Shri Vile Parle Kelavani Mandal's
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
(Autonomous College Affiliated to the University of Mumbai)
NAAC Accredited with "A" Grade (CGPA : 3.18)

Visualizing Kinematics

Submitted in partial fulfillment of the requirements
of the course Innovative Product Development
under

T. Y. B. Tech. Computer Science and Engineering (Data Science)

By

Dev Patel 60009200016

Sharvari Chawade 60009200037

Dhruv Salot 60009200048

Guide

Prof. Kanchan Dabre
Assistant Professor

University of Mumbai
2021-2023

CERTIFICATE

This is to certify that the project entitled **“Visualizing Kinematics”** is a bonafide work of **Dev Patel 60009200016, Sharvari Chawade 60009200037, Dhruv Salot 60009200048** submitted to the **Department of Computer Science and Engineering (Data Science)** in partial fulfillment of the requirement for the course of Innovative Product Development.

--	--------------------------------------
(Prof. Kanchan Dabre) **Internal Guide**	**(Dr. Kriti Srivastava)** **Head of the Department**

(Dr. Hari Vasudevan)
Principal

Project Report Approval for Innovative Product Development.

This project report entitled ***Visualizing Kinematics*** by ***Dev Patel 60009200016, Sharvari Chawade 60009200037, Dhruv Salot 60009200048*** is approved for the course of Innovative Product Development**.**

Examiners

1. ______________________________

2. ______________________________

Date: 20/05/2023

Place: Mumbai

Declaration

I/We declare that this written submission represents my/our ideas in my/our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that We have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

- -

Dev Patel (60009200016)

- -

Sharvari Chawade (60009200037)

- -

Dhruv Salot (60009200048)

Date: 20/05/2023

Acknowledgement

We would like to express our gratitude to our professor Kanchan Dabre for her continuous input and constructive criticism which helped to complete the work. This work would not have been possible without her constant motivation and support.

We would like to acknowledge our sincere thanks and are indebted to the Principal Dr. Hari Vasudevan for granting us the usage of the college library facilities for the purpose of reference for this report.

We are also thankful to our Head of Department Dr. Kriti Srivastava for allowing us to use the computer laboratory and the internet facilities.

We would also like to thank the people who contributed directly and indirectly towards the success of this work.

Abstract

The ability to visualize complex Physics concepts is essential for students to fully comprehend and master the subject. However, traditional teaching methods often fall short in this aspect, and as a result, students struggle to grasp abstract concepts. This has led to the development of this tool that aids in the visualization of physics concepts through animation. Previous research in this area has primarily focused on using machine learning algorithms such as Convolutional Neural Networks (CNNs) for image recognition. However, these approaches have had limited success in accurately identifying and interpreting hand-drawn physics diagrams due to a lack of suitable training data and insufficient preprocessing techniques. To overcome these limitations, a novel approach that combines CNNs with artificial intelligence models for improved accuracy and results is proposed. This proposed method utilizes a pre-processing step to the generated training data, which is then fed to the CNN model for shape recognition. To further enhance this approach, additional preprocessing techniques such as Gaussian blur, Canny Edge Detection, Grayscale conversion, and thresholding to improve image segmentation and classification were implemented. Several techniques to reduce overfitting and improve model performance were implemented which showed an accuracy of 98.20%. The input diagrams will then be processed through the Artificial Intelligence model which visualizes how the system performs under set conditions. Overall, the proposed method provides an effective approach for visualizing complex physics concepts through hand-drawn diagrams. The approach has significant implications for education and can aid in the visualization of complex physics concepts, providing a more interactive and immersive learning experience for students.

Table of Contents

List of Figures

List of Abbreviations

Sr. No.	Abbreviation	Expanded form
1	AI	Artificial Intelligence
2	CNN	Convolutional Neural Network
3	SVM	Support Vector Machine
4	GPU	Graphic Processing Unit

1. INTRODUCTION

The proposed work aims to develop a tool that can transform hand-drawn physics diagrams into animated representations, enhancing students' understanding of complex concepts. By utilizing a Convolutional Neural Network (CNN) model, the tool provides accurate shape recognition and realistic visualizations in real-world conditions. This research seeks to bridge the gap between abstract physics theories and their practical applications, improving the learning experience for students and offering teachers a more innovative teaching method.

1.1 Aim of the research

The proposed idea aims to address a significant issue in the field of education, particularly in the teaching and learning of abstract concepts in physics. These concepts often require visual representation to aid in comprehension, as they can be challenging for students to visualize. The aim of this research is to develop a tool that can take hand-drawn physics diagrams as input, understand their meaning, and produce an animated output that demonstrates the hand-drawn diagram in real-world conditions. This will provide students with a better and more comprehensive learning experience that can also help teachers explain physics concepts in a more innovative way.

The tool will enable students to draw shapes and visualize complex physics concepts like kinematics, dynamics, and optics. This will help them to understand these concepts better by seeing practical applications of the theories they are learning. The implementation of the proposed tool will be done with the use of a Convolutional Neural Network (CNN) model. This model will help in better accuracy and results, ensuring that the visual output of the tool is as close to real-world conditions as possible.

The tool will have a user-friendly interface, where students can draw their diagrams on a canvas. The hand-drawn images will then undergo an Artificial Intelligence (AI) model to interpret the meaning behind the image. This will be followed by an animation video output that demonstrates the image in the given conditions. The tool's animation video will depict the shapes drawn by the user under the effects of gravity, friction, and other physical laws, creating a more engaging and realistic representation.

The proposed tool's significance lies in its ability to improve the learning experience of students and provide teachers with a more innovative way of teaching physics concepts. Traditional teaching methods may not always be enough to engage students and help them understand complex concepts. Therefore, this tool provides an alternative method of teaching that is both engaging and interactive. This will help students visualize the concepts more clearly and create a better foundation of understanding, leading to better academic performance.

The proposed tool will help to bridge the gap between abstract concepts and their practical applications, providing students with a more comprehensive learning experience. While there are existing algorithms and papers that have attempted to create similar systems, the proposed tool aims to overcome the gaps in these systems, such as bias and misclassification, by utilizing a CNN model. The results obtained from the proposed system will be compared with the best performing benchmark method to measure the level of improvement achieved. The effects of preprocessing, augmentation, additional segmentation, residual connection, parallelization, and multiclass categorization will also be studied to show the impact of these techniques on the tool's performance.

1.2 Motivation

The motivation behind the proposed work stems from the difficulty students often face in comprehending and visualizing complex Physics concepts. Traditional teaching and learning methods may not always effectively convey the practical applications and real-world scenarios of these concepts. This lack of visualization can impede the development of a strong foundational understanding of Physics, which is necessary for further progress in the field.

