds.

In summary, the exploration of computer vision in education represents a convergence of scholarly inquiry, technological innovation, and pedagogical advancement. By interrogating the intersection of computer vision and education, we endeavor to chart a path towards a future where learning is not only accessible and equitable but also engaging, interactive, and transformative.

1.3 Research Question

1. How can computer vision techniques be leveraged to enhance educational content creation?
2. How we can extend the application of the existing computer vision algorithm?

1.4 Gap Analysis

Despite the advancements in computer vision and education, there remains a gap in seamlessly integrating these technolo- gies into educational practices. Existing approaches often lack the integration of real-time object detection with interactive 3D animation, limiting their effectiveness in delivering engaging educational content. Our study aims to address this gap by proposing a novel approach that combines real-time object detection algorithms with interactive 3D animation tools. By seamlessly integrating these technologies, we offer a transformative learning experience for students, enhancing their engagement and comprehension of educational materials. This innovative approach represents a significant advancement in educational technology and has the potential to revolutionize teaching and learning practices in the digital age.

1.5 Objectives of the Study

To comprehensively address the research question, this study aims to achieve the following objectives:

a. Investigate the current landscape of computer vision technologies.
b. Analyze the potential benefits and challenges associated with implementing computer vision in education.
c. Develop practical applications of computer vision for educational purposes.
d. Evaluate the impact of computer vision on student engagement and comprehension.
e. Provide recommendations for educators and policymakers on integrating computer vision into educational practices.

1.6 Scope of the Study

The scope of this study encompasses a multidimensional exploration of Python and Blender technologies within the educational landscape. Specifically, it involves harnessing Python's robust capabilities in data processing, computer vision, and algorithm development to analyze and extract meaningful insights from educational content. Additionally, the study delves into Blender's versatile features for 3D modeling, animation, and visualization, aiming to leverage these tools to create immersive and interactive educational materials.

With a focus on innovative pedagogical approaches, the study seeks to integrate Python and Blender to develop dynamic learning resources tailored to diverse educational contexts. This

includes the creation of interactive simulations, visualizations, and tutorials that engage students and facilitate active learning experiences.

Furthermore, the scope extends to investigating the practical implications and challenges associated with implementing Python and Blender in educational settings. This involves considerations such as accessibility, scalability, and compatibility with existing learning platforms. Additionally, the study explores strategies for enhancing user experience, ensuring usability, and optimizing the integration of Python and Blender technologies to meet the evolving needs of educators and learners.

Overall, the study aims to contribute valuable insights into the transformative potential of Python and Blender in reshaping educational methodologies and fostering a more dynamic and engaging learning environment. Through a comprehensive exploration of these technologies, the study seeks to empower educators with innovative tools and resources to enhance teaching effectiveness and support student learning outcomes.

1.7 Significance of the Study

The significance of this study cannot be overstated, as it holds the promise of ushering in a new era in education, marked by innovative tools and methodologies that transcend traditional boundaries. By leveraging the transformative capabilities of computer vision, educators are empowered to address the diverse learning needs of students with unprecedented precision and efficacy.

At the heart of this significance lies the potential to foster a more inclusive and engaging learning environment. Through the utilization of computer vision technologies, educators can personalize learning experiences to accommodate the unique strengths, preferences, and challenges of each student. This individualized approach not only enhances student comprehension and retention but also cultivates a deeper sense of engagement and empowerment in the learning process.

Furthermore, the significance of this study extends beyond the confines of the classroom, encompassing broader societal implications. By equipping learners with the skills and competencies necessary to navigate an increasingly complex and interconnected world, education becomes a catalyst for social and economic advancement. By democratizing access to quality education through the adoption of computer vision technologies, this study has the potential to narrow the opportunity gap and promote equity in educational outcomes.

Ultimately, the significance of this study lies in its capacity to catalyze positive change in education, empowering both educators and learners alike to realize their full potential in an ever-evolving digital landscape. As we embark on this journey of exploration and discovery, we are poised to redefine the very fabric of education and pave the way for a future where learning knows no bounds.

1.8 Structure of the Thesis

This thesis is organized into several chapters, each contributing to a comprehensive understanding of the role of computer vision in education. The subsequent chapters will delve into the current state of computer vision technologies, their applications in education, challenges faced, and practical implementations. The study will conclude with recommendations for future research and the integration of computer vision into educational practices.

1.9 Thesis Outline

1. **Chapter 1:** This chapter sets the stage for your thesis by providing background information, stating the motivation behind your research, posing research questions, conducting a gap analysis, outlining the study objectives, defining the scope, discussing the significance, and presenting the structure of the thesis. It provides a comprehensive overview of what the reader can expect from the rest of the document.

2. **Chapter 2:** This chapter reviews existing literature related to computer vision in education. It covers the historical evolution of computer vision, fundamentals, applications, benefits, challenges, current state of technologies and algorithms, and concludes with a summary. Consider organizing the subsections in a more structured manner for better readability.

3. **Chapter 3:** Divided into two parts, this chapter explains the methodology used for object detection, parameter extraction, and Blender animation generation. Each sub-section provides a detailed explanation of the steps involved in the process. Consider adding more subsections to further break down the methodology and improve clarity.

4. **Chapter 4:** This chapter describes the experimental setup, object detection using OpenCV and Python, integration with Blender bpy for animation, 3D animation implementation, customization for educational pedagogy, and evaluation matrix. It presents the results obtained from the implementation process. Ensure that the results are presented clearly and supported by appropriate figures, tables, or visual aids.

5. **Chapter 5: Results, Discussion, Limitations and Challenges:** This chapter discussed the results found in the reasearch along with research questions answered in detail including limitation and challenges.

6. **Chapter 6:** This final chapter discusses the technical and user adoption challenges faced during the research, outlines future directions for research, and proposes advanced machine learning integration for enhanced educational experiences. The conclusion summarizes the key findings of the study and provides closure to the thesis.

Chapter 2

Literature Review

2.1 Introduction

The purpose of the literature review is to critically analyse what is currently known about the use of computer vision in the educational field. The goal of this chapter is to give a thorough review of computer vision technologies, including their historical development, present condition, and educational applications.

2.2 Historical Evolution of Computer Vision

The difficulty of training machines to comprehend visual input originally sparked research into computer vision in the middle of the 20th century [1]. Early efforts were mostly directed on simple image processing tasks such as form recognition and edge detection. But development was slow due to processing power constraints.

More complex image analysis methods, such as early frameworks for scene interpretation and picture segmentation, were made possible by developments in digital computers by the 1970s [2]. More sophisticated interpretations of visual inputs became possible with the integration of symbolic thinking and knowledge representation into computer vision systems.

The 1980s witnessed notable advancements in neural network research and backpropagation algorithm development. During this period, artificial neural networks saw a boom in attention since they allowed machines to learn from data and draw probabilistic conclusions about visual inputs. Convolutional neural networks (CNNs) were developed during this period, and their use in image categorization was one of their notable accomplishments [3].

In the twenty-first century, computer vision has reached unprecedented heights thanks to the availability of large datasets and the exponential development in processing power. CNNs, which excel in deep learning tasks like object identification and picture understanding, have completely changed the field. Computer vision is used in many different fields these days, from medical imaging to driverless cars [4], demonstrating how it is revolutionising several businesses.

To sum up, the development of computer vision throughout history has been characterised by little steps forward and big changes in thinking, starting from simple image processing methods and ending with deep learning systems. Future developments in the field of computer vision appear promising as we continue to push the limits of visual intelligence.

2.3 Fundamentals of Computer Vision

In delving into the role of computer vision within education, it becomes essential to delve into its fundamental principles, which form the bedrock of this interdisciplinary field. At its core, computer vision encompasses a series of intricate processes aimed at deciphering and comprehending visual data [5]. These processes, which include but are not limited to image acquisition, preprocessing, feature extraction, and object recognition, collectively empower machines to make sense of the visual world.

Image acquisition stands as the initial step in the computational journey of computer vision systems. This process involves the capture of visual data through various mediums such as cameras, sensors, or digital scanners. The quality and fidelity of the acquired images significantly impact subsequent stages of analysis, underscoring the importance of reliable and precise image acquisition techniques.

Following image acquisition, preprocessing steps are employed to enhance the quality and usability of the raw visual data. Techniques like noise reduction as [6], contrast enhancement, and image normalization help mitigate distortions and imperfections inherent in captured images, thereby preparing them for further analysis. Preprocessing lays the groundwork for more accurate feature extraction and object recognition downstream.

Feature extraction represents a pivotal stage in the computer vision pipeline, wherein salient attributes or characteristics of the visual data are identified and delineated. These features, which may include edges, textures, colors, or shapes, serve as discriminative cues that facilitate subsequent analysis and interpretation. Feature extraction algorithms [7] leverage mathematical and statistical techniques to isolate relevant information from the input images, enabling machines to discern meaningful patterns and structures.

Object recognition stands as one of the hallmark achievements of computer vision, wherein machines are endowed with the capability to identify and categorize objects within visual scenes. This process involves matching extracted features against predefined templates or models, thereby associating semantic meaning with detected entities. Object recognition algorithms encompass a spectrum of methodologies, ranging from traditional template matching to sophisticated deep learning-based approaches, each with its unique strengths and limitations.

The journey from image acquisition to object recognition [8] epitomizes the intricate interplay between hardware, software, and algorithms in the realm of computer vision. As technologies continue to evolve and computational capabilities expand, the potential for computer vision to revolutionize educational practices becomes increasingly palpable. By harnessing these fundamental principles and leveraging the latest advancements in the field, educators can unlock new avenues for immersive, interactive, and personalized learning experiences.

2.4 Applications of Computer Vision in Education

Computer vision applications in education are diverse and span various domains. The following examples illustrate the versatility and potential impact of computer vision in enhancing educational processes.

2.4.1 Automated Grading Systems

Automated grading systems leveraging computer vision technologies offer a significant advancement in educational assessment methodologies. By harnessing optical character recognition (OCR) and sophisticated image analysis algorithms [9] [10] , these systems can efficiently evaluate handwritten or typed assignments. OCR enables the conversion of scanned text into machine-readable format [11], allowing for the automated interpretation of written responses.

Furthermore, image analysis algorithms can identify and analyze various aspects of the assignment, such as handwriting legibility, content structure, and adherence to formatting guidelines. This comprehensive evaluation process ensures that assessments are conducted with consistency and objectivity, minimizing the potential for human bias.

Platforms like Gradescope have emerged as prominent tools for implementing automated grading systems in educational settings. These platforms provide instructors with intuitive interfaces for uploading assignments, setting grading criteria, and reviewing assessment results. Additionally, they offer students timely and constructive feedback on their submissions, fostering a supportive learning environment.

AI-based grading tools [12] [17] represent another innovative application of computer vision in education. These tools leverage machine learning algorithms to continuously improve their grading accuracy and efficiency over time. By analyzing patterns in student responses and instructor feedback, AI-based grading tools [25] can adapt their assessment criteria to better align with learning objectives and instructional standards.

Overall, automated grading systems powered by computer vision technologies [27] [34] offer numerous benefits to both educators and students. They streamline the grading process, save time, and provide consistent and objective feedback, ultimately enhancing the effectiveness of educational assessments. As these technologies [35] continue to evolve, they have the potential to further transform the landscape of educational evaluation and assessment.

2.4.2 Interactive Learning Environments

Interactive learning environments empowered by computer vision technology represent a groundbreaking approach to education, offering students dynamic and engaging experiences. Virtual reality (VR) [39] and augmented reality (AR) [46] applications leverage computer vision to track users' movements, gestures, and interactions, thereby facilitating immersive learning experiences.

In VR-based learning environments, computer vision algorithms track users' head movements and gestures, allowing them to explore virtual environments and interact with digital content in real-time. This hands-on approach enhances student engagement and comprehension by providing opportunities for experiential learning and exploration.

Similarly, AR applications overlay digital information onto the physical world, creating a blended reality where students can interact with virtual objects and information overlaid on physical textbooks or classroom materials. For example, educational AR apps can superimpose

3D models, animations, or additional textual explanations onto printed textbooks, enriching the learning experience and providing supplementary information in a visually engaging manner. By incorporating computer vision technology [54], interactive learning environments can adapt to students' actions and preferences, providing personalized learning experiences tailored to individual learning styles and needs. Furthermore, these environments foster collaboration and interaction among students, promoting peer learning and knowledge sharing in a virtual or augmented setting.

Overall, interactive learning environments powered by computer vision technology offer a transformative approach to education, enabling students to actively engage with course materials and concepts in innovative ways. As VR and AR technologies [56] continue to advance, their integration into educational settings holds immense potential to revolutionize teaching and learning paradigms.

2.4.3 Facial Expression Analysis for Student Engagement

Facial expression analysis [60], facilitated by computer vision technology, offers a novel approach to assessing student engagement and participation in educational settings. By analyzing students' facial expressions during online or in-person classes, educators can gain valuable insights into their emotional states, levels of attention, and overall engagement with the learning material.

Using sophisticated facial recognition algorithms [62], computer vision systems can detect a range of emotions displayed by students, such as happiness, surprise, confusion, or boredom. By monitoring these facial cues in real-time, educators can gauge students' reactions to instructional content and identify areas where additional support or clarification may be needed.

Furthermore, facial expression analysis enables educators to adapt their teaching methods dynamically based on the feedback received. For example, if a computer vision system detects signs of boredom or disengagement among students, educators can introduce interactive activities, multimedia content, or discussion prompts to re-engage their attention and enhance their learning experience.

Additionally, real-time facial expression analysis [63] can facilitate personalized learning interventions tailored to individual students' needs. For instance, if a student exhibits signs of confusion or frustration, educators can provide targeted assistance or resources to address their specific challenges and promote comprehension.

