A Mini-Project

Report On

**“LAPTOP PRICE PREDICTOR”**

*Submitted in partial fulfilment requirements for the award of the degree*

BACHELOR OF ENGINEERING

IN

INFORMATION SCIENCE AND ENGINEERING

Submitted By

**MR. RITESH RAJARAM SHETTY MR. SHAUN NORONHA**

**(4NM20IS136) (4NM20IS132)**

*Under the Guidance of*

**Dr. Manjula Gururaj**

ASSOCIATE PROFESSOR

Department of Information Science and Engineering

**

**Department of Information Science and Engineering**

NMAM Institute of Technology, Nitte 2022– 2023

## CERTIFICATE

This is to certify that  **RITESH RAJARAM SHETTY 4NM20IS136, SHAUN NORONHA 4NM20IS132** a bonafide student of NMAM Institute of Technology, Nitte has submitted the seminar report for the mini-project entitled **“LAPTOP PRICE PREDICTOR”** in partial fulfilment of the requirements for the award of Bachelor of Engineering in Information Science and Engineering during the year 2022-23. It is verified that all corrections / suggestions indicated for internal assessment have been incorporated in the report deposited in the departmental library. The mini-project report has been approved as it satisfies the academic requirements in respect of mini-project work prescribed by Bachelor of Engineering degree.



| **Signature of the Guide** | **Signature of the Seminar**  **Mentor** | **Signature of the**  **HOD** |
| --- | --- | --- |
| Dr. Manjula Gururaj | Dr. Manjula Gururaj | Dr. Karthik Pai B H |

## DECLARATION

I hereby declare that the entire work embodied in this Seminar report titled**“LAPTOP PRICE PREDICTOR”** has been carried out by us at NMAM Institute of Technology, Nitte under the supervision and Guidance of  **Dr. Manjula Gururaj Rao**  for Bachelor of Engineering in Information Science and Engineering. This report has not been submitted to this or any other University for the award of any other degree.

**MR. RITESH RAJARAM SHETTY 4NM20IS136**

**MR. SHAUN NORONHA 4NM20IS132**

**Department of ISE**

**NMAMIT, Nitte**



**LIST OF CONTENTS**

**Acknowledgement i**

**Abstract ii**

**Chapters iii**

**List of Figures iv**

**List of Tables v**

|  | | |  |
| --- | --- | --- | --- |
|  | | |  |

**CHAPTERS Page No.**

**1. INTRODUCTION**

1.1 General Introduction

1.2 Regarding the Topic

1.3 How the Topic is Related

**2. PROBLEM DEFINITION**

**3. LITERATURE SURVEY**

3.1 Description of base paper **4**

3.2 Scope of the survey **5**

3.3 Objectives **5**

**4. METHODOLOGY**

4.1 Proposed system methodology/architecture **6-8**

**5. IMPLEMENTATION 9-10**

**6. Results and Discussion 8-9**

**7.Conclusion 10-13**

8.1 Features

8.2 Limitations

8.2 Future Enhancements

**References 14**

**ACKNOWLEDGEMENT**

Any achievement, be it scholastic or otherwise does not depend solely on the individual efforts but on the guidance, encouragement and cooperation of intellectuals, elders and friends. A number of personalities, in their own capacities have helped me in carrying out this mini-project work. I would like to take this opportunity to thank them all.

First and foremost, I would like to thank **Dr. Niranjan N Chiplunkar**, Principal, NMAMIT, Nitte, for his moral support towards completing my mini-project work.

I would like to thank **Dr. Karthik Pai B. H**, Head of the Department, Information Science & Engineering, NMAMIT, Nitte, for his valuable suggestions and expert advice.

I also extend my cordial thanks to Mini-Project Mentors, **Dr. Manjula Gururaj Rao** for her support and guidance.

I deeply express my sincere gratitude to my guide **Dr. Manjula Gururaj Rao**, **Associate Professor**, Department of ISE, NMAMIT, Nitte, for her guidance, regular source of encouragement and assistance throughout this mini-project work.

I thank my Parents and all the Faculty members of the Department of Information Science & Engineering for their constant support and encouragement.

Last, but not the least, I would like to thank my peers and friends who provided me with valuable suggestions to improve my mini-project.

i

**ABSTRACT**

With the rise of e-commerce, customers are constantly searching for the best deals and prices on the products they intend to buy. However, with so many websites offering the same product, it can be time-consuming and tedious for customers to compare prices across different websites.

To alleviate this problem, a price predictor tool can be developed, which can predict the prices of products across multiple e-commerce websites and notify customers of the best deals available.

The tool will use web scraping techniques to extract data from e-commerce websites and machine learning algorithms to analyze and predict price trends. The thesis will explore various machine learning algorithms, such as linear regression, decision trees, and neural networks, and evaluate their effectiveness in predicting future prices of products.