To address this issue, the proposed research aims to create a system that can detect and analyze hand-drawn Physics diagrams and provide an animated output for better understanding. By using this tool, students will be able to draw shapes and visualize complex Physics concepts like kinematics, which will help them understand better. The implementation of a CNN model will provide better accuracy and results, making the system more efficient and effective.

Through this work, it is hoped that it can provide students with a new and innovative approach to learning Physics that is engaging, interactive, and practical. By providing a visual representation of the concepts, students will be able to develop a more robust understanding of Physics, which will ultimately benefit them in their academic and professional pursuits.

1.3 Organization of Report

The report is organized into several sections that provide a comprehensive understanding of the proposed system for detecting and animating hand-drawn physics diagrams. The first section of the report presents an introduction to the work, including the motivation behind it, the problem formulation, and the aims and objectives. This section sets the context for the proposed system and provides an overview of the entire report.

The second section of the report describes the literature review that was conducted to explore the existing approaches and algorithms used for detecting and visualizing hand-drawn diagrams. This section will provide an understanding of the gaps in the existing system and the need for a more efficient solution.

The third section focusses on the problem statement of the work and its scope. This will focus on describing the exact problem that is being tackled. It will also analyze the scope of the research and where it can used and how.

The fourth section of the report provides a detailed description of the proposed system's architecture, including the various stages involved in detecting and animating the hand-drawn diagrams. This section also discusses the implementation of the system using a CNN model and the advantages of using this approach.

The fifth section of the report presents the experimental setup and the results obtained from the system's evaluation. This section will discuss the performance metrics used to evaluate the system's efficiency and compare the results obtained with those of existing methods.

The final section of the report will provide a conclusion summarizing the main findings and contributions of the work, along with the limitations and future scope of the proposed system.

2. LITERATURE SURVEY

There has been a lot of research done in the field of shape detection using CNN. In contrast to that, the collection of research papers on visualizing Physics using Python was less. A lot of papers were found to be useful for the proposed system after thorough scanning, some of them are:

Table 1. Comparative analysis of existing methods for object visualization and simulations

Paper Title and Year	Methodology	Implementation Details	Results	Dataset Description	Relevance to Research
S. Gomez, "Shape recognition using machine learning" (2011)	SVM Classification	Angles of the shape used as branching factor for tree-based recognition	97.3% accuracy	Shapes with sharp borders and closed nature	Solves shape detection in the work
J. C. Garcia, G. A. Angulo, "Shape classification using deep convolutional neural networks" (2017)	Deep Convolutional Neural Networks	Deep CNN architecture for shape classification	Higher accuracy compared to traditional methods	Various shape datasets	Shape classification is relevant to the work
M. Liu et al. "Shape recognition using a hybrid neural network with improved learning algorithm" (2019)	Hybrid Neural Network	Hybrid neural network with improved learning algorithm	Improved accuracy compared to other methods	Shape dataset with different classes	Utilizes a hybrid neural network for shape recognition
J. Liu et al. "Deep shape recognition with depth images" (2019)	Convolutional Neural Networks	Deep CNN architecture for shape recognition with depth images	Improved recognition performance with depth information	Depth images of various shapes	Applies deep learning to shape recognition
M. Z. Khan, A. Majid, "Shape recognition using machine learning: A survey" (2019)	Machine Learning	Survey of different machine learning techniques for shape recognition	Overview of various shape recognition methods	N/A	Provides an overview of shape recognition technique

Ajeet Ram Pathak et al. "Application of Deep Learning for Object Detection" (Year not provided)	Keras, Caffe, Theano, Torch	Various deep learning frameworks for object detection	Improved performance with better GPUs	Datasets like CIFAR, ImageNet, and others	Object detection is relevant to the work
Zhao, G., Xie, L. "Research on Kinematics Simulation of Parameterized Mechanism Based on Visualization in Scientific Computing" (2008)	OpenGL, VC++, MFC Compiler	Development of software for visualization and analysis of kinematic mechanisms	Efficient setup, motion simulation, visualization	Kinematic mechanisms and motion properties	Provides insights into visualization of physics simulations
Schroeder, D.V. "Physics Simulations in Python" (2018)	GlowScript, VPython	Implementation of physics concepts like projectile motion and pendulums	Application of physics concepts	N/A	Python-based simulations and visualization
P. Payeur et al. "Trajectory prediction for moving objects using artificial neural networks" (1995)	VLSI neural network chips (ETANN)	Training of neural network chips for real-time trajectory prediction	Trajectory prediction for moving objects	N/A	Neural network-based trajectory prediction
C. Li et al. "Real-Time Physics-Based Animation with Machine Learning" (2018)	Neural networks	Real-time physics-based animation using machine learning techniques	Improved efficiency and realism of animations	N/A	Application of machine learning in animation
N. A. Amir et al. "Physics Visualization for Virtual Reality: An Overview" (2020)	Virtual Reality (VR)	Overview of VR in physics visualization and its applications	Enhances understanding of physics concepts	N/A	Explores VR applications in physics visualization
P. Gupta et al. "Interactive Visualization of Physics	WebGL	WebGL for interactive visualization of physics simulations	Creation of interactive and immersive visualizations	N/A	Demonstrates WebGL for physics simulations

Simulations Using WebGL" (2018)					visualization
S. K. Mishra et al. "Augmented Reality for Enhancing Physics Education: A Review" (2020)	Augmented Reality (AR)	Review of AR applications in physics education and enhancing understanding	Enhances physics education and engagement	N/A	Discusses AR's potential in physics education
M. E. Wirthlin et al. "Physics Education with Python: Analysing Data from Video" (2016)	Python	Python and video analysis techniques for physics education	Data analysis and quantitative measurements	Video data analysis for physics experiments	Python-based video analysis for physics education

S. Gomez proposed a paper on Shape recognition using machine learning [1], this paper solves the shape detection part of the problem statement. In the work, the machine learning technique used is SVM Classification which bodes a result with 97.3% accuracy. The proposed model uses angles of the shape as a branching factor on which they branch trees at every level. The shapes that are considered are just the one with sharp borders and closed in nature. Shapes like circles, crosses, lines are not considered which will be needed for shape detection.

Ajeet Ram Pathak, Manjusha Pandey, and Siddharth Rautaray discuss the different ways to perform object detection [2]. Various frameworks have been used to perform deep learning like Keras, caffe, Theano, torch etc., with this information on the programming language that they are compatible with is also discussed. Various datasets like cifar, image net and others that are used for object detection are stated. Having discussed different models of object detection they concluded the paper with stating that better GPUs are needed to deliver better performance as image data is increasing day by day.