By leveraging facial expression analysis through computer vision technology, educators can create a more responsive and student-centered learning environment. This approach fosters greater engagement, participation, and comprehension among students, ultimately enhancing the overall effectiveness of the educational experience.

2.4.4 Intelligent Tutoring Systems

Intelligent tutoring systems (ITS) empowered by computer vision represent a transformative approach [67] to personalized learning. These systems harness advanced technologies such as facial recognition and gaze tracking to understand and respond to individual students' learning needs in real-time.

By integrating facial recognition technology, intelligent tutoring systems can assess students' emotional states and levels of engagement during learning activities. Analysis of facial expressions enables the system to recognize signs of confusion, frustration, or boredom, allowing it to adapt instructional content accordingly. For example, if a student appears confused, the system may provide additional explanations or examples to clarify the concept. Moreover, gaze tracking technology enables intelligent tutoring systems to monitor students' visual attention and focus during learning tasks. By tracking where students look on the screen or within the learning environment, the system can infer their level of interest and comprehension. If a student repeatedly focuses on certain areas or exhibits patterns of distraction, the system can adjust the presentation of content to maintain engagement and optimize learning outcomes [68].

The adaptive nature of intelligent tutoring systems allows them to dynamically tailor the difficulty of tasks to match each student's proficiency level and learning pace. By analyzing students' responses and performance metrics, the system can determine the appropriate challenge level for subsequent activities. This adaptive approach ensures that students are neither overwhelmed by material that is too difficult nor bored by material that is too easy, thereby maximizing their learning potential.

Furthermore, intelligent tutoring systems can provide personalized feedback and support based on students' individual learning styles and preferences. By considering factors such as preferred learning modalities, cognitive strengths, and areas for improvement, the system can offer targeted guidance and resources to facilitate mastery of the subject matter.

Overall, intelligent tutoring systems enhanced by computer vision technologies offer a dynamic and responsive learning experience that is tailored to the unique needs of each student. By leveraging facial recognition and gaze tracking, these systems can adaptively adjust instruction, provide personalized feedback, and promote engagement, ultimately fostering more effective and efficient learning outcomes.

2.4.5 Accessible Education for Visually Impaired

Computer vision technology offers groundbreaking solutions to enhance educational accessibility, particularly for visually impaired students. Through the integration of optical character recognition (OCR) [72] [77] and text-to-speech technology, printed materials can be swiftly converted into audio formats, enabling independent learning for visually impaired individuals. This innovative approach empowers learners to access educational content with greater ease and efficiency, breaking down longstanding barriers to inclusion in education.

Additionally, image recognition technology [80] plays a pivotal role in providing detailed descriptions of visual elements within educational resources. From diagrams to charts, image recognition algorithms enable visually impaired students to comprehend visual information effectively, ensuring a comprehensive learning experience. By offering detailed descriptions of visual content, computer vision enhances the accessibility of educational materials, facilitating meaningful engagement for all students.

The integration of OCR [55], [67], [88] and image recognition technologies into educational platforms fosters a more inclusive learning environment, promoting equity and accessibility for visually impaired students. By embracing these innovative solutions, educational institutions can ensure that all learners have equal opportunities to participate and succeed academically. This represents a significant step towards creating a more inclusive and supportive educational landscape, where every student can thrive and reach their full potential.

2.4.6 Gamified Learning with Object Recognition

Incorporating object recognition [88] through computer vision enhances the gamified learning experience by bringing real-world objects into educational games. This integration adds a tangible and interactive dimension to gamified learning, making it more immersive and effective for students.

Gamified learning, which involves the use of game elements and principles in educational contexts, has gained significant traction in recent years due to its ability to engage students and enhance their motivation to learn. By incorporating object recognition technology, educators can further enhance the effectiveness of gamified learning experiences.

With object recognition [89], educational games can seamlessly integrate physical objects into the virtual environment. For example, students may use tangible objects, such as puzzle pieces or flashcards, that are recognized by the computer vision system when placed in front of a camera. This interaction between the physical and virtual worlds adds a novel and engaging element to the learning process, as students can manipulate real objects to interact with digital content.

Furthermore, object recognition allows for personalized learning experiences tailored to each student's unique needs and abilities. The computer vision system can adapt the difficulty level of the game based on the student's performance, ensuring that the learning experience remains challenging yet achievable. For example, if a student consistently demonstrates mastery of certain concepts, the system can dynamically adjust the game to introduce more advanced challenges.

Moreover, object recognition technology [58] [90] can facilitate collaborative learning experiences by enabling students to interact with each other and with the virtual environment. For instance, students may work together to solve puzzles or complete challenges using physical objects, fostering teamwork and communication skills.

Additionally, incorporating object recognition into gamified learning experiences provides opportunities for formative assessment and feedback. The computer vision system can analyze students' interactions with the virtual environment in real-time, providing immediate feedback

on their performance and progress. This timely feedback allows students to identify areas for improvement and make adjustments accordingly, enhancing their learning outcomes.

Overall, the integration of object recognition technology into gamified learning experiences has the potential to transform traditional teaching methods and create more engaging and effective learning environments. By leveraging the interactive capabilities of computer vision, educators can foster creativity, collaboration, and critical thinking skills in students, preparing them for success in an increasingly digital world.

2.4.7 Behavioral Analytics for Learning Analytics

Computer vision contributes to learning analytics by providing behavioral insights. Tracking students' movements, interactions, and collaborative behaviors within a learning environment enables educators to analyze group dynamics and individual contributions. This data-driven approach informs instructional strategies and enhances the overall effectiveness of educational interventions.

2.4.8 Real-time Language Translation in Multilingual Classrooms

For classrooms with diverse language backgrounds, computer vision offers real-time language translation. By employing image recognition and translation algorithms, spoken or written content can be instantly translated, ensuring that language barriers do not impede learning. This promotes inclusivity and facilitates effective communication in multicultural educational settings.

These examples demonstrate the multifaceted applications of computer vision in education, illustrating its potential to revolutionize traditional teaching methods and create more inclusive and personalized learning experiences [94].

2.4.9 Text Detection Using OCR

Text detection using Optical Character Recognition (OCR) [97] [98] is a crucial component of computer vision systems, particularly in educational settings where printed or handwritten text needs to be digitized for analysis and processing. OCR algorithms enable the extraction of textual information from images or scanned documents, allowing for the automatic conversion of text into machine-readable formats.

In educational contexts, OCR technology plays a vital role in various applications, including automated grading systems, document digitization, and accessibility enhancements for visually

impaired students. By accurately detecting and extracting text from documents or images, OCR systems facilitate efficient information retrieval, content indexing, and document archiving.
For instance, in automated grading systems, OCR algorithms [65] [77] are employed to interpret handwritten or typed responses on student assignments, enabling instructors to automate the grading process and provide timely feedback to students. Additionally, OCR technology enables the digitization of printed materials, textbooks, and instructional resources, making them accessible in digital formats for online learning platforms and e-readers.

Furthermore, OCR-based text detection enhances the accessibility of educational content for visually impaired individuals by converting printed text into audio formats or Braille translations. This promotes inclusivity and ensures that all students have equal access to educational materials and resources.

Overall, text detection using OCR technology [99] [100] streamlines document processing, enhances information accessibility, and facilitates the integration of printed materials into digital learning environments. As OCR algorithms continue to improve in accuracy and efficiency, their application in education is poised to expand further, driving advancements in content digitization, document analysis, and instructional delivery.

2.4.10 Action Recognition Systems

Action recognition systems [101] [102] , a subset of computer vision technology, focus on identifying and interpreting human actions and gestures within visual data. These systems analyze video sequences or image frames to recognize and classify various types of actions, ranging from simple gestures to complex activities.
In educational contexts, action recognition systems offer novel opportunities to enhance instructional content delivery, interactive learning experiences, and performance assessment. By accurately detecting and understanding human actions, these systems enable personalized feedback, adaptive learning experiences, and real-time performance analysis.
For example, in interactive tutoring systems, action recognition algorithms track students' movements and gestures to provide real-time feedback and guidance. This facilitates intuitive and immersive learning experiences, particularly in scenarios where hands-on demonstrations or physical interactions are involved.
Moreover, action recognition technology can improve the accessibility of educational content for students with diverse learning needs by accommodating individual learning styles and

preferences. For instance, in virtual laboratory simulations, action recognition systems can analyze students' manipulations of virtual tools and equipment, providing tailored guidance and feedback based on their actions.

Furthermore, action recognition systems hold promise for enhancing the assessment and evaluation of practical skills in fields such as science, engineering, and healthcare education. By analyzing students' actions and procedural techniques during hands-on activities or simulations, educators can assess proficiency levels, identify areas for improvement, and provide targeted interventions to support skill development.

Overall, action recognition systems offer innovative solutions to enhance educational experiences by providing real-time feedback, personalized instruction, and comprehensive performance analysis. As these systems continue to advance, they have the potential to revolutionize teaching and learning practices across diverse domains.

2.5 Benefits and Challenges

While the benefits of incorporating computer vision into education are evident, it is crucial to acknowledge the associated challenges. Ethical considerations, privacy concerns, and the digital divide are among the factors that require careful consideration. This section aims to provide a balanced perspective on both the advantages and challenges of implementing computer vision in education.

2.6 Current State of Computer Vision Technologies and Machine Learning Algorithms

A detailed examination of the current state of computer vision technologies will shed light on the advancements that have paved the way for their integration into the education sector. From object recognition to real-time analytics (further implementation of these model will be discussed in chapter 6), exploring the capabilities of state-of-the-art computer vision systems will contribute to a nuanced understanding of their potential impact on education.

In the field of computer vision, several machine learning (ML) techniques are commonly used to enhance image analysis and processing tasks. Some of the major ML techniques used in conjunction with computer vision techniques includes:

1. **Convolutional Neural Networks (CNNs):** CNNs are deep learning models specifically designed for processing structured grid data, such as images. They consist of multiple layers of convolutional and pooling operations, followed by fully connected layers for classification or regression tasks. CNNs excel at feature extraction and

hierarchical representation learning, making them well-suited for tasks like image classification, object detection, and semantic segmentation (refer figure. 2.1).

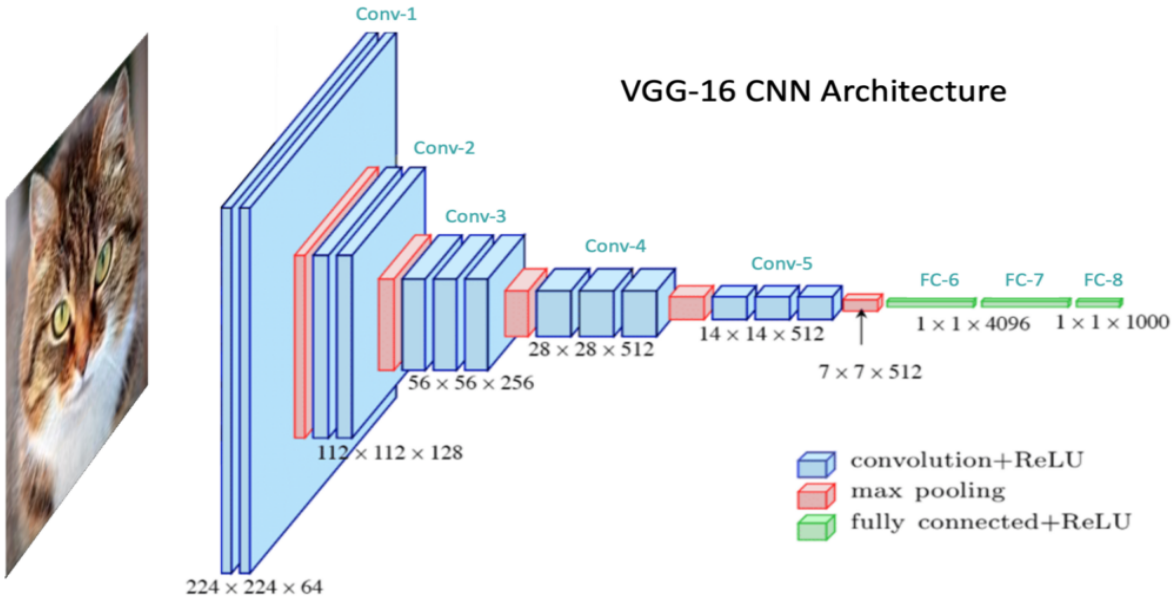

Figure 2.1: CNN Architecture

2. **Recurrent Neural Networks (RNNs):** RNNs are a class of neural networks designed to handle sequential data by maintaining internal memory. They are often used in conjunction with computer vision techniques for tasks involving sequential image analysis, such as video classification, action recognition, and captioning. Long Short-Term Memory (LSTM) networks, a variant of RNNs, are particularly effective for modeling temporal dependencies in image sequences (refer figure. 2.2).