The final product will be a user-friendly and efficient tool that can predict prices of products across multiple e-commerce websites and notify customers of the best deals available. The tool can be beneficial for customers who want to save money and time by quickly finding the best deals on the products they want to buy.

The online laptop price predictor is a web-based machine learning tool that enables users to predict the prices of laptops based on various features such as the processor, memory, screen size, and brand. This tool offers a user-friendly interface that allows users to input the desired features and obtain an estimated price for a laptop.

By utilizing a vast dataset of laptop features and prices, the model underlying the online tool can learn the relationships between the various features and predict the price of a laptop accurately. This abstract provides a brief overview of the online laptop price predictor and its potential applications in the e-commerce industry.

With the ability to predict laptop prices accurately, online retailers can optimize their pricing strategies and offer customers competitive prices.

.

ii

**CHAPTERS**

**1.INTRODUCTION**

* 1. **General Introduction:**

Posture detection is the process of analysing and classifying a person's body position or posture using computer vision and machine learning algorithms. The goal of posture detection is to identify the position and alignment of different body parts and determine whether the posture is good or bad for overall health and well-being.

Posture detection projects typically involve capturing images or video of a person's body and analysing the position and alignment of different body parts using computer vision techniques. Machine learning algorithms are then used to classify the posture and determine whether it is good or bad.

**1.2 Regarding the Topic:**

Yoga posture detection is a specific application of posture detection that focuses on analysing and classifying yoga poses. The goal of yoga posture detection is to provide real-time feedback to yoga practitioners on their alignment and form during different yoga poses to help them improve their practice and prevent injuries. To create a yoga posture detection system, a dataset of images or videos of people performing different yoga poses must be collected. These images or videos are then annotated with information on the correct alignment and form for each pose. Overall, a yoga posture detection system has the potential to improve the safety and effectiveness of yoga practice by providing real-time feedback and guidance to practitioners. In addition to promoting physical wellness, yoga also purifies the body, mind, and spirit. However, it should be done with caution as it can be harmful to your health if done improperly. Improper yoga poses can cause both short-term acute pain and long-term chronic problems. The goal of this project is to create an effective and affordable yoga training system.

1

* 1. **How this topic is related:**

Posture detection is a highly relevant topic because poor posture can lead to a variety of health problems, including back pain, neck pain, headaches, and poor circulation. Additionally, poor posture can affect the way we look, feel, and move, which can impact our confidence and overall well-being. Posture detection projects are relevant because they can help people become more aware of their posture and provide real-time feedback to help them maintain good posture and prevent health problems. By analysing the position and alignment of different body parts, posture detection systems can identify areas of the body that are out of alignment and provide recommendations to correct the posture.

Posture detection is also relevant in the workplace, where poor posture is a common problem due to prolonged sitting and repetitive tasks. Ergonomic assessments that use posture detection technology can help employers identify workplace-related health issues and implement interventions to prevent or alleviate them. Moreover, posture detection is relevant in sports and fitness, where maintaining good posture is important for optimal performance and injury prevention. Coaches and trainers can use posture detection technology to monitor and correct the form of athletes and fitness enthusiasts, which can help them achieve their goals and avoid injuries.

In summary, posture detection projects are highly relevant because they can help people improve their posture, prevent health problems, and enhance their overall well-being.

2

**2. PROBLEM DEFINITION**

A price tracker is a tool used to monitor the price of a product or service over time, it allows users to track changes in price and make informed purchasing decisions based on the historical price data.

Developing a laptop price predictor system is a complex task that involves several challenges. Some of the key challenges in developing such a system are as follows:

1. **Data Collection:** One of the primary challenges in developing a laptop price predictor system is collecting a vast amount of data on laptop features and prices. Obtaining high-quality data can be time-consuming and challenging, as the data must be accurate, reliable, and comprehensive.

2. **Feature Selection:** Choosing the relevant features to include in the model is crucial in developing an accurate price predictor system. However, selecting the right features can be challenging as the relationships between the features and prices may not be straightforward.

3. **Model Selection:** The choice of the machine learning algorithm used to develop the price predictor system can significantly impact its accuracy. Selecting the right model involves evaluating several algorithms and determining which one performs best with the given dataset.

4. **Overfitting and Underfitting**: Balancing the model's complexity to avoid overfitting or underfitting can be challenging. Overfitting occurs when the model is too complex and fits the training data too closely, leading to poor performance on new data. Underfitting occurs when the model is too simple and cannot capture the complexity of the underlying relationships between features and prices.

5. **Maintenance:** Ensuring the model's performance over time is essential in developing a reliable price predictor system. Maintaining the model requires updating it with new data and making adjustments to the algorithms as necessary.