Zhao, G., Xie, L. move towards discussing the second part of the proposed system [3]. The focus is on the topic of visualizing physics simulation. It is about a software developed for visualization using openGL, VC++ and MFC Compiler. Conclusion drawn from this work were that their software could efficiently set up the mechanism, simulate the motion, plot the trace of any point on the mechanism, output any kinematic variables for analysis, adjust the angle of view (AoV) and zoom the field of view (FoV), details about the motion properties.

Schroeder and D.V. start from the basics of creating a shape in Python using GlowScript–VPython [4]. Further implementation of various physics concepts like projectile motion, pendulums, orbital motion, molecular dynamics, and random processes were discussed. Discussion on why to choose Python as a language to perform these actions were carried out.

P. Payeur, Hoang Le-Huy and C. M. Gosselin deal with giving a smart solution by predicting using an artificial neural net [5]. The problem of giving the trajectory of a body in motion in real time has been picked up. The training is done by VLSI neural network chips Electrically Trainable Analog Neural Network (ETANN) offered by Intel Neural Network Group.

C. Li, H. Zhang, and C. Liang proposed a paper on "Real-Time Physics-Based Animation with Machine Learning" [6]. The paper focuses on using machine learning techniques, specifically neural networks, to generate real-time physics-based animations. The authors discuss the challenges in physics-based animation and how machine learning can be applied to improve the efficiency and realism of the animations.

N. A. Amir, S. B. Shafie, and M. F. R. Rahim published a paper titled "Physics Visualization for Virtual Reality: An Overview" [7]. The paper provides an overview of using virtual reality (VR) technology for physics visualization. It explores the benefits of VR in enhancing the understanding of physics concepts and discusses various applications and techniques used in physics visualization with VR.

P. Gupta, S. V. Singh, and R. Patel presented a paper on "Interactive Visualization of Physics Simulations Using WebGL" [8]. The paper focuses on using WebGL, a JavaScript API for rendering interactive 2D and 3D graphics, for visualizing physics simulations. The authors discuss the implementation of WebGL and its advantages in creating interactive and immersive visualizations of physics concepts.

S. K. Mishra and S. K. Pradhan published a paper titled "Augmented Reality for Enhancing Physics Education: A Review" [9]. The paper provides a comprehensive review of the use of augmented reality (AR) in physics education. It discusses the potential of AR in visualizing abstract physics concepts, conducting virtual experiments, and enhancing student engagement and understanding.

M. E. Wirthlin and J. D. Ridgway presented a paper on "Physics Education with Python: Analysing Data from Video" [10]. The paper focuses on using Python programming language and video analysis techniques to enhance physics education. The authors discuss how Python can be used to analyse video data, extract relevant information, and make quantitative measurements for various physics experiments and demonstrations.

J. C. Garcia, G. A. Angulo propose a shape classification method using deep convolutional neural networks (CNNs) [11]. They design a deep CNN architecture specifically tailored for shape recognition tasks. The network is trained on a large dataset of shape images, leveraging the hierarchical features learned by the CNN layers to discriminate between different shapes. The experimental results demonstrate that the proposed deep CNN approach achieves higher accuracy compared to traditional shape recognition methods, highlighting the effectiveness of deep learning for shape classification.

"Shape recognition using a hybrid neural network with improved learning algorithm" (2019): In the paper presented by M. Liu et al., a shape recognition approach utilizing a hybrid neural network with an improved learning algorithm [12]. The proposed hybrid neural network combines multiple neural network models, including convolutional neural networks (CNNs) and multilayer perceptrons (MLPs), to enhance the recognition performance. Additionally, an improved learning algorithm is developed to optimize the training process and improve the convergence speed of the network. Experimental results demonstrate that the hybrid neural network achieves improved accuracy compared to other shape recognition methods, indicating the effectiveness of the proposed approach.

J. Liu et al. focuses on is on deep shape recognition using depth images. Depth images capture the geometric structure of objects and provide additional information compared to traditional RGB images [13]. The authors propose a deep convolutional neural network (CNN) architecture that takes depth images as input and extracts discriminative features for shape recognition. Experimental evaluations demonstrate that incorporating depth information leads to improved recognition performance compared to using RGB images alone. The study highlights the potential of leveraging depth images for deep shape recognition tasks.

M. Z. Khan, A. Majid provide a comprehensive survey of shape recognition techniques using machine learning [14]. The authors review various machine learning approaches applied to shape recognition tasks, including traditional methods and deep learning-based methods. The survey covers different types of shape representations, feature extraction techniques, and classification algorithms employed in shape recognition.

D. M. Green, D. A. Paterson explores the potential of visualization techniques in improving simulation processes [15]. The authors discuss various visualization methods, such as 3D rendering, animation, and interactive interfaces, and highlight their benefits in enhancing the understanding and analysis of simulated physics phenomena.

P. K. Sharma, K. Sharma present a framework for physics simulation and visualization using the Unity3D game engine [16]. The authors demonstrate the implementation of various physics concepts, including projectile motion, pendulums, and collisions, and showcase the real-time interactive visualization capabilities offered by Unity3D.

K. Bodin et al. discusses the use of Unity3D as a platform for visualizing physics simulations [17]. The authors explore the implementation of physics-based simulations, including rigid body dynamics, fluid dynamics, and particle systems, and showcase the interactive visualization capabilities provided by Unity3D.

M. Elsner, M. Zambelli explore the use of Unity3D for developing physics simulations in educational contexts [18]. The authors discuss the implementation of simulations for concepts like Newton's laws, kinematics, and electromagnetism, emphasizing the potential of interactive and immersive visualizations in enhancing physics education.

A. Subhan presents a framework for physics simulation and visualization using MATLAB [19]. They demonstrate the implementation of simulations for concepts like projectile motion, simple harmonic motion, and wave propagation, highlighting the capabilities of MATLAB's visualization tools in teaching undergraduate physics courses.

N. Aspragathos, S. Sakka explore the integration of machine learning techniques, specifically deep learning, in physics-based animation [20]. The authors discuss the application of neural networks for modelling complex physics phenomena and generating realistic animations. The paper highlights the potential of machine learning in improving the efficiency and realism of physics-based animations.

3. PROBLEM DESCRIPTION

3.1 Problem Statement

The current teaching methods for Physics concepts lack visual representation, making it difficult for students to grasp the concepts in entirety. This can lead to a lack of interest in the subject and poor performance in exams. The proposed tool aims to bridge this gap by providing a platform for students to draw hand-drawn physics diagrams and understand the practical application of these concepts in real-world scenarios.