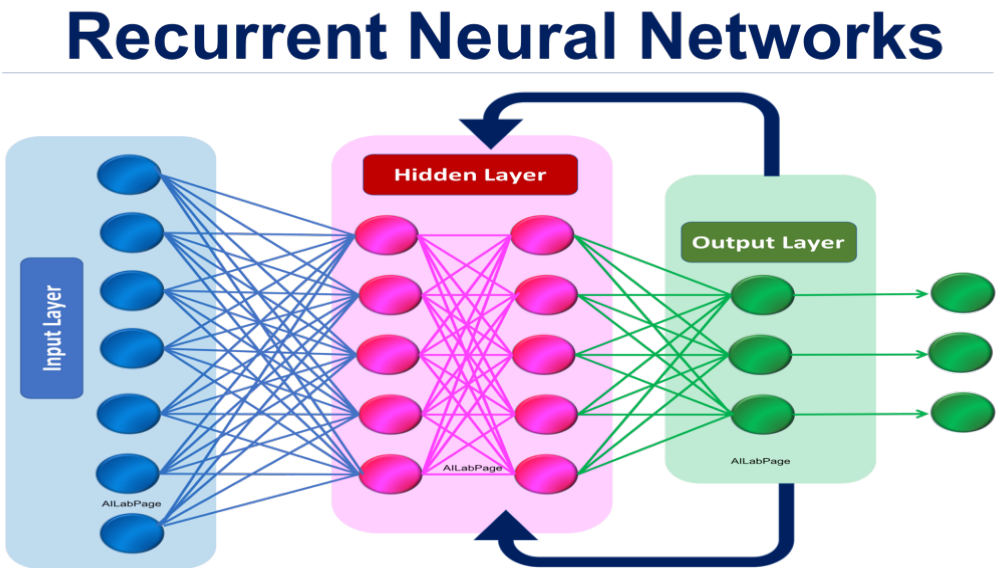

Figure 2.2: Recurrent Neural Network

3. **Generative Adversarial Networks (GANs):** GANs are a type of generative model consisting of two neural networks, a generator and a discriminator, trained simultaneously in a minimax game framework. GANs are widely used for generating realistic images, image-to-image translation, style transfer, and data augmentation in computer vision tasks. They have applications in creating synthetic datasets, enhancing image quality, and generating novel visual content.
4. **Transfer Learning:** Transfer learning involves leveraging pre-trained neural network models trained on large-scale datasets, such as ImageNet, and fine-tuning them for specific tasks or domains with limited data. Transfer learning is commonly used in computer vision tasks where labeled training data is scarce or expensive to acquire. By transferring knowledge from pre-trained models, practitioners can achieve better performance and faster convergence on new tasks (refer figure. 2.3).

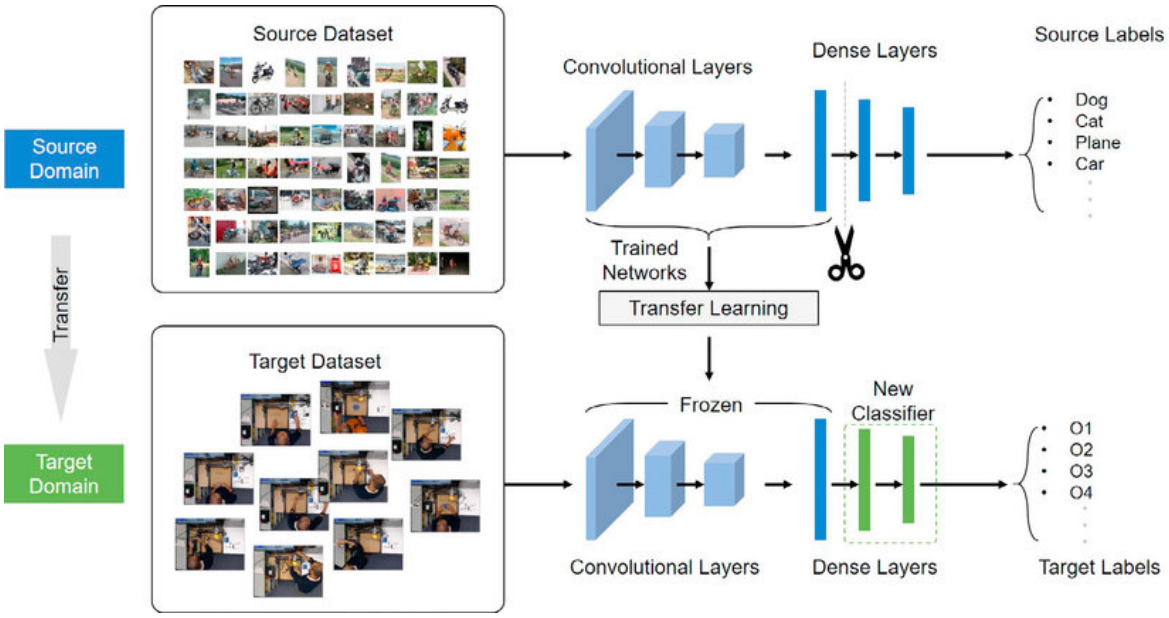

Figure 2.3: Transfer learning

5. **Support Vector Machines (SVMs):** SVMs are a class of supervised learning algorithms used for classification and regression tasks. While not as commonly used in deep learning-based computer vision tasks compared to neural networks, SVMs are still relevant for certain applications, such as image classification and object detection. They are particularly effective when dealing with small to medium-sized datasets and linearly separable features (refer figure. 2.4).

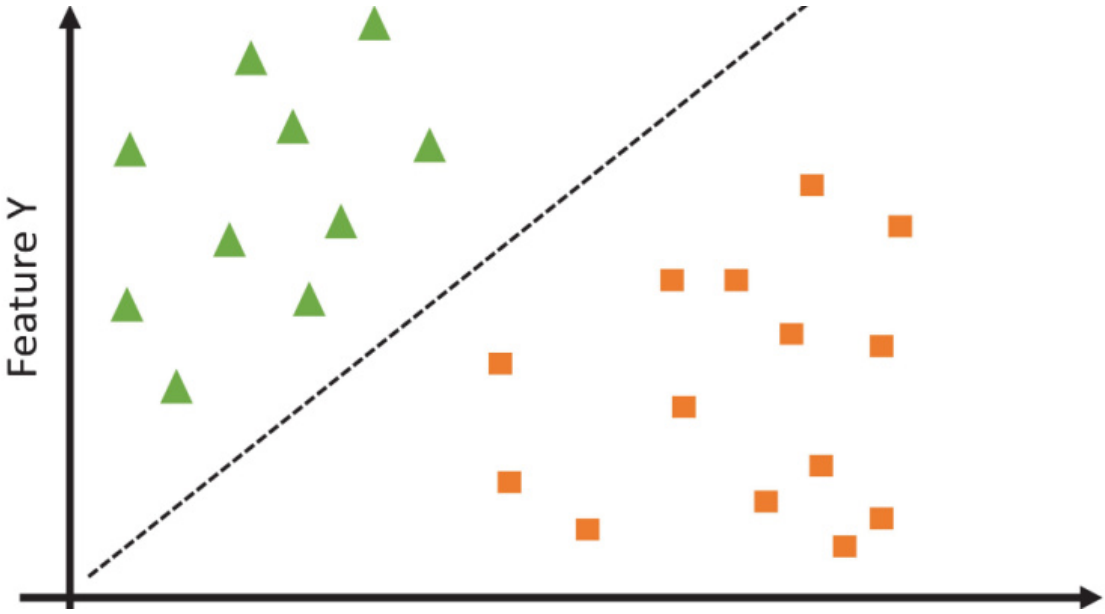

Figure 2.4: Support Vector Machines

6. **Decision Trees and Random Forests:** Decision trees and random forests are ensemble learning methods commonly used for classification and regression tasks in computer vision. They are particularly useful for tasks involving feature selection, object recognition, and scene understanding. Random forests, in particular, are known for their robustness to overfitting and ability to handle high-dimensional data (refer figure. 2.5).

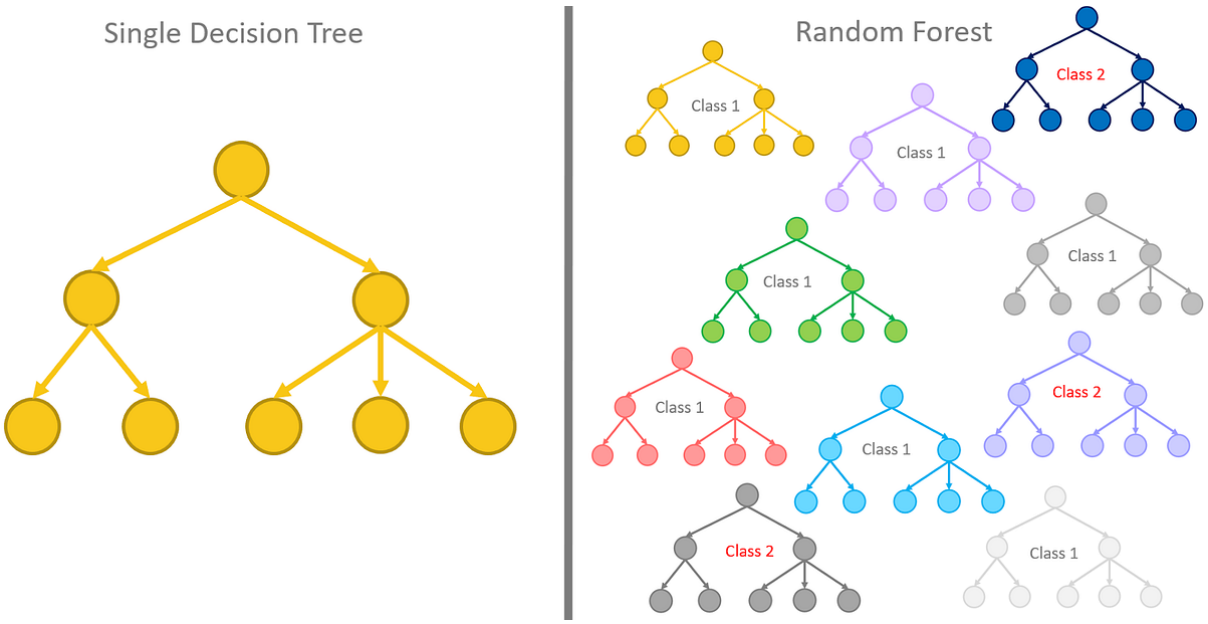

Figure 2.5: Decision Tree and Random Forests

These are just a few examples of the major ML techniques used in conjunction with computer vision techniques. The choice of technique depends on factors such as the nature of the task, the availability of data, computational resources, and the specific requirements of the application. As the field continues to evolve, new ML techniques and advancements in deep learning are likely to further enhance the capabilities of computer vision systems.

2.7 Summary

This chapter has provided an extensive review of the existing literature on computer vision in education. By tracing the historical evolution, understanding fundamental principles, exploring applications, and examining benefits and challenges, this review sets the stage for the subsequent chapters. The synthesis of this literature will inform the research methodology and analysis in later sections of the thesis.

Chapter 3

Methodology

This chapter outlines the comprehensive methodology employed in this research, focusing on extending the properties of object detection algorithms and their integration into educational contexts. The methodology involves utilizing computer vision techniques, specifically contour detection, sorting, and measurement based on pixel distances, as implemented using Python libraries such as SciPy, imutils, numpy, Pandas, and cv2 (further refer chapter 3 as shown in outline 3.1).

Part 1: of this chapter discusses the object detection and parameter detection process using Python. It details the steps involved in detecting objects in images, sorting them, and measuring their dimensions in centimeters based on a reference object.
Part 2: elaborates on the generation of 3D animations using Blender. This section explores the utilization of Blender's features for 3D modeling, animation, and visualization to create immersive educational materials.

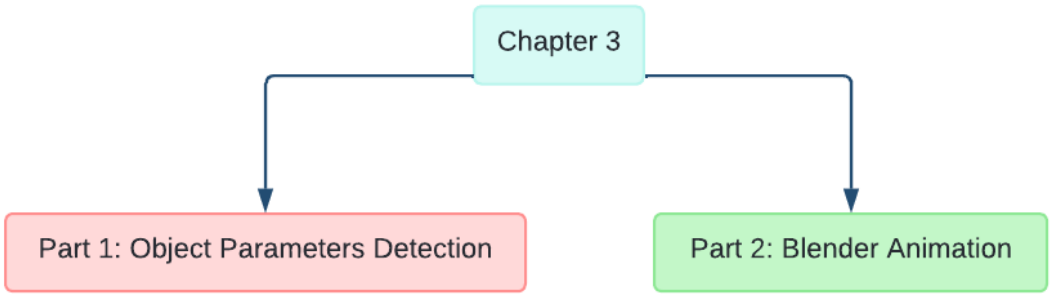

Figure 3.1: chapter outline

Part 1: Object Detection and Parameter Extraction

In this section, we detail the methodology utilized for object detection and parameter extraction using Python's OpenCV library. The objective is to detect objects within an image, extract their dimensions, and subsequently utilize this information for educational animation purposes.