The main challenge in achieving this goal is the ability to develop an Artificial Intelligence model that can accurately detect and comprehend the meaning of hand-drawn physics diagrams. Additionally, the model must be capable of generating animations that accurately demonstrate the concept being portrayed in the diagram. The model must be trained on a large dataset of hand-drawn physics diagrams to improve its accuracy and performance.

This research will provide a solution to the lack of visual representation in current Physics teaching methods and help students understand the practical application of concepts. Additionally, the proposed tool can be used by teachers as a teaching aid to make Physics concepts more engaging and interactive. Overall, this work aims to enhance the learning experience for students and make Physics concepts more accessible and easier to understand.

3.2 Scope of the Research

The scope of this research is to develop a tool that facilitates the visualization of complex physics concepts for students. The primary objective is to create an interactive system that can take hand-drawn physics diagrams as input and generate animated outputs illustrating the concepts in real-world scenarios. This tool aims to enhance understanding and comprehension by providing visual representations of abstract physics principles.

The implementation of the tool will rely on a Convolutional Neural Network (CNN) model. CNNs are chosen due to their ability to handle image recognition tasks effectively, providing high accuracy and improved results. By leveraging the power of CNNs, the system can accurately analyse hand-drawn diagrams and generate animated simulations corresponding to the depicted physics concepts.

The envisioned tool has broad applications, particularly in the fields of physics education, engineering, and research. It will serve as an educational aid, assisting students in comprehending various physics concepts, particularly in the domain of kinematics. Through the tool, students will be able to draw shapes and visualize complex physics phenomena, facilitating their understanding and making it easier to grasp abstract concepts.

Moreover, the work includes the development of an artificial intelligence model that can visualize the system under specific conditions. Factors such as gravity, friction, and other real-world parameters will be incorporated into the model, allowing users to observe the behaviour of the system in different scenarios. The model will also possess the capability to recognize and interpret various shapes drawn by the user, enabling the implementation of those shapes within the simulated physics environment. This functionality aims to provide a comprehensive understanding of the concepts and address related questions through interactive simulations.

However, several challenges must be addressed within the research scope. One major challenge is dealing with imbalanced datasets used for training the model, as this can lead to biased results and inaccurate shape recognition. Additionally, the system may face difficulties in correctly classifying shapes with specific shades of colours, requiring mitigation strategies to improve accuracy and performance. Resolving these challenges is vital to ensure the tool's effectiveness, reliability, and accuracy, ultimately enhancing the learning experience for students.

In summary, the research aims to develop a tool utilizing a CNN model to visualize complex physics concepts. The tool will accept hand-drawn physics diagrams, generate animated simulations, and provide an interactive learning experience. Its broad applications encompass physics education, engineering, and research. However, challenges such as imbalanced datasets and shape misclassification need to be addressed to ensure accurate and effective performance.

4. DESIGN

Visualization of complex physics concepts is a challenging task for students, and the proposed system aims to provide a solution to this problem. The system will enable students to draw shapes and visualize complex concepts with ease. The system will recognize various shapes drawn by the user and implement them under gravity, friction, etc., providing the user with a better understanding of the concepts and related questions.

The proposed system will consist of three main modules: Dataset Preprocessing, Train Model, and Output Display. The first module involves generating and processing the training data using Pillow, which involves three datasets: hand-drawn by people, drawn on the app's canvas, and preprocessed. These image files are parsed, extracted, and modified into matrices of pixel intensities. The images are converted to grayscale, inverted, and saved as a NumPy array for ease of use. The second module involves training a Convolutional Neural Network (CNN) model for shape recognition with the preprocessed dataset. The shapes will then be placed on the canvas by after determining the bound region for the same for appropriate positioning.

Algorithm 1 Determining Object Bounding Area

Require: $image = np.array$
Ensure: $image$ is not empty
$x \leftarrow 0$
$y \leftarrow 0$
$max_x \leftarrow -\infty$
$min_x \leftarrow \infty$
$max_y \leftarrow -\infty$
$min_y \leftarrow \infty$
$row, col = image.shape$
while $x \neq row$ **do**
  while $y \neq col$ **do**
    if $y \leq min_y \cap image(x, y) = 1$ **then**
      $min_y \leftarrow y$
    else if $y \geq max_y \cap image(x, y) = 1$ **then**
      $max_y \leftarrow y$
    end if
    if $x \leq min_x \cap image(x, y) = 1$ **then**
      $min_x \leftarrow x$
    else if $x \geq max_x \cap image(x, y) = 1$ **then**
      $max_x \leftarrow x$
    end if
  end while
end while
Return $((min_x, min_y), (max_x, max_y))$

Fig 1. Algorithm for determining bounding region

The algorithm for determining the bounding area of an object in an image as shown in Figure 1 is explained as follows. It takes an input image represented as a NumPy array and returns the coordinates of the object's bounding box.

1. Initialize variables:
   a. x and y are set to 0, representing the current row and column indices.
   b. maxx and maxy are set to negative infinity, representing the maximum row and column indices encountered so far.
   c. minx and miny are set to positive infinity, representing the minimum row and column indices encountered so far.
2. Get the dimensions of the image:
   a. The variables row and col are set to the number of rows and columns in the image, respectively.
3. Iterate through the rows of the image:
   a. While x is not equal to the total number of rows in the image, continue to the next step.
4. Iterate through the columns of the image:
   a. While y is not equal to the total number of columns in the image, continue to the next step.
5. Check if the current pixel is part of the object:
   a. If y is less than or equal to miny and the pixel at coordinates (x, y) in the image is equal to 1 (indicating the presence of an object), update miny with the value of y.
   b. Otherwise, if y is greater than or equal to max_y and the pixel at coordinates (x, y) in the image is equal to 1, update max_y with the value of y.
6. Check if the current row is part of the object:
   a. If x is less than or equal to minx and the pixel at coordinates (x, y) in the image is equal to 1, update minx with the value of x.
   b. Otherwise, if x is greater than or equal to max_x and the pixel at coordinates (x, y) in the image is equal to 1, update max_x with the value of x.
7. Increment the column index:
   a. Increment y by 1.
8. Increment the row index:
   a. Increment x by 1.

9. Return the coordinates of the bounding box:
    a. Return ((min_x, min_y), (max_x, max_y)), representing the top-left and bottom-right coordinates of the object's bounding box.

This algorithm essentially scans through the image, pixel by pixel, to find the minimum and maximum row and column indices that contain the object. By keeping track of these indices, it can determine the bounding area of the object in the image.

Once the model is trained, the following steps are followed for each test image: Preprocessing, Segmentation, Classification, and Output Display. The preprocessing steps involve Gaussian blur, Canny Edge Detection, Grayscale conversion, and thresholding. The Segmentation module involves isolating each shape in images with multiple shapes. After segmentation, each shape undergoes classification individually, and the model saves the object orientation and other properties. The resulting video is a representation of world-like conditions, enabling users to visualize kinematics using animations.

Architectural Design:

The system architecture of the work comprises four key components: Screen Inputs, Physical Environment Variables, Environment, and Animations. Each component plays a vital role in enabling users to visualize complex physics concepts with ease. Each component is explored further in detail:

1. *Screen Inputs:*

The Screen Inputs component allows users to draw shapes directly on the screen. These hand-drawn shapes serve as inputs for the system. Convolutional Neural Networks (CNNs) are utilized to recognize and segment the shapes accurately. After segmentation and shape correction, the recognized shapes are added to the final diagram. Users have the flexibility to draw multiple shapes, and each shape goes through the same recognition process.

2. *Physical Environment Variables*:

The Physical Environment Variables component empowers users to manipulate the positions and orientations of shapes within the system. Users can reposition the shapes as required to create different configurations. Additionally, users can introduce physical variables into the system, such as friction coefficients, gravity, and other real-world factors. This feature allows users to explore the impact of different variables on the behavior of the simulated objects.

3. Environment:

The Environment component takes the recognized shapes and the defined physical variables and feeds them into the Initial Environment Creator. This creator utilizes an AI algorithm to generate a realistic and accurate representation of a real-world situation based on the provided inputs. This ensures that the simulation closely aligns with the desired scenario and provides an immersive learning experience.

4. Animations:

The Animations component takes the initial environment and applies the defined physical variables to simulate the behavior of the objects in the system. It generates a series of frames capturing the movement and interactions of the objects under the specified physical conditions. These frames are combined to produce a cohesive video output that visually represents the simulation in real-time. Users can observe and analyze the dynamic behavior of the objects, enhancing their understanding of complex physics concepts.

By integrating these four components, the system enables students and users to interactively visualize and comprehend intricate physics phenomena. The combination of hand-drawn inputs, customizable physical variables, AI-driven environment creation, and realistic animations fosters a comprehensive learning experience, making the comprehension of physics concepts more intuitive and engaging. This is explained diagrammatically in Figure 2.

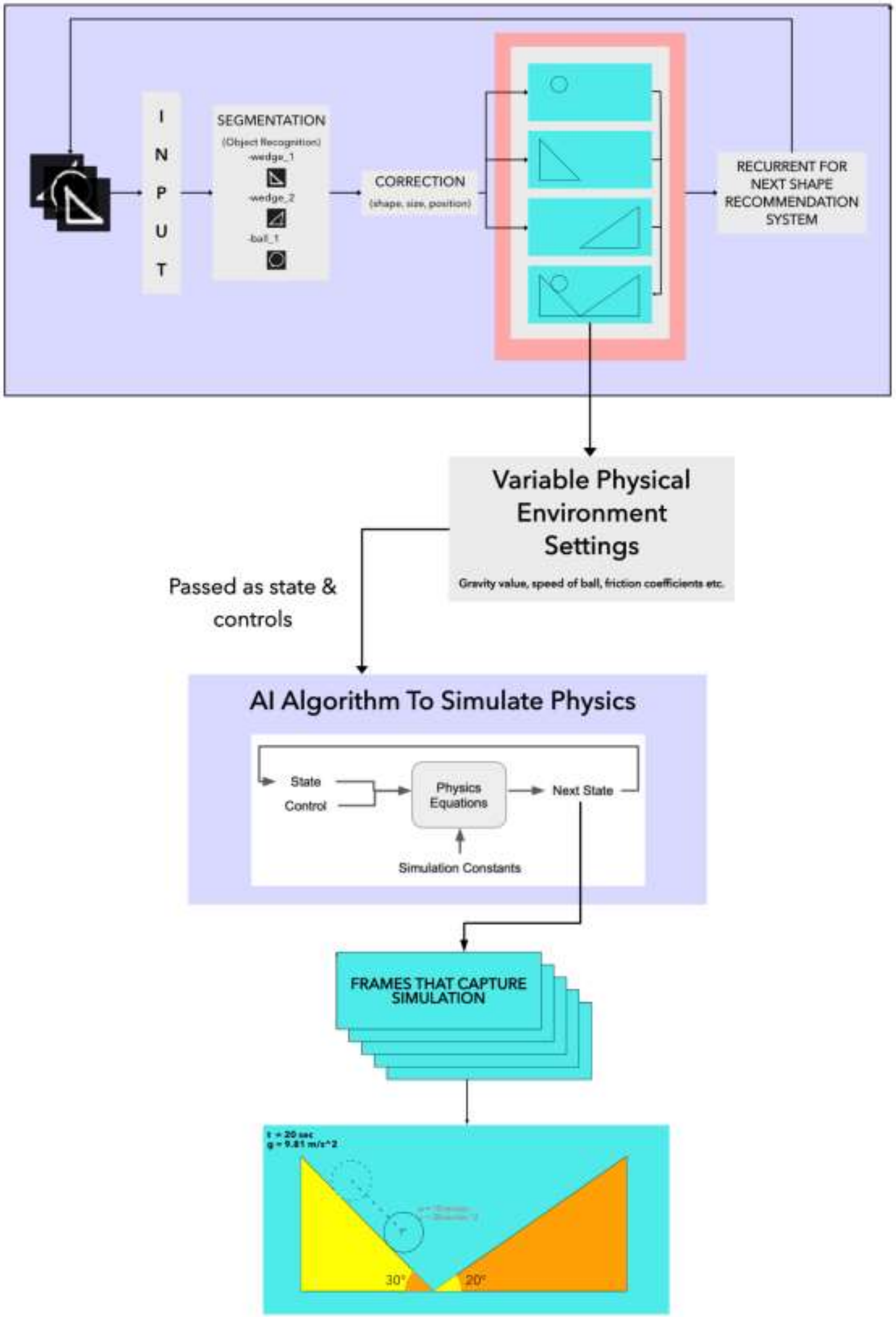


Fig 2. Design of the architectural model

User Interface Design:

The user interface design of the research work encompasses three fundamental components: Input Drawing on Canvas/Selection of Shapes, setting up Environment Variables, and Displaying of Output. These components facilitate a seamless and user-friendly experience for interacting with the system. Each component is explained further as:

1. *Input Drawing on Canvas/Selection of Shapes:*

This component of the user interface allows users to interact with the system by either drawing shapes directly on the canvas or selecting pre-defined shapes from a menu. By incorporating these options, the research work aims to enhance the ease of use and flexibility for users. Once the desired shape configuration is drawn or selected, the user can submit the input, which triggers the transition to the next step.