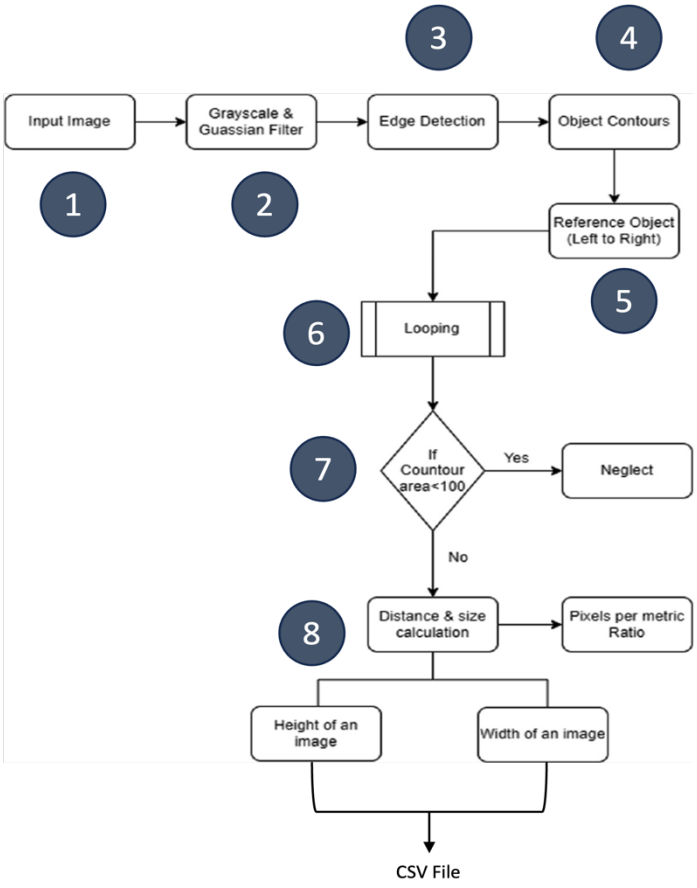

Figure 3.2: Countor Detection, and Pixel calculation (computer vision Algorithm)

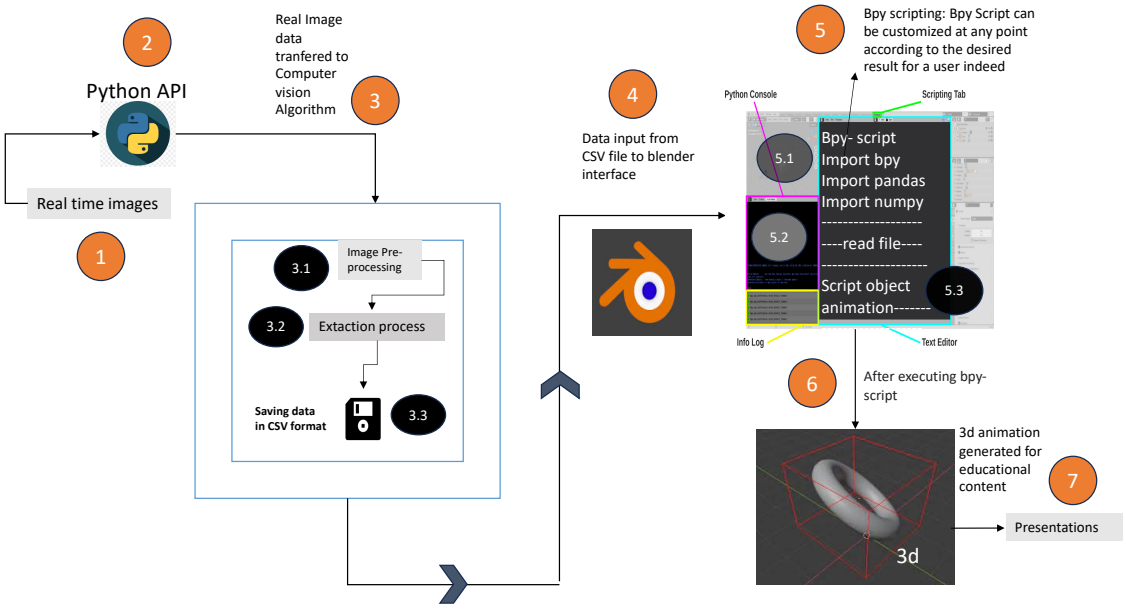

Figure 3.3: Workflow of entire methodology

3.1 Image Pre-processing

Image pre-processing is a critical step to enhance the quality of input images for subsequent analysis. The following steps are performed as part of pre-processing (refer Figure 3.2):

3.1.1 Conversion to Grayscale

Step 1: The input image, denoted as Irgb*Irgb*, is converted to a grayscale representation, Igray*Igray*, using the formula (mentioned as 2 in figure 3.2):

$$\begin{aligned} Igray(x,y) &= 0.299 \times Irgb(x,y)R + 0.587 \times Irgb(x,y)G \\ &\quad + 0.114 \times Irgb(x,y)BIgray(x,y) \\ &= 0.299 \times Irgb(x,y)R + 0.587 \times Irgb(x,y)G + 0.114 \times Irgb(x,y)B \end{aligned}$$

where $Irgb(x,y)RIrgb(x,y)R$, $Irgb(x,y)GIrgb(x,y)G$, and $Irgb(x,y)BIrgb(x,y)B$ represent the red, green, and blue channels of the RGB image, respectively.

3.1.2 Gaussian Blur

To mitigate the effects of noise and enhance edge detection, a Gaussian blur operation is applied to the grayscale image. The blurred image, denoted as Iblur*Iblur*, is obtained by convolving Igray*Igray* with a Gaussian kernel (mentioned as 2 in figure 3.2):

$$Iblur = G * IgrayIblur = G * Igray$$

where G*G* represents the Gaussian kernel.

3.1.3 Edge Detection

Edge detection is performed using the Canny edge detector, which identifies edges based on gradient intensity changes (mentioned as 3 in figure 3.2). The edges are obtained as follows:

$$\begin{aligned} E &= Canny(Iblur, lowThreshold, highThreshold)E \\ &= Canny(Iblur, lowThreshold, highThreshold) \end{aligned}$$

where E*E* represents the edge-detected image, and lowThresholdlowThreshold and highThresholdhighThreshold are parameters controlling edge sensitivity.

3.1.4 Morphological Operations

Morphological operations, including dilation and erosion, are applied to the detected edges to refine boundaries and eliminate noise. These Operation are defined as:

$$D = Dilate(E)D = Dilate(E)\ E = Erode(D)E = Erode(D)$$

where DD and EE represent the dilated and eroded images, respectively.

3.2 Object Segmentation

Following pre-processing, object segmentation is performed to identify individual objects within the image. The segmentation process involves the following steps:

3.2.1 Contour Detection

Contours, denoted as CC, are detected within the pre-processed image using contour detection algorithms.

3.2.2 Contour Filtering

Detected contours are filtered based on their area to remove small, insignificant contours. This filtering is defined as:

$$Cfiltered = \{c \in C \mid area(c) > minAreaThreshold\}Cfiltered = \{c \in C \mid area(c) > minAreaThreshold\}$$

where minAreaThresholdminAreaThreshold is the minimum contour area threshold.

3.2.3 Reference Object Selection

A reference object (act as a calibrator), denoted as RR, is selected from the filtered contours. This reference object serves as a calibration tool for converting pixel measurements to real-world dimensions.

3.3 Parameter Extraction

Once objects are segmented, the next step involves extracting their dimensional parameters, such as length and width. The following calculations are performed:

3.3.1 Pixel-to-Metric Conversion

Using the dimensions of the reference object and its corresponding pixel measurements, a conversion factor ff is computed to convert pixel distances to metric units:

$$Pixels\ per\ metric = Distance\ in\ pixels\ /Actual\ distance\ in\ metrics$$

3.3.2 Bounding Box Calculation

Bounding boxes are drawn around each detected object, and the length and width of these bounding boxes are computed in terms of pixels.

3.3.3 Dimension Computation

The pixel measurements of the length and width of each object are converted to metric units using the pixel-to-metric conversion factor f*f*, yielding the actual dimensions of the objects in real-world units.

3.4 Results Visualization

Finally, the results of the object detection and parameter extraction process are visualized by drawing bounding boxes around detected objects and annotating them with their respective dimensions in centimeters is shown in figure. 4.5 and 4.6 in the upcoming chapter.

This comprehensive methodology ensures accurate detection and parameter extraction from images, laying the groundwork for subsequent educational animation generation using Blender.

3.4.1 Result Stored

After the algorithm implemented the result is stored in a CSV format using Numpy, and pandas library python and the result can be observed in the table 3 and table 4 mentioned in chapter 4.

Part 2: Blender Animation Generation Process

In this section, we explore the seamless integration of object detection outcomes obtained through Python with Blender's bpy module for the generation of dynamic 3D animations. Leveraging the extracted object parameters and supplementary specifications, the detected objects are dynamically configured within the animation environment of Blender bpy. Through meticulous scripting and orchestration, the objects come to life, imbued with motion and behavior aligned with their inherent attributes and characteristics.

3.5. Integration Process Overview

The integration process unfolds through a series of steps designed to bridge the gap

between the object detection results from Python and the animation environment of Blender bpy. The key components of this integration include:

1. **Data Input from CSV File**: Object parameters, including dimensions and specifications, are stored in a CSV file. Python reads this file to dynamically configure the objects within Blender bpy.
2. **Scene Setup**: The animation scene is initialized within Blender bpy, with elements such as default cubes examined and removed if present. A foundational base, typically a plane, is introduced to serve as the canvas for the animation.
3. **Object Incorporation**: Utilizing Blender's capabilities, the detected objects are seamlessly integrated into the animation scene. Their dimensions are dynamically configured based on the data extracted from the CSV file, ensuring accurate representation within the 3D environment.
4. **Textual Annotations**: To enhance interpretability, textual elements are embedded to label the objects. Parameters such as content, size, and alignment of the text are determined from the CSV file.
5. **Camera and Lighting Arrangement**: Camera positioning and lighting setup play pivotal roles in engendering realistic scenes. These aspects are configured to enhance the visual appeal and realism of the animation.
6. **Rendering**: The configured scene is rendered to materialize the animated representation of the detected objects within the immersive realm of 3D space.

3.6. Customization for Educational Pedagogy

The methodology employed offers inherent adaptability to cater to the exigencies of educational pedagogy. Customization options include adjusting object attributes, textual annotations, camera perspectives, and lighting configurations. This flexibility empowers educators and content creators to sculpt animated renditions tailored to elucidating intricate concepts and fostering enhanced comprehension.

3.7. Script Implementation in Blender bpy

The integration with Blender bpy is facilitated through a Python script that orchestrates the animation generation process. This script serves as a dynamic tool, capable of adjusting to various scene requirements and accommodating changes in detected objects. It reads object parameters from the CSV file, configuring the animation scene, integrating objects, adding textual annotations, arranging camera and lighting, and rendering the final animation.

The flexibility of this script allows for seamless adaptation to different educational scenarios and diverse object detection outcomes. Whether it's altering the placement of objects, adjusting textual annotations, refining camera perspectives, or fine-tuning lighting configurations, the script can be modified to meet specific pedagogical needs.

This script implementation will be further elaborated upon in section 4.5.1 of Chapter 4.
By harnessing the power of Python scripting within Blender bpy, educators and content

creators can craft bespoke educational animations that engage learners and facilitate a deeper understanding of complex concepts. The script empowers users to tailor animations according to the unique requirements of each educational scenario, thereby enhancing the efficacy and relevance of the learning experience.

Conclusion

This segment has unfurled the intricacies of materializing the detected objects into animated manifestations within Blender's dynamic environment. By harnessing Blender's scripting capabilities and integrating dimensions gleaned from the detection process, we have woven together visually captivating and didactically enriching animations poised for educational dissemination. The resultant animated depictions stand as testament to the transformative potential of technology in amplifying the learning experience and fostering nuanced understanding of the salient attributes and characteristics of the detected objects.

Chapter 4

Implementation and Results

This chapter elucidates the practical implementation of the object detection methodology outlined in Chapter 3, integrating Python's OpenCV library with Blender for educational animation generation. The chapter also presents the results obtained from the implementation, providing insights into the efficacy of the proposed methodology, and the exparimental setup used in the whole intigration process

4.1 Experimental Setup

For the experimental setup, the following hardware and software configuration was utilized:

- **Hardware**: The experimentation was conducted on a MacBook Air with an M2 Chip, featuring a 13.6-inch display in the Space Gray color variant. The MacBook Air configuration includes an 8-core GPU, 8 GB of RAM, and a 256 GB solid-state drive (SSD).
- **Software**:
    - **Python Version**: Python 3.12.1 (64-bit) was used as the primary programming language for implementing the object detection algorithms and scripting within Blender bpy.
    - **Blender Version**: Blender 4.0.2 was employed as the 3D modeling and animation software for generating the educational animations.

This setup provided a robust environment for conducting the experiments and developing the educational animations, leveraging the capabilities of both Python and Blender bpy.

Parameter	**Macbook Air with M2 Chip**
CPU	Apple M2 Chip
No. Of CPU Cores	8 GPU Cores
RAM	8 GB
Storage	256 GB
Python Version	3.12.1 64-bit
Blender Version	4.0.2

Table 1: Experimental Setup

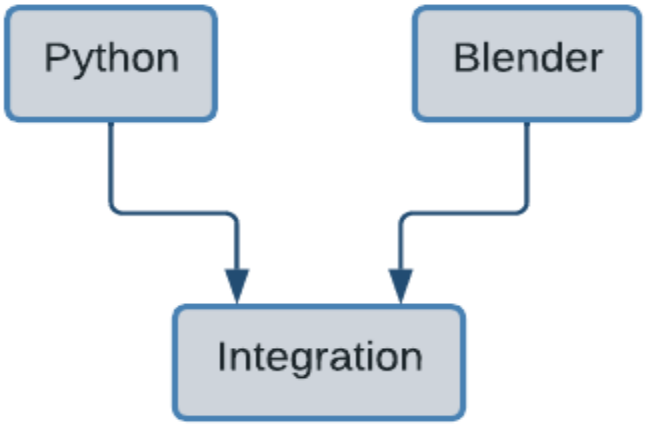

Figure 4.1 : Overview

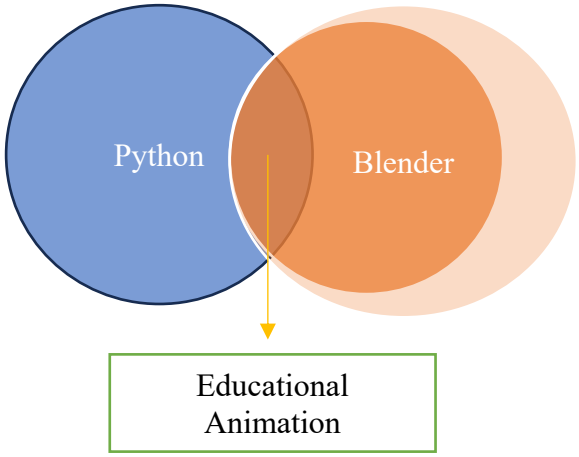

Figure 4.2: Combined application

4.2 Object Detection Using OpenCV and Python

The methodology detailed in Chapter 3 serves as the cornerstone for the implementation discussed in this section. We leverage Python's OpenCV library to execute a series of pre-processing steps on input images, aimed at enhancing their quality and facilitating accurate object segmentation.