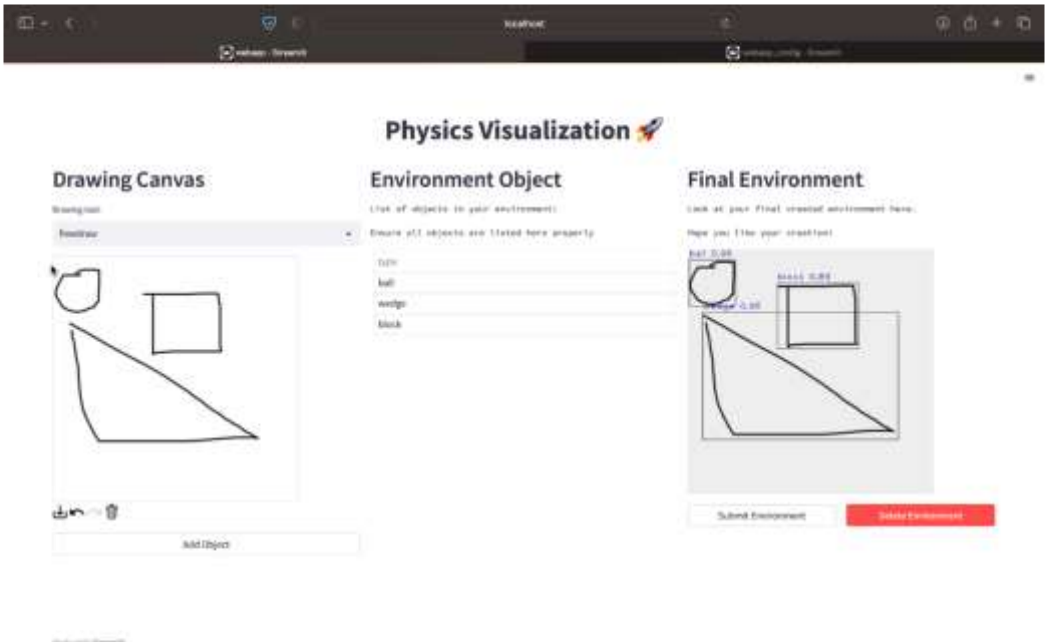
2. *Setting up Environment Variables:*

The Setting up Environment Variables component offers a menu-based user interface that enables users to specify various values for physical measurements. Users have the option to set custom values for parameters such as friction, mass, and gravity, allowing them to explore different scenarios and observe the effects of these variables on the system. Alternatively, default values may be provided for convenience. After setting up the environment variables, the user can submit the form, initiating the generation of the output.

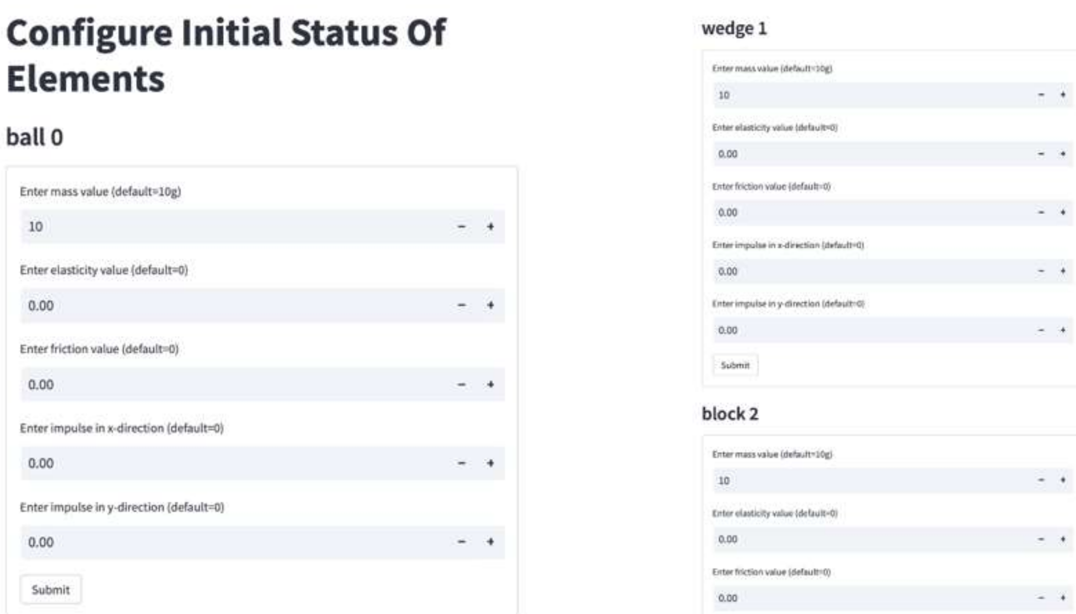
3. *Displaying of Output:*

The Displaying of Output component focuses on presenting the physical visualization of the system. It employs a frame-based animation approach, where the output is represented as a sequence of frames. Each frame depicts a distinct stage or moment of the physical visualization. These frames are displayed rapidly in succession, creating the illusion of fluid movement and providing a dynamic representation of the simulated physics scenario. Through this visual output, users can observe and analyze the behavior and interactions of the objects in the simulated environment.

By designing the user interface with these three components, the research work aims to provide a comprehensive and intuitive interaction platform as seen in Figure 3. The incorporation of input drawing, shape selection, environment variable customization, and frame-based animation output enhances the user experience and facilitates a deeper understanding of the underlying physics concepts.



(a) Drawing the shapes



(b) Setting Environment variables for each object

Fig 3. User Interface Design

5. RESULTS AND DISCUSSIONS

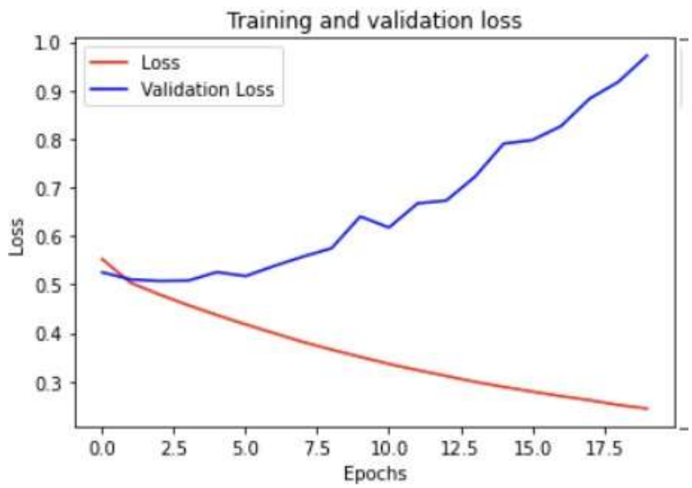
The shape recognition model underwent a comprehensive training process using different datasets to determine the most effective data source. The initial training attempts involved computer-generated data. However, during the validation phase, it became apparent that the model's performance was not satisfactory as shown in Figure 4 (a). The validation loss, which measures the model's accuracy on unseen data, kept increasing after each epoch. This indicated that the computer-generated data did not adequately capture the variability and complexity of hand-drawn physics diagrams, rendering it unsuitable for training the model.

Recognizing the limitations of computer-generated data, the research team shifted their focus to collecting data directly from individuals who hand-drew physics diagrams on sheets of paper. This dataset yielded more promising results compared to the computer-generated data as apparent in Figure 4 (b). The validation loss remained relatively constant, indicating that the model was able to learn and generalize from this dataset. While the performance was improved compared to the initial attempt, there was still room for further improvement.

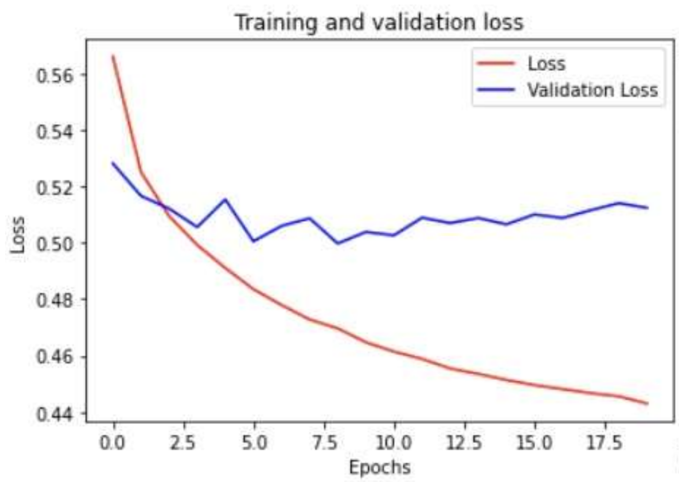
The final dataset utilized in the training process consisted of physics diagrams drawn directly on the app's canvas. This dataset resulted in the best performance among the three datasets. During the training and validation phases, the model's performance significantly improved, as evidenced by the decrease in validation loss after each epoch as seen in Figure 4 (c). This indicated that the model was effectively learning from the hand-drawn diagrams and could accurately recognize and classify the shapes depicted.

Based on these observations, the team concluded that the canvas drawn data was the most useful and appropriate for training the shape recognition model. The inherent variability and real-world nature of the hand-drawn diagrams on the canvas allowed the model to better capture the nuances and intricacies of physics concepts depicted in the diagrams. As a result, the model achieved higher accuracy and demonstrated the ability to generalize to unseen data.

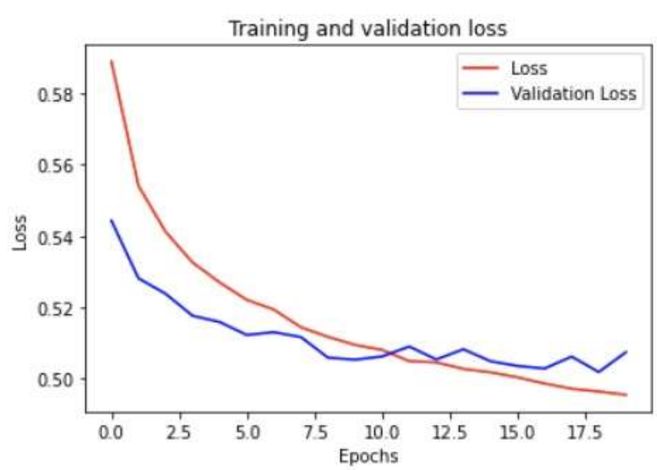
The decision to select the canvas drawn data for training the model was based on the combination of improved performance metrics, including reduced validation loss, and the model's ability to accurately recognize and classify shapes. This dataset provided the necessary foundation for the subsequent stages of the work, enabling the generation of accurate animations that visualized complex physics concepts.



(a) Computer Generated data



(b) Hand Drawn data



(c) Canvas Data

Fig. 4 Training and Validation Loss Graph for different datasets

Fig 5. Simulation of a drawn system in animation form

The implemented work successfully generates animations of hand-drawn objects, considering environmental variables such as gravity, friction, and mass (Figure 5). These animations serve as powerful visual aids that enhance students' understanding of physics concepts. By observing the animated objects in motion, students can comprehend how these variables influence the behavior and dynamics of the objects.

The animations accurately depict real-world scenarios, allowing students to witness the practical application of physics principles. For example, they can observe how objects accelerate under the influence of gravity, how friction affects the motion of objects on different surfaces, and how changes in mass impact the overall dynamics of a system. This visual representation provides students with a comprehensive and intuitive understanding of complex physics concepts.

The system's accuracy in recognizing hand-drawn objects and generating corresponding animations is a significant achievement. Through the integration of a trained Convolutional Neural Network (CNN) model, the system reliably identifies the shapes drawn by the students and applies the appropriate physics principles to animate them realistically. This ensures that the animations accurately reflect the intended physics concepts, enabling students to grasp the subject matter more effectively.

Furthermore, the system's user interface design facilitates a seamless user experience. Students can easily draw shapes on the canvas or select shapes from a menu, making the input process intuitive and user-friendly. Additionally, the interface allows users to set up various environmental variables and monitor specific measurements related to the objects' behavior, enabling a personalized and interactive learning experience.

Overall, the implemented work successfully bridges the gap between traditional teaching methods and visual representation in physics education. By providing students with the opportunity to draw hand-drawn physics diagrams and visualize their concepts through animations, the system enhances their engagement, interest, and understanding of physics principles. This tool holds great potential in making physics more accessible, interactive, and enjoyable for students, thereby improving their performance and overall learning experience in the subject.

6. CONCLUSION AND FUTURE SCOPE

6.1 Conclusion

In this work, a comprehensive and an effective tool for helping students visualise complex Kinematics concepts was developed. The aim to develop a machine learning model that bridges the gap between technology and the traditional learning approach in an intuitive manner is achieved. The tool is meant to be focused on making learning fun, interactive, and delightful for students, providing them newer ways to acquire and retain concepts.