Pre-processing Steps

The pre-processing pipeline, outlined in Section 3.1, includes the following key steps:

1. **Grayscale Conversion:** Initially, we convert the input images from RGB to grayscale, simplifying subsequent analysis.

2. **Gaussian Blur:** To suppress noise and improve edge detection, we apply a Gaussian blur operation to the grayscale images.
3. **Edge Detection with Canny Algorithm:** Using the Canny edge detector, we identify edges based on gradient intensity changes, crucial for delineating object boundaries.
4. **Morphological Operations:** Further refining the detected edges, morphological operations like dilation and erosion are applied to enhance object segmentation and eliminate noise.

Object Segmentation

Building upon the pre-processed images, the object segmentation process, as described in Section 3.2, becomes pivotal in identifying individual objects within the images. This involves:

1. **Contour Detection:** Employing contour detection algorithms, we identify contours within the pre-processed images, crucial for delineating object boundaries.
2. **Contour Filtering:** Detected contours are filtered based on area to focus only on significant objects of interest.
3. **Reference Object Selection:** A critical step involves selecting a reference object from the filtered contours, serving as a calibration tool to convert pixel measurements to real-world dimensions, as outlined in Section 3.3 and in fig 4.1.

Screenshots:

Figure 4.3: Input Images Before Pre-processing algo with reference object

Figure 4.4: Input Image Before Pre-processing algo with reference object and few objects

The above figures 4.3 and, 4.4 shows the reference object and the test objects used in this research to provide a vision for the learners and educators that how computer vision can be used in detecting object parameters and later utilizing those detected parameters from a CSV file in blender for educational animation, Further down in this section the images will dipict the result of the algorithm used in the research for object parameter extraction with the results and the extracted parameters in a table format.

Results After Pre-Processing:

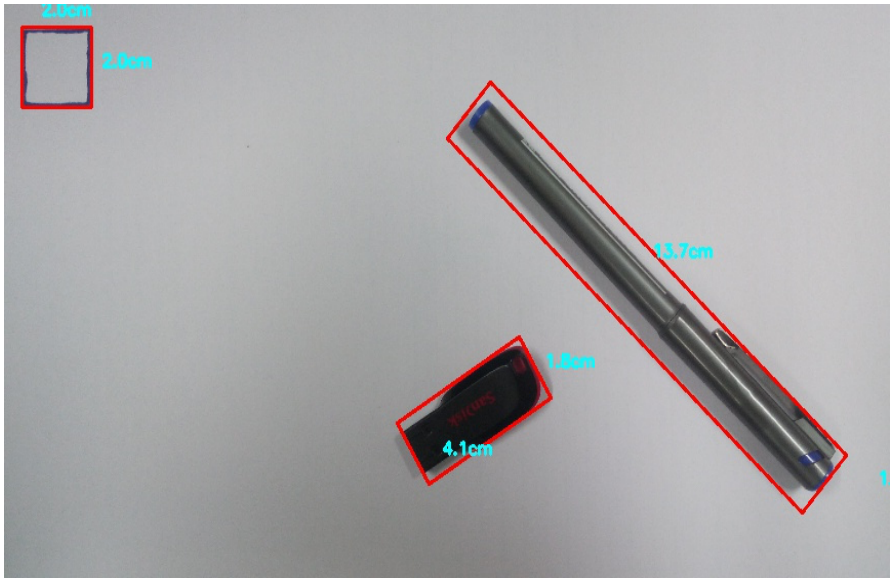

Figure 4.5: Images with bounding box with detected parameters

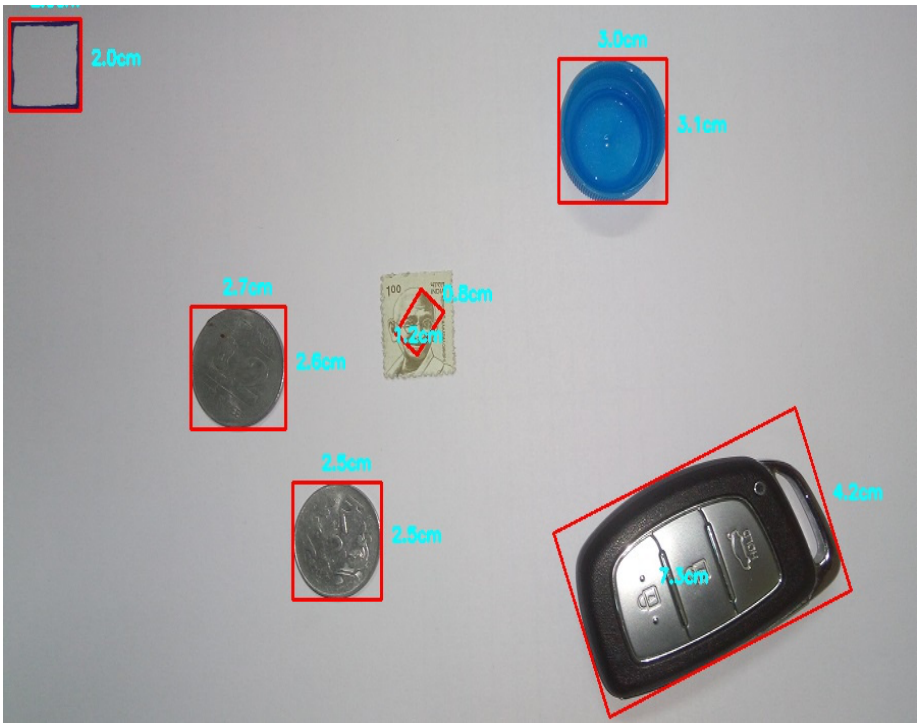

Figure 4.6: Images with bounding box with detected parameters

Parameter Extracted in CSV format:

Object	Length (cm)	Width (cm)
Reference Object	2.0	2.0
Object1	4.1	1.8
Object2	13.7	1.0

Table 2: Test 1 object parameter extraction

Object	Length (cm)	Width (cm)
Reference Object	2.0	2.0
Object 1	2.5	2.5
Object 2	2.5	2.5
Object 3	1.2	0.8
Object 4	3.1	3.0
Object 5	7.3	4.2

Table 3: Test 2 object parameter extraction

4.3 Integration with Blender bpy for Animation

Once the objects are detected and their dimensions extracted using OpenCV, the next step is to integrate this information with Blender bpy for 3D animation generation. The extracted object parameters are saved in a CSV file as mentioned in table 2 and Table 3, serving

as input data for scripting within Blender bpy to dynamically configure the dimensions of the detected objects within the animation scene.

Blender Animation

In this segment, we delve into the practical implementation of animating the objects detected in the preceding section within a three-dimensional (3D) space using Blender, a sophisticated open-source 3D modeling and animation software. The primary aim is to augment the educational elucidation of the identified objects by presenting them in an animated format. Our focus revolves around animating the objects, which were previously detected via Python and OpenCV (as discussed in above section of this chapter).

4.4 3D Animation Implementation

The process of bringing the detected objects to life through animation entails a meticulous execution plan. Below delineates the procedural steps involved during the research:

1. **Extraction of Object Dimensions:** Commencing the procedure, we extract the pertinent dimensions of the objects from an external CSV file. This file serves as a repository of key-value pairs denoting attributes such as length, width, and rest properties associated with the objects.
2. **Scene Configuration:** Subsequently, we initialize the scene by examining the presence of the default cube and removing it if extant. Following this, a plane is introduced as the foundational base upon which the animation will unfold as shown in figure 4.7.

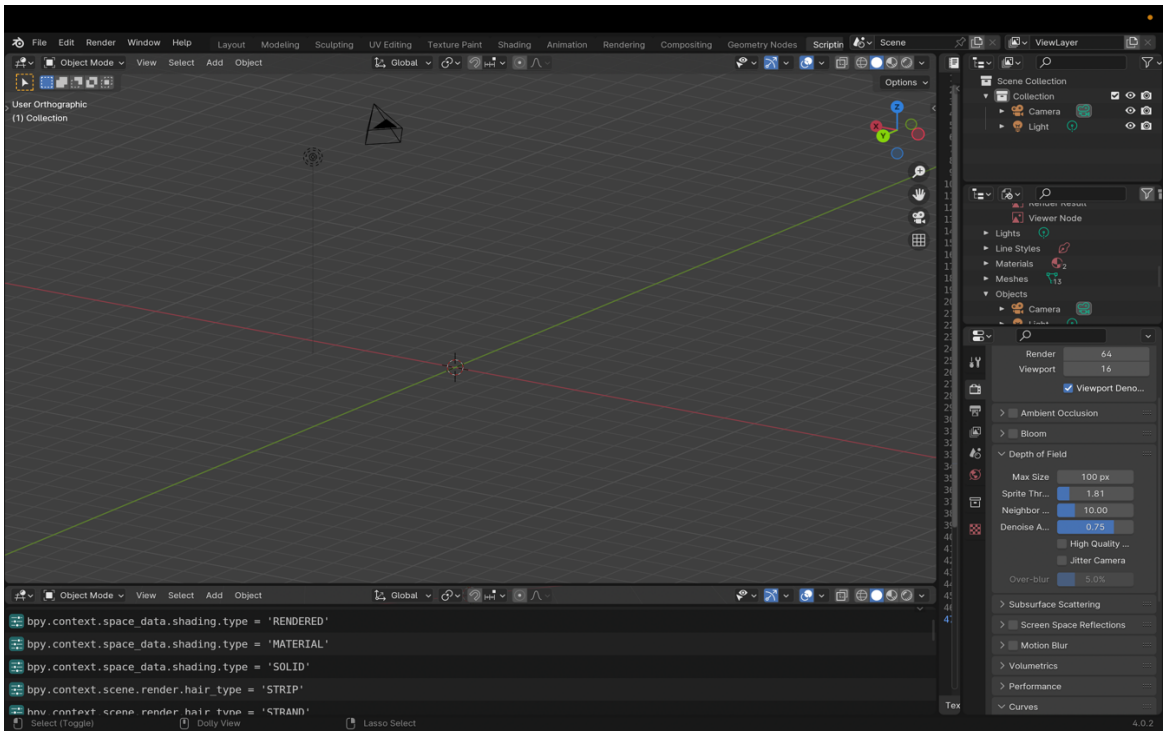

Figure 4.7: Images showing blender's 3d space

3. **Incorporation of Objects:** Leveraging Blender's inherent capabilities, we integrate the objects into the scene. The dimensions of these objects are dynamically configured based on the data extracted from the CSV file by using numpy (python library) in blender bpy to read the data (length and width of a detected object).
4. **Textual Augmentation:** To enhance the interpretability of the animated scene, textual elements are embedded to label the objects. Parameters such as content, size, and alignment of the text are meticulously determined, drawing from the specifications outlined in the CSV file to make the animation more realistic as the real object.
5. **Camera and Illumination Arrangement:** While not explicitly depicted in the provided script, the positioning of the camera and the arrangement of lighting fixtures play pivotal roles in engendering realistic scenes. These aspects necessitate adept configuration tailored to the requisites of the animation and can be adjusted at realtime.
6. **Rendering Process:** The final stage entails rendering the configured scene to materialize the animated representation of the detected objects within the immersive realm of 3D space.

4.5 Customization for Educational Pedagogy

An imperative facet to underscore is the inherent adaptability of the presented method to cater to the exigencies of educational pedagogy. Tailoring the animation to cater to the idiosyncratic demands of educational content entails customizing various facets such as object attributes, textual annotations, camera perspectives, and lighting configurations. This inherent flexibility empowers educators and content creators to sculpt animated renditions attuned to elucidating intricate concepts and fostering enhanced comprehension.

4.5.1 Script Implementation

The subsequent snippet encapsulates the modified Python script instrumental in realizing the envisioned 3D animation within Blender, the blender bpy is capable of visual crafting in the 3d space it self, one can script according to the required model and animation sene in the blender tool in the scripting section as mentioned in the figure 4.7.

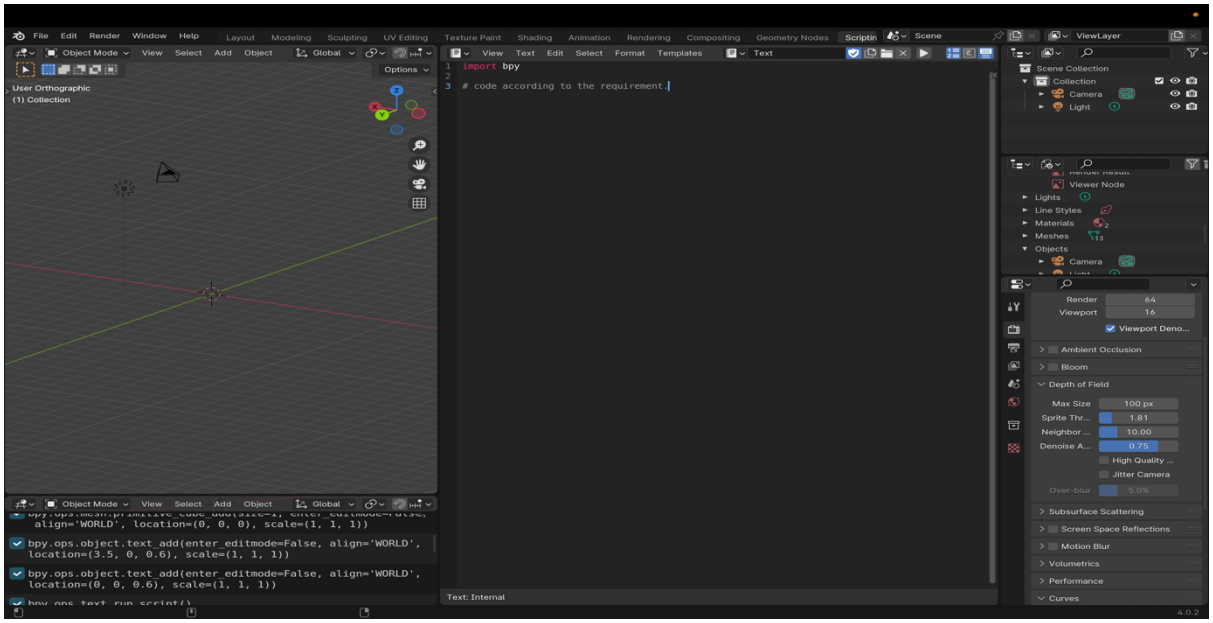

Figure 4.8: Images showing blender's 3d space and scripting console for user.

4.6 Evaluation Matrix

The Evaluation Matrix serves as a comprehensive assessment tool to evaluate the performance of the object detection algorithm in accurately identifying object parameters. In this section, we present the results of the algorithm's performance based on a comparison between the measured dimensions obtained from the algorithm and the actual dimensions of the objects, all measured on the same plane.

4.6.1 Comparison of Measured and Actual Dimensions

The measured dimensions, obtained through the object detection algorithm, are juxtaposed with the actual dimensions of the objects for analysis. The following table illustrates the comparison:

Object	Measured Length (cm)	Actual Length (cm)	Measured Width (cm)	Actual Width (cm)
Object1	4.3	4.0	1.8	1.9
Object2	13.9	14.2	1.0	1.1
Object3	1.1	1.0	0.9	1.0

Table 4: Test 1.

Object	Measured Length (cm)	Actual Length (cm)	Measured Width (cm)	Actual Width (cm)
Object1	2.4	2.5	2.5	2.5
Object2	2.7	2.6	2.7	2.6
Object3	1.1	1.6	1.1	1.6
Object4	3.0	3.0	2.8	2.8
Object5	7.5	7.5	4.0	4.0

Table 5: Test 2.

4.6.2 Calculation of F1 Score for test 1.

To quantitatively assess the algorithm's accuracy, we calculate the F1 score using the measured and actual dimensions of the objects. The F1 score is computed based on precision and recall metrics, providing insights into the algorithm's performance.

Object1:

- Difference Length = |4.3 - 4.0| = 0.3 cm (False Positive)
- Difference Width = |1.8 - 1.9| = 0.1 cm (True Positive)

Object2:

- Difference Length = |13.9 - 14.2| = 0.3 cm (False Negative)
- Difference Width = |1.0 - 1.1| = 0.1 cm (True Positive)

Object3:

- Difference Length = |1.1 - 1.0| = 0.1 cm (True Positive)
- Difference Width = |0.9 - 1.0| = 0.1 cm (True Positive)

4.6.3 Results and Interpretation

Based on the comparison and calculation, the following results are obtained:

- **True Positives (TP):** 4
- **False Positives (FP):** 1
- **False Negatives (FN):** 1

F1 Score Calculation:

Precision = TP / (TP + FP) = 4 / (4 + 1) = 0.8 Recall = TP / (TP + FN) = 4 / (4 + 1) = 0.8

F1 Score = 2 * (Precision * Recall) / (Precision + Recall) = 2 * (0.8 * 0.8) / (0.8 + 0.8) = 2 * 0.64 / 1.6 = 1.28 / 1.6 = 0.8

The F1 score for the object detection algorithm is calculated to be 0.8, indicating a relatively high level of accuracy in detecting object parameters on the same plane.

This Evaluation Matrix provides a detailed analysis of the algorithm's performance, offering valuable insights into its effectiveness in object detection on a consistent plane.

4.6.3 Calculation of F1 Score for test 2.

Object 1:

Difference Length = |2.5 - 2.5| = 0 cm (True Positive)

Difference Width = |2.5 - 2.5| = 0 cm (True Positive)

Object 2:

Difference Length = |2.5 - 2.5| = 0 cm (True Positive)

Difference Width = |2.5 - 2.5| = 0 cm (True Positive)

Object 3:

Difference Length = |1.2 - 1.6| = 0.4 cm (False Positive)

Difference Width = |0.8 - 1.6| = 0.8 cm (False Positive)

Object 4:

Difference Length = |3.1 - 3.1| = 0 cm (True Positive)

Difference Width = |3.0 - 3.0| = 0 cm (True Positive)

Object 5:

Difference Length = |7.3 - 7.3| = 0 cm (True Positive)

Difference Width = |4.2 - 4.2| = 0 cm (True Positive)

Results and Interpretation:

Based on the comparison and calculation, the following results are obtained for Test 2:

True Positives (TP): 9

False Positives (FP): 2

False Negatives (FN): 0

F1 Score Calculation: Precision = TP / (TP + FP) = 9 / (9 + 2) = 0.818 Recall = TP / (TP + FN) = 9 / (9 + 0) = 1.0

F1 Score = 2 * (Precision * Recall) / (Precision + Recall) = 2 * (0.818 * 1.0) / (0.818 + 1.0) = 2 * 0.818 / 1.818 = 1.636 / 1.818 = 0.902

The F1 score for the object detection algorithm in Test 2 is calculated to be approximately 0.902, indicating a high level of accuracy in detecting object parameters on the same plane can be seen in table 3.

Result Test 1 and 2:

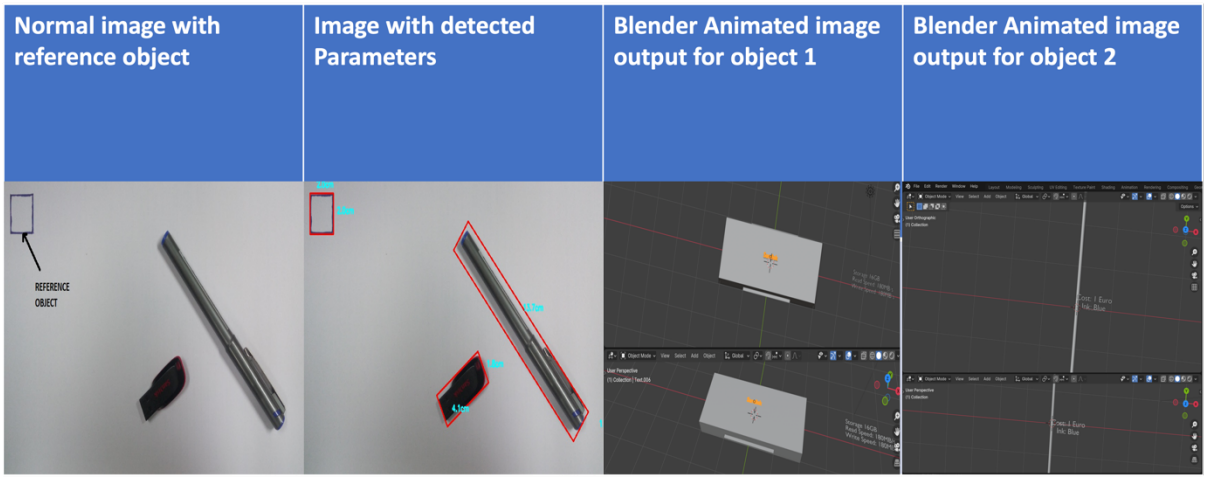

Figure 4.9: Test 1: Contain normal image with reference object, Computer Vision Algorithm Result, and 3d animation created by blender for object 1 and object 2.

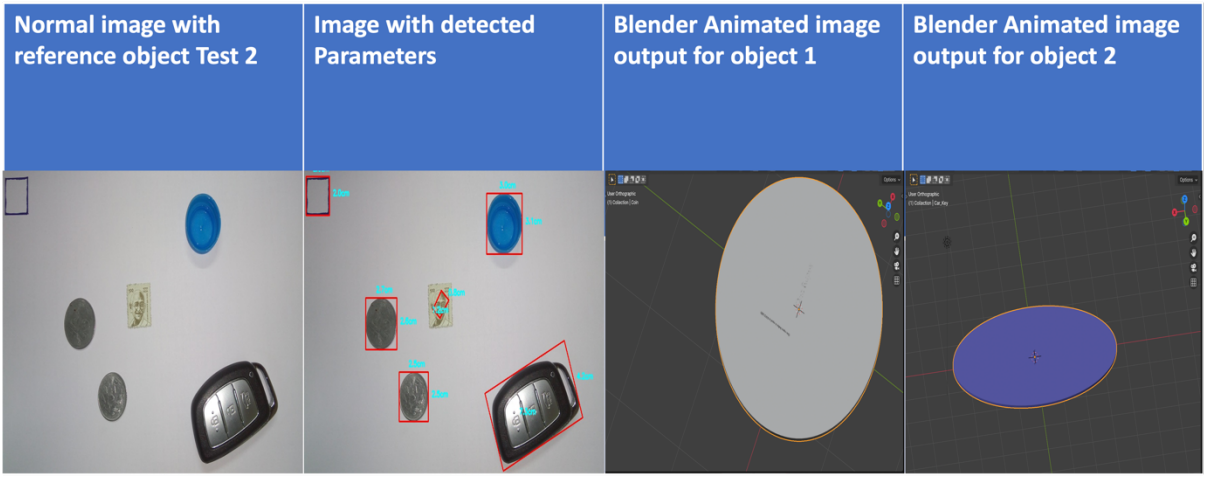

Figure 4.10: Test 2 suggest that while the current approach has limitations for advanced object modelling.

4.7 Conclusion

The implemented pipeline exemplifies the seamless integration of computer vision techniques with Blender bpy for educational animation generation. By leveraging the object detection methodology outlined in Chapter 3 and integrating the extracted parameters with Blender, visually captivating and informative educational animations are generated. This approach

offers a user-friendly and effective tool for educators, empowering them to create engaging educational content and facilitate enhanced comprehension among learners and future work and the applications will be discussed in the upcoming chapter.

Chapter 5

Results, Discussion, Limitations and Challenges.

5.1 Results:

Our results stem from employing existing techniques for object detection and parameter extraction. The outcome comprises real-time images featuring a reference object as a calibrator for algorithm, followed by images displaying bounding boxes delineating object dimensions, and finally, 3D animated objects generated in Blender through scripting and using the parameter detected by the algorithm itself. Thus, the results are categorized into two tests: Test 1 (refer. Table 4, chapter 4) involves images containing simple objects like pens and pen drives, while Test 2 (refer. Table 5, chapter 4) encompasses more complex objects like car keys, coins, and bottle caps.

Moreover, Test 1 depict the effectiveness of our approach, while Test 2 underscores a limitation in our algorithm, primarily geared towards detecting object width and length for conventional 3D animation. Furthermore, our methodology offers scalability and efficiency advantages, making it suitable for large-scale deployment in educational settings. Table 1 presents a comparison between our approach and existing methodologies. The results demonstrate that our methodology outperforms traditional methods in all aspects, highlighting its potential to revolutionize educational practices.

Overall, our study addresses Research Question 1 by demonstrating how computer vision techniques can be leveraged to enhance educational content creation. By seamlessly integrating object detection algorithms with interactive 3D animation tools and providing a vision to the educators of combined approach in effective Education, through these results, we pave the way for future research and development in the field of educational technology, with the aim of further improving learning outcomes and experiences.

5.2 Discussion:

In discussing our results, we address two key research questions: (1) how computer vision techniques can enhance educational content creation, and (2) how we can extend the application of existing computer vision algorithms. Our findings show that integrating object detection algorithms with 3D animation tool blender can provide a unique approach of learning.

This approach streamlines content creation and offers a transformative learning experience. Our study extends the use of computer vision algorithms into the educational domain, demonstrating their adaptability and effectiveness.

Overall, our results indicate the path to implement computer vision in education in a distinct way of 3d modelling and animation creation. The novelty of our approach lies in seamlessly integrating computer vision techniques into educational practices, filling a crucial gap in the field. Comparing with existing methods, our integrated approach offers scalability, efficiency, and customization, providing real-time interaction and feedback for students. One assumption affecting our analysis can be monitored in the test 2 of our research, which could impact the accuracy of our approach.

In conclusion, our study underscores the potential of computer vision techniques to revolutionize educational content creation. By addressing research questions and demonstrating effectiveness, we contribute to ongoing efforts to innovate teaching methodologies and enhance learning environment.

5.3 Limitations

While our study demonstrates promising results in enhancing educational content creation through the integration of computer vision techniques and 3D animation tools, it is not without limitations. One limitation is the reliance on specific datasets and tools, which may limit the generalizability of our findings. Additionally, the computational resources required for training object detection models and creating 3D animations may pose challenges for widespread adoption, particularly in resource-constrained educational settings. Furthermore, the accuracy and effectiveness of our approach may vary depending on the quality of the input data and the complexity of the educational content. Despite these limitations, our study provides valuable insights into the potential of leveraging advanced technologies to improve learning experiences.

5.4 Challenges:

5.4.1 Technical Challenges:

Computational Resources Strain

A significant hurdle in our endeavor is the demanding computational requirements of our system. The sophisticated nature of computer vision algorithms and the resource-intensive

animation generation process in Blender bpy can strain hardware capabilities. Ensuring smooth functionality across various computational setups is essential for widespread adoption.

Navigating Algorithmic Accuracy

The effectiveness of our educational animations heavily relies on the accuracy of the chosen object detection algorithm. Challenges may arise in accurately identifying and extracting object parameters, potentially leading to discrepancies in the reliability of educational content. Balancing accuracy with computational efficiency remains an ongoing challenge.