Moreover, the proposed solution includes an intuitive user interface design that enhances ease of use for students. The design focuses on providing users with an immersive learning experience that is not only educational but also visually appealing. The interface provides options for users to draw shapes or select pre-drawn shapes on a canvas, set up environmental variables, and display physical visualizations using frames.

The architectural design of the system is well thought out, with a focus on providing a seamless user experience. It includes screen inputs, physical environment variables, an environment creator, and animations. This design ensures that the system is flexible, adaptable and can be used by different users, whether they are students, educators or researchers.

In conclusion, the proposed system is a significant step towards revolutionizing the way Kinematics concepts are taught. The integration of machine learning and a user-friendly interface in the tool will allow students to learn complex concepts in a fun and engaging way. It has the potential to provide a unique and immersive experience that is more efficient and effective than traditional learning methods. This work is a clear example of how technology can be harnessed to improve education, and it is expected to pave the way for more innovations in the field of education technology.

6.2 Further Scope

The future scope of this work is vast and promising. The tool can be further developed to incorporate more complex concepts like dynamics, electromagnetism, and quantum mechanics, among others. The integration of such concepts would help students to understand physics on a deeper level and enhance their problem-solving skills.

The model can be trained on a larger and more diverse dataset to overcome the issue of biased recognition of some shapes over others. Additionally, more advanced algorithms can be incorporated to reduce the misclassification of shapes with certain shades of colours.

Moreover, the tool can be integrated with virtual reality and augmented reality technologies to create an immersive learning experience for students. This will enable students to visualize complex concepts in a more interactive and intuitive manner.

In conclusion, the future scope of this work is extensive, and it has the potential to revolutionize the way students learn physics. The development of this tool is just the beginning, and with further advancements in technology, it is possible to create a more comprehensive and effective learning tool that makes learning physics more accessible, engaging, and enjoyable for students.

7. REFERENCES

[1] S. Gomez, "Shape recognition using machine learning," 2011 6th Colombian Computing Congress (CCC), 2011, pp. 1-3, doi: 10.1109/COLOMCC.2011.5936308.

[2] Ajeet Ram Pathak, Manjusha Pandey, Siddharth Rautaray "Application of Deep Learning for Object Detection"

[3] Zhao, G., Xie, L. (2008). Research on Kinematics Simulation of Parameterized Mechanism Based on Visualization in Scientific Computing. In: Li, D. (eds) Computer and Computing Technologies In Agriculture, Volume II. CCTA 2007. The International Federation for Information Processing, vol 259. Springer, Boston, MA. doi: 10.1007/978-0-387-77253-0_53

[4] Schroeder, D.V., 2018. Physics Simulations in Python.

[5] P. Payeur, Hoang Le-Huy and C. M. Gosselin, "Trajectory prediction for moving objects using artificial neural networks," in IEEE Transactions on Industrial Electronics, vol. 42, no. 2, pp. 147-158, April 1995, doi: 10.1109/41.370380.

[6] Li, C., Zhang, H., & Liang, C. (2018). Real-Time Physics-Based Animation with Machine Learning. ACM Transactions on Graphics, 37(6), 232. doi:10.1145/3272127.3275097

[7] Amir, N. A., Shafie, S. B., & Rahim, M. F. R. (2020). Physics Visualization for Virtual Reality: An Overview. Journal of Physics: Conference Series, 1529, 022061. doi:10.1088/1742-6596/1529/2/022061

[8] Gupta, P., Singh, S. V., & Patel, R. (2018). Interactive Visualization of Physics Simulations Using WebGL. 2018 International Conference on Advances in Computing, Communication Control and Networking (ICACCCN). doi:10.1109/ICACCCN.2018.8740917

[9] Mishra, S. K., & Pradhan, S. K. (2020). Augmented Reality for Enhancing Physics Education: A Review. Journal of Physics: Conference Series, 1593, 012011. doi:10.1088/1742-6596/1593/1/012011

[10] Wirthlin, M. E., & Ridgway, J. D. (2016). Physics Education with Python: Analyzing Data from Video. The Physics Teacher, 54(1), 24-26. doi:10.1119/1.4939884

[11] J. C. Garcia, G. A. Angulo, "Shape classification using deep convolutional neural networks," Expert Systems with Applications, vol. 72, pp. 319-327, 2017. doi: 10.1016/j.eswa.2016.10.042.

[12] M. Liu, et al., "Shape recognition using a hybrid neural network with improved learning algorithm," Neural Computing and Applications, vol. 31, no. 12, pp. 9141-9150, 2019. doi: 10.1007/s00521-018-3764-6.

[13] J. Liu, et al., "Deep shape recognition with depth images," Pattern Recognition, vol. 96, 106984, 2019. doi: 10.1016/j.patcog.2019.106984.

[14] M. Z. Khan, A. Majid, "Shape recognition using machine learning: A survey," Artificial Intelligence Review, vol. 52, no. 1, pp. 195-242, 2019. doi: 10.1007/s10462-018-9674-3.

[15] D. M. Green, D. A. Paterson, "Visualization in simulation: opportunities for the next generation," Proceedings of the 2003 Winter Simulation Conference, 2003, pp. 71-77. doi: 10.1109/WSC.2003.1261475.

[16] P. K. Sharma, K. Sharma, "Physics simulation and visualization of dynamical systems using Unity3D," 2016 International Conference on Computational Techniques in Information and Communication Technologies (ICCTICT), 2016, pp. 1-6. doi: 10.1109/ICCTICT.2016.7510056.

[17] K. Bodin, et al., "Visualizing physics simulations with Unity3D," 2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR), 2019, pp. 532-533. doi: 10.1109/VR.2019.8797796.

[18] M. Elsner, M. Zambelli, "Physics simulations with Unity3D for educational purposes," 2018 IEEE Global Engineering Education Conference (EDUCON), 2018, pp. 874-878. doi: 10.1109/EDUCON.2018.8363507.

[19] A. Subhan, "Physics simulation and visualization using MATLAB: Teaching undergraduate physics courses," 2018 International Conference on Computing, Mathematics and Engineering Technologies (iCoMET), 2018, pp. 1-6. doi: 10.1109/ICOMET.2018.8378356.

[20] N. Aspragathos, S. Sakka, "Physics-based animation using machine learning," 2020 15th International Conference on Engineering of Modern Electric Systems (EMES), 2020, pp. 1-6. doi: 10.1109/EMES51248.2020.9145274.