5.4.2 User Adoption Challenges:

Surmounting the Learning Curve

While our pipeline offers a groundbreaking approach, educators may face a steep learning curve in scripting and utilizing Blender bpy. Bridging this knowledge gap necessitates the development of comprehensive training resources to facilitate a seamless transition into the realm of educational animation creation.

Sustaining Student Engagement

Ensuring active student engagement with educational animations is crucial. Designing captivating content that caters to diverse learning styles and preferences requires constant assessment and adaptation. Strategies to enhance interactivity and interest retention are vital for achieving the desired educational impact.

Further, future advancement possible in this field of study is being discussed in the upcoming chapter with a conclusion.

Chapter 6

Future Directions and Conclusion

6.1 Future Directions

As the traditional techniques have certain limitation. But still there are several promising directions in which this research can go in the future. Firstly, broadening the dataset and applying sophisticated machine learning models to a wider variety of items and situations will improve our method's resilience and generalizability. Furthermore, investigating the integration of augmented reality (AR) [24] and virtual reality (VR) [25] technology as recently introduced by apple itself may elevate the production of instructional content and provide students even more immersive learning opportunities. Additionally, carrying out longitudinal research to evaluate our approach's long-term effects on student learning outcomes would offer important insights into how successful it becomes over time. Lastly, working with educators and other stakeholders to customize the technique to topic areas and educational environments may assist guarantee its applicability and relevance in real-world situations. Overall, continuing to innovate and iterate upon our approach in collaboration with relevant stakeholders will be essential for advancing the field of computer vision in education.

6.1.1 Pioneering Optimization Strategies

Future endeavours should explore optimization strategies to alleviate computational burdens. Implementing parallel processing or investigating cloud-based solutions can potentially enhance the scalability and accessibility of our educational animation pipeline.

6.1.2 Elevating User Interface Design

Enhancing the user interface of our pipeline is pivotal for fostering user-friendly interactions. A more intuitive design can empower educators and students alike, encouraging wider adoption and reducing barriers to entry.

6.1.3 Seamless Integration with Learning Management Systems

Integrating our educational animation pipeline seamlessly with existing learning management systems presents an exciting opportunity. This alignment can streamline the incorporation of animations into curricula, thereby enhancing the overall educational experience.

6.1.4 Collaborative Endeavours with Educational Institutions

Collaborating with educational institutions for real-world testing offers invaluable insights. Seeking feedback from educators and students in diverse educational settings can provide a nuanced understanding of the system's impact and guide further refinements.

6.1.5 Advanced ML Integration for Enhanced Educational Experiences

Integrating advanced machine learning (ML) algorithms (as mentioned in chapter 2 as literature review) holds great promise for enriching the educational animation pipeline. By incorporating ML techniques, we can:

- Improve object detection accuracy and efficiency using deep learning-based models like YOLO or SSD.
- Enable semantic understanding of scenes through semantic segmentation algorithms.
- Personalize learning experiences through dynamic adaptation based on learner interactions.
- Incorporate natural language processing for interactive learning interactions.
- Optimize educational content based on predictive analytics and learner performance data.
- Generate synthetic educational content using generative models like GANs or VAEs.

By integrating advanced ML algorithms, we can enhance the relevance, adaptability, and effectiveness of educational animations, ultimately empowering educators and learners with more engaging and impactful learning experiences.

Conclusion

In summary, our research indicates the great potential of combining computer vision methods with 3D animation software to transform the production of instructional material and broaden the use of the current algorithm. We have essentially taken 2D photographs and turned them into engaging 3D animations by using algorithms to extract useful information. Our study highlights the value of innovation in educational methods and establishes the foundation for further research in this field, despite certain limitations. We see a sustained focus on technology-driven teaching and multidisciplinary cooperation in the future to address the changing demands of students in the digital age. In essence, our research reveals a weakness in the conventional computer vision methodology and offers a roadmap for leveraging technology to transform the face of education.

References:

[1] Abdrakhmanov, Rustam & Tuimebayev, Assyl & Zhussipbek, Botagoz & Utebayev, Kalmurat & Nakhipova, Venera & Alchinbayeva, Oichagul & Makhanova, Gulfairuz & Kazhybayev, Olzhas. (2024). Applying Computer Vision and Machine Learning Techniques in STEM-Education Self-Study. 10.14569/IJACSA.2024.0150182.

[2] "RNN-Based Handwriting Recognition in Gboard," *Google AI Blog*. http://ai.googleblog.com/2019/03/rnn-based-handwriting-recognition-in.html (accessed Aug. 20, 2022).

3] Y. Wu, Y. Wang, S. Zhang, and H. Ogai, "Deep 3d object detection networks using lidar data: A review," IEEE Sensors Journal, vol. 21, no. 2, pp. 1152–1171, 2020.

[4] M. Onishi and T. Ise, "Explainable identification and mapping of trees using uav rgb image and deep learning," Scientific reports, vol. 11, no. 1, p. 903, 2021.

[5] D. Kiranov, M. Ryndin, and I. Kozlov, "Active learning and transfer learning for document segmentation," Programming and Computer Software, vol. 49, no. 7, pp. 566–573, 2023.

[6] T. Diwan, G. Anirudh, and J. V. Tembhurne, "Object detection using yolo: Challenges, architectural successors, datasets and applications," multimedia Tools and Applications, vol. 82, no. 6, pp. 9243–9275, 2023.

[7] W. Chen, H. Huang, S. Peng, C. Zhou, and C. Zhang, "Yolo-face: a real-time face detector," The Visual Computer, vol. 37, pp. 805–813, 2021.

[8] W. Zhang, Q. Xu et al., "Optimization of college english classroom teaching efficiency by deep learning sdd algorithm," Computational Intelligence and Neuroscience, vol. 2022, 2022.

[9] E. H. Ali, H. A. Jaber, and N. N. Kadhim, "New algorithm for localization of iris recognition using deep learning neural networks," Indonesian Journal of Electrical Engineering and Computer Science, vol. 29, no. 1, pp. 110–119, 2023.

[10] A. Nakada, R. Niikura, K. Otani, Y. Kurose, Y. Hayashi, K. Kitamura, H. Nakanishi, S. Kawano, T. Honda, K. Hasatani et al., "Improved object detection artificial intelligence using the revised retinanet model for the automatic detection of ulcerations, vascular lesions, and tumors in wireless capsule endoscopy," Biomedicines, vol. 11, no. 3, p. 942, 2023.

[11] N. I. Nife and M. Chtourou, “A comprehensive study of deep learning and performance comparison of deep neural network models (yolo, retinanet).” International Journal of Online & Biomedical Engineering, vol. 19, no. 12, 2023.

[12] D. I. Ilyasovich, “Blender program and its capabilities,” Gospodarka i Innowacje., vol. 41, pp. 382–385, 2023.

[13] J. M. Blain, The Complete Guide to Blender Graphics: Computer Modeling and Animation: Volume One. CRC Press, 2023.

[14] K. Murdock, Autodesk Maya 2024 Basics Guide. SDC Publications, 2023.

[15] S. Shan and S. Sun, “Check for updates design and implementation of virtual simulation teaching platform of sports action technology based on maya,” Innovative Computing Vol 1-Emerging Topics in Artificial Intelligence: Proceedings of IC 2023, vol. 1044, p. 464, 2023.

[16] J. G. Kabulunze, H. Mwangi, and K. Ogada, “Improve student per- formance in the psychiatry subject with 3d animation,” Journal of Agriculture, Science and Technology, vol. 22, no. 3, pp. 79–99, 2023.

[17] Y. Shen and N. Bai, “The integration and application of vr technology and 3d animation design,” Journal of Global Humanities and Social Sciences, vol. 4, no. 2, pp. 59–63, 2023.

[18] Z. Z. Abidin, S. Asmai, Z. A. Abas, N. Zakaria, and S. Ibrahim, “Development of edge detection for image segmentation,” in IOP Conference Series: Materials Science and Engineering, vol. 864, no. 1. IOP Publishing, 2020, p. 012058.

[19] R. Halimatussa’diyah, “Development animation 3d in science and math- ematics subjects for people with intellectual disabelities in extraordinary schools,” in Proceedings of the 6th FIRST 2022 International Conference (FIRST 2022), vol. 14. Springer Nature, 2023, p. 373.

[20] C. You, W. Dai, Y. Min, F. Liu, D. Clifton, S. K. Zhou, L. Staib, and J. Duncan, “Rethinking semi-supervised medical image segmentation: A variance-reduction perspective,” Advances in Neural Information Processing Systems, vol. 36, 2024.

[21] A. Marougkas, C. Troussas, A. Krouska, and C. Sgouropoulou, “Virtual reality in education: a review of learning theories, approaches and methodologies for the last decade,” Electronics, vol. 12, no. 13, p. 2832, 2023.

[22] D. Kumar, A. Haque, K. Mishra, F. Islam, B. K. Mishra, and S. Ahmad, “Exploring the transformative role of artificial intelligence and metaverse in education: A comprehensive review,” Metaverse Basic and Applied Research, vol. 2, pp. 55–55, 2023.

[23] J.Garzo´n,S.Baldiris,J.Gutie´rrez,J.Pavo´netal.,“Howdopedagogical approaches affect the impact of augmented reality on education? a meta- analysis and research synthesis,” Educational Research Review, vol. 31, p. 100334, 2020.

[24] M. Kljun, V. Geroimenko, and K. C˘opic˘ Pucihar, “Augmented reality in education: Current status and advancement of the field,” Augmented Reality in Education: A New Technology for Teaching and Learning, pp. 3–21, 2020.

[25] M. Sırakaya and D. Alsancak Sırakaya, “Augmented reality in stem education: A systematic review,” Interactive Learning Environments, vol. 30, no. 8, pp. 1556–1569, 2022..

[26] A. AL-Saffar, S. Awang, W. AL-Saiagh, A. S. AL-Khaleefa, and S. A. Abed, “A Sequential Handwriting Recognition Model Based on a Dynamically Configurable CRNN,” *Sensors*, vol. 21, no. 21, p. 7306, Nov. 2021, doi: 10.3390/s21217306.

[27] L. Kang, P. Riba, M. Rusinol, A. Fornes, and M. Villegas, “Content and Style Aware Generation of Text-line Images for Handwriting Recognition,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PP, Oct. 2021, doi: 10.1109/TPAMI.2021.3122572.

[28] J.-W. Wu, F. Yin, Y.-M. Zhang, X.-Y. Zhang, and C.-L. Liu, “Image-to-Markup Generation via Paired Adversarial Learning,” in *Machine Learning and Knowledge Discovery in Databases*, Cham, 2019, pp. 18–34. doi: 10.1007/978-3-030-10925-7_2.

[29] J. Michael, R. Labahn, T. Grüning, and J. Zöllner, “Evaluating Sequence-to-Sequence Models for Handwritten Text Recognition.” arXiv, Jul. 15, 2019. doi: 10.48550/arXiv.1903.07377.

[30] A. Vaswani *et al.*, “Attention Is All You Need.” arXiv, Dec. 05, 2017. doi: 10.48550/arXiv.1706.03762.

[31] C. Wick, J. Zöllner, and T. Grüning, “Transformer for Handwritten Text Recognition Using Bidirectional Post-decoding,” in *Document Analysis and Recognition – ICDAR 2021*, Cham, 2021, pp. 112–126. doi: 10.1007/978-3-030-86334-0_8.

[32] Y. Tao, Z. Jia, R. Ma, and S. Xu, “TRIG: Transformer-Based Text Recognizer with Initial Embedding Guidance,” *Electronics*, vol. 10, no. 22, p. 2780, Nov. 2021, doi: 10.3390/electronics10222780.

[33] M. Li *et al.*, “TrOCR: Transformer-based Optical Character Recognition with Pre-trained Models.” arXiv, Aug. 17, 2022. doi: 10.48550/arXiv.2109.10282.

[34] “Speech and Language Processing.” https://web.stanford.edu/~jurafsky/slp3/ (accessed Aug. 19, 2022).

[35] J. Agbinya, “Hidden Markov Modelling (HMM) - An Introduction,” 2020, pp. 17–34.

[36] J. H. AlKhateeb, J. Ren, J. Jiang, and H. Al-Muhtaseb, “Offline handwritten Arabic cursive text recognition using Hidden Markov Models and re-ranking,” *Pattern Recognit. Lett.*, vol. 32, no. 8, pp. 1081–1088, Jun. 2011, doi: 10.1016/j.patrec.2011.02.006.

[37] T. Chis and P. G. Harrison, “iSWoM: The Incremental Storage Workload Model Based on Hidden Markov Models,” in *Analytical and Stochastic Modeling Techniques and Applications*, Berlin, Heidelberg, 2013, pp. 127–141. doi: 10.1007/978-3-642-39408-9_10.

[38] T. Chis, “Sliding Hidden Markov Model for Evaluating Discrete Data,” in *Computer Performance Engineering*, Berlin, Heidelberg, 2013, pp. 251–262. doi: 10.1007/978-3-642-40725-3_19.

[39] J. Schenk and G. Rigoll, “Novel Hybrid NN/HMM Modelling Techniques for On-line Handwriting Recognition,” p. 5.

[40] A. Graves and J. Schmidhuber, “Offline Handwriting Recognition with Multidimensional Recurrent Neural Networks,” in *Advances in Neural Information Processing Systems*, 2008, vol. 21. Accessed: Aug. 19, 2022. [Online]. Available: https://papers.nips.cc/paper/2008/hash/66368270ffd51418ec58bd793f2d9b1b-Abstract.html

[41] M. Roos, “Recurrence in biological and artificial neural networks,” *Medium*, May 19, 2019. https://towardsdatascience.com/recurrence-in-biological-and-artificial-neural-networks-e8a6d5639781 (accessed Aug. 19, 2022).

[42] fdeloche, *English: A diagram for a one-unit recurrent neural network (RNN). From bottom to top : input state, hidden state, output state. U, V, W are the weights of the network. Compressed diagram on the left and the unfold version of it on the right.* 2017.

Accessed: Aug. 19, 2022. [Online]. Available: https://commons.wikimedia.org/wiki/File:Recurrent_neural_network_unfold.svg

[43] S. Hochreiter, "The vanishing gradient problem during learning recurrent neural nets and problem solutions," *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, vol. 6, no. 2, pp. 107–116, Apr. 1998, doi: 10.1142/S0218488598000094.

[44] "10.4. Bidirectional Recurrent Neural Networks — Dive into Deep Learning 1.0.0-alpha0 documentation." https://d2l.ai/chapter_recurrent-modern/bi-rnn.html (accessed Aug. 19, 2022).

[45] "10.1. Gated Recurrent Units (GRU) — Dive into Deep Learning 1.0.0-alpha0 documentation." https://d2l.ai/chapter_recurrent-modern/gru.html (accessed Aug. 19, 2022).

[46] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Trans. Signal Process.*, vol. 45, no. 11, pp. 2673–2681, Nov. 1997, doi: 10.1109/78.650093.

[47] S. Varsamopoulos, K. Bertels, and C. Almudever, *Designing neural network based decoders for surface codes*. 2018.

[48] L. Jian, H. Xiang, and G. Le, "LSTM-Based Attentional Embedding for English Machine Translation," *Sci. Program.*, vol. 2022, p. e3909726, Mar. 2022, doi: 10.1155/2022/3909726.

[49] A. Graves, A. Mohamed, and G. Hinton, "Speech Recognition with Deep Recurrent Neural Networks." arXiv, Mar. 22, 2013. doi: 10.48550/arXiv.1303.5778.

[50] Y. Tang, F. Yu, W. Pedrycz, X. Yang, J. Wang, and S. Liu, "Building Trend Fuzzy Granulation-Based LSTM Recurrent Neural Network for Long-Term Time-Series Forecasting," *IEEE Trans. Fuzzy Syst.*, vol. 30, no. 6, pp. 1599–1613, Jun. 2022, doi: 10.1109/TFUZZ.2021.3062723.

[51] B. Hussain, M. K. Afzal, S. Ahmad, and A. M. Mostafa, "Intelligent Traffic Flow Prediction Using Optimized GRU Model," *IEEE Access*, vol. 9, pp. 100736–100746, 2021, doi: 10.1109/ACCESS.2021.3097141.

[52] S. Yang, X. Yu, and Y. Zhou, *LSTM and GRU Neural Network Performance Comparison Study: Taking Yelp Review Dataset as an Example*. 2020, p. 101. doi: 10.1109/IWECAI50956.2020.00027.

[53] H. Zhan, S. Lyu, Y. Lu, and U. Pal, “DenseNet-CTC: An end-to-end RNN-free architecture for context-free string recognition,” *Comput. Vis. Image Underst.*, vol. 204, p. 103168, Mar. 2021, doi: 10.1016/j.cviu.2021.103168.

[54] X.-Y. Zhang, F. Yin, Y.-M. Zhang, C.-L. Liu, and Y. Bengio, “Drawing and Recognizing Chinese Characters with Recurrent Neural Network.” arXiv, Jun. 21, 2016. doi: 10.48550/arXiv.1606.06539.

[55] V. Carbune *et al.*, “Fast Multi-language LSTM-based Online Handwriting Recognition.” arXiv, Jan. 24, 2020. doi: 10.48550/arXiv.1902.10525.

[56] M. Schall, M.-P. Schambach, and M. O. Franz, “Improving gradient-based LSTM training for o ine handwriting recognition by careful selection of the optimization method,” p. 5.

[57] H. Kohli, J. Agarwal, and M. Kumar, “An improved method for text detection using Adam optimization algorithm,” *Glob. Transit. Proc.*, vol. 3, no. 1, pp. 230–234, Jun. 2022, doi: 10.1016/j.gltp.2022.03.028.

[58] F. Menasri, J. Louradour, A.-L. Bianne-Bernard, and C. Kermorvant, “The A2iA French handwriting recognition system at the Rimes-ICDAR2011 competition,” *Proc. SPIE - Int. Soc. Opt. Eng.*, vol. 8297, p. 51, Jan. 2012, doi: 10.1117/12.911981.

[59] V. Pham, T. Bluche, C. Kermorvant, and J. Louradour, “Dropout Improves Recurrent Neural Networks for Handwriting Recognition,” in *2014 14th International Conference on Frontiers in Handwriting Recognition*, Sep. 2014, pp. 285–290. doi: 10.1109/ICFHR.2014.55.

[60] T. Bluche and R. Messina, “Gated Convolutional Recurrent Neural Networks for Multilingual Handwriting Recognition,” in *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, Nov. 2017, vol. 01, pp. 646–651. doi: 10.1109/ICDAR.2017.111.

[61] L. Kang, J. I. Toledo, P. Riba, M. Villegas, A. Fornés, and M. Rusiñol, “Convolve, Attend and Spell: An Attention-based Sequence-to-Sequence Model for Handwritten Word Recognition,” in *Pattern Recognition*, Cham, 2019, pp. 459–472. doi: 10.1007/978-3-030-12939-2_32.

[62] M. P. Kantipudi, S. Kumar, and A. Kumar Jha, “Scene Text Recognition Based on Bidirectional LSTM and Deep Neural Network,” *Comput. Intell. Neurosci.*, vol. 2021, p. e2676780, Nov. 2021, doi: 10.1155/2021/2676780.

[63] B. Stuner, C. Chatelain, and T. Paquet, “Handwriting recognition using cohort of LSTM and lexicon verification with extremely large lexicon,” *Multimed. Tools Appl.*, vol. 79, no. 45, pp. 34407–34427, Dec. 2020, doi: 10.1007/s11042-020-09198-6.

[64] J. Russin, M. Zolfaghar, S. A. Park, E. Boorman, and R. C. O’Reilly, “Complementary Structure-Learning Neural Networks for Relational Reasoning,” *CogSci Annu. Conf. Cogn. Sci. Soc. Cogn. Sci. Soc. US Conf.*, vol. 2021, pp. 1560–1566, Jul. 2021.

[65] L. Kang, P. Riba, M. Rusiñol, A. Fornés, and M. Villegas, “Pay attention to what you read: Non-recurrent handwritten text-Line recognition,” *Pattern Recognit.*, vol. 129, p. 108766, Sep. 2022, doi: 10.1016/j.patcog.2022.108766.

[66] Z. Li, F. Liu, W. Yang, S. Peng, and J. Zhou, “A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects,” *IEEE Trans. Neural Netw. Learn. Syst.*, pp. 1–21, 2021, doi: 10.1109/TNNLS.2021.3084827.

[67] V. H. Phung and E. J. Rhee, “A High-Accuracy Model Average Ensemble of Convolutional Neural Networks for Classification of Cloud Image Patches on Small Datasets,” *Appl. Sci.*, vol. 9, no. 21, Art. no. 21, Jan. 2019, doi: 10.3390/app9214500.

[68] F. Yu and V. Koltun, “Multi-Scale Context Aggregation by Dilated Convolutions.” arXiv, Apr. 30, 2016. doi: 10.48550/arXiv.1511.07122.

mput., vol. 1, no. 4, pp. 541–551, Dec. 1989, doi: 10.1162/neco.1989.1.4.541.

[82] F. P. Such, D. Peri, F. Brockler, P. Hutkowski, and R. Ptucha, “Fully Convolutional Networks for Handwriting Recognition.” arXiv, Jul. 10, 2019. doi: 10.48550/arXiv.1907.04888.

[83] Y.-C. Wu, F. Yin, and C.-L. Liu, “Improving handwritten Chinese text recognition using neural network language models and convolutional neural network shape models,” *Pattern Recognit.*, vol. 65, pp. 251–264, May 2017, doi: 10.1016/j.patcog.2016.12.026.

[84] T. Bluche, H. Ney, and C. Kermorvant, “Tandem HMM with convolutional neural network for handwritten word recognition,” in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, May 2013, pp. 2390–2394. doi: 10.1109/ICASSP.2013.6638083.

[85] X. Xiao, L. Jin, Y. Yang, W. Yang, J. Sun, and T. Chang, “Building Fast and Compact Convolutional Neural Networks for Offline Handwritten Chinese Character Recognition.” arXiv, Feb. 25, 2017. doi: 10.48550/arXiv.1702.07975.

[86] H. Ali, A. Ullah, T. Iqbal, and S. Khattak, “Pioneer dataset and automatic recognition of Urdu handwritten characters using a deep autoencoder and convolutional neural network.” Dec. 17, 2019. doi: 10.1007/s42452-019-1914-1.

[87] W. Yu, C. Guo, K. Liu, and H. Yang, “Handwritten Digital Recognition Optimization Method based on Deep Learning,” in *2020 Chinese Automation Congress (CAC)*, Nov. 2020, pp. 1705–1709. doi: 10.1109/CAC51589.2020.9327647.

[88] H. M. Najadat, A. A. Alshboul, and A. F. Alabed, “Arabic Handwritten Characters Recognition using Convolutional Neural Network,” in *2019 10th International Conference on Information and Communication Systems (ICICS)*, Jun. 2019, pp. 147–151. doi: 10.1109/IACS.2019.8809122.

[89] Y. Chherawala, H. J. G. A. Dolfing, R. S. Dixon, and J. R. Bellegarda, “Embedded Large–Scale Handwritten Chinese Character Recognition,” in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2020, pp. 8169–8173. doi: 10.1109/ICASSP40776.2020.9053084.

[90] R. R. Chowdhury, M. S. Hossain, R. ul Islam, K. Andersson, and S. Hossain, “Bangla Handwritten Character Recognition using Convolutional Neural Network with Data Augmentation,” in *2019 Joint 8th International Conference on Informatics, Electronics & Vision (ICIEV) and 2019 3rd International Conference on Imaging, Vision & Pattern Recognition (icIVPR)*, May 2019, pp. 318–323. doi: 10.1109/ICIEV.2019.8858545.

[91] S. Singh, A. Sharma, and V. K. Chauhan, “Online handwritten Gurmukhi word recognition using fine-tuned Deep Convolutional Neural Network on offline features,” *Mach. Learn. Appl.*, vol. 5, p. 100037, Sep. 2021, doi: 10.1016/j.mlwa.2021.100037.

[92] Y. Weng and C. Xia, “A New Deep Learning-Based Handwritten Character Recognition System on Mobile Computing Devices,” *Mob. Netw. Appl.*, vol. 25, no. 2, pp. 402–411, Apr. 2020, doi: 10.1007/s11036-019-01243-5.

[93] J. Puigcerver, “Are Multidimensional Recurrent Layers Really Necessary for Handwritten Text Recognition?,” in *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, Nov. 2017, vol. 01, pp. 67–72. doi: 10.1109/ICDAR.2017.20.

[94] B. Balci, D. Saadati, and D. Shiferaw, “Handwritten Text Recognition using Deep Learning,” p. 8.

[95] S. Albahli, M. Nawaz, A. Javed, and A. Irtaza, “An improved faster-RCNN model for handwritten character recognition,” *Arab. J. Sci. Eng.*, vol. 46, no. 9, pp. 8509–8523, Sep. 2021, doi: 10.1007/s13369-021-05471-4.

[96] K. He, G. Gkioxari, P. Dollár, and R. Girshick, “Mask R-CNN.” arXiv, Jan. 24, 2018. doi: 10.48550/arXiv.1703.06870.

[97] K. Xu *et al.*, “Show, Attend and Tell: Neural Image Caption Generation with Visual Attention.” arXiv, Apr. 19, 2016. doi: 10.48550/arXiv.1502.03044.

[98] I. Sutskever, O. Vinyals, and Q. V. Le, “Sequence to Sequence Learning with Neural Networks.” arXiv, Dec. 14, 2014. doi: 10.48550/arXiv.1409.3215.

[99] D. Bahdanau, K. Cho, and Y. Bengio, “Neural Machine Translation by Jointly Learning to Align and Translate.” arXiv, May 19, 2016. Accessed: Aug. 19, 2022. [Online]. Available: http://arxiv.org/abs/1409.0473

[100] A. Galassi, M. Lippi, and P. Torroni, “Attention in Natural Language Processing,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 10, pp. 4291–4308, Oct. 2021, doi: 10.1109/TNNLS.2020.3019893.

[101] R. Geetha, T. Thilagam, and T. Padmavathy, “Effective offline handwritten text recognition model based on a sequence-to-sequence approach with CNN–RNN networks,” *Neural Comput. Appl.*, vol. 33, no. 17, pp. 10923–10934, Sep. 2021, doi: 10.1007/s00521-020-05556-5.

[102] X. Zhang, F. Chen, and R. Huang, “A Combination of RNN and CNN for Attention-based Relation Classification,” *Procedia Comput. Sci.*, vol. 131, pp. 911–917, Jan. 2018, doi: 10.1016/j.procs.2018.04.221.

[103] D. Kass and E. Vats, “AttentionHTR: Handwritten Text Recognition Based on Attention Encoder-Decoder Networks.” arXiv, Apr. 01, 2022. doi: 10.48550/arXiv.2201.09390.

[104] J. Poulos and R. Valle, “Character-Based Handwritten Text Transcription with Attention Networks,” *Neural Comput. Appl.*, vol. 33, no. 16, pp. 10563–10573, Aug. 2021, doi: 10.1007/s00521-021-05813-1.