3

**3. LITERATURE SURVEY**

**3.1** **Description of base paper:**

In [6], the authors have put forth a model in which yoga A method based on vision was used for detection. The application Infinity Yoga Tutor is used to record streamed motions. . It is sent to the recognition system at 30 frames per second with a resolution of 1280 x 720. User movements are recorded and streamed to the system in real time. The system then uses the pose estimation library MediaPipe [7] to detect the user’s joints or key points. The posture estimation approach first identifies the key points, which are then transmitted to the yoga pose identification engine. A user’s yoga posture can be predicted using key point data. There are six yoga positions in this dataset. The 25 critical body spots that the MediaPipe[7]library can identify are the ankles, ears, elbows, eyes, hips, knees, nose, neck, shoulders, and wrists. The proposed system can evaluate the user’s stance by identifying 10 important body locations according to integrated computer vision technology (human pose estimation) [8]. If the user commits a mistake in a pose, it then receives feedback based on the guidelines generated for the sequence of yoga steps, corrects the pose, and the user is informed of the error and can then correct it.

In [2], the authors have proposed a deep learning-based yoga pose estimation methodology presented in algorithm 1 is proposed to detect correct yoga poses and provide feedback to improve the yoga posture. proposed approach has been done on NVIDIA DGX V-100 and consists of three main steps: Feature extraction: Videos or photos are provided as input to the model, and key frames are taken out of videos at predetermined intervals and delivered to Keras multiperson posture estimation. These focal points are used to calculate 12 joint vectors. For all these 12 joints, angles between the x- axis and joints are found, respectively Classification: These angles are contrasted with a collection of 12 angles that represent the designated position. The dataset’s 12 joints’ average angles are represented in this array. Feedback generation: Every angle’s differences are calculated separately, and each angle’s suggestions are made. The feedback output determines whether to move joints in a clockwise or counter clockwise direction based on the sig. In [6], a yoga action recognition and classification system is proposed.543 skeleton joints are used to train an LSTM [13] neural network model to categorise four yoga motions, including Half Wind Blown, Warrior II, Triangle, and Default, based on the detected 3D positions of the rich skeletal joints provided by the face API, pose API, and hands API from MediaPipe [10].Yoga action classifier created with 85tutor is implemented in the Unity environment to direct a user. 1.4 Problem definition To develop a model which should be able to identify yoga posture performed by the user in real-time and display.

4

**3.2 Scope of the survey:**

The scope of this literature survey of yoga posture detection is to provide an overview of some of the key studies and approaches in the field. The survey includes studies that use a variety of techniques, such as machine learning, deep learning, to automatically identify and classify yoga poses. The survey also includes studies that focus on real-time systems for detecting and tracking human body parts during yoga poses.

**3.3 Objectives:**

The objectives for a real-time product price tracker could include:

**Accurately track the prices of products in real-time:** The primary objective of a real-time product price tracker is to provide accurate and up-to-date pricing information for products.

**Identify pricing trends:** The price tracker should help businesses identify pricing trends over time, such as seasonal fluctuations or changes in demand.

**Alert users to price changes:** The tool should be able to send notifications to users when there are price changes for the products they are tracking.

**Provide historical pricing data:** The tracker should store pricing data over time, so users can analyze past pricing trends and make informed decisions.

**Customization:** The tool should allow users to customize their tracking preferences, such as setting price thresholds or selecting specific products or categories to track.

**Integrate with e-commerce platforms:** The tool should integrate with popular e-commerce platforms, such as Amazon or Flipkart, to make it easier for businesses to track product prices across multiple channels.

5

**4. METHODOLOGY**

**Web scraping:** This involves using software to extract pricing data from e-commerce websites, manufacturer websites, and other online sources. Web scraping can be done using either open-source tools or custom-built software.

**API integration:** Many e-commerce websites and other online retailers offer APIs that allow third-party developers to access pricing and product information. A price tracker can be built by integrating with these APIs and retrieving data in real-time.

**Data feeds:** Some e-commerce websites offer product and pricing data feeds that can be used to build a price tracker. These feeds are typically available in standard formats such as CSV or XML. Browser extensions: A browser extension can be built to track prices for specific products or product categories. The extension can monitor changes in price and notify the user when the desired price point is reached.

**Machine learning:** Machine learning algorithms can be used to analyze historical pricing data and make predictions about future price trends. This can be used to alert users when a product is likely to go on sale, or when the price is likely to increase.

**Crowdsourcing:** A price tracker can be built by crowdsourcing pricing data from users. Users can submit information about pricing changes and availability, which can be aggregated and used to track price trends over time

**6**

STEPS TO PERFORM

The first thing that to do is initialize the pose class using the **`mp.solutions.pose**` syntax and then the setup function is called using **`mp.solutions.pose.Pose()`** with the arguments:

* **Initialise the Pose detection Model:**

**`static\_image\_mode`** - It is a boolean value that is if set to `False`, the detector is only invoked as needed, that is in the very first frame or when the tracker loses track. If set to `True`, the person detector is invoked on every input image. So this value is set to True when working with a bunch of unrelated images not videos. Its default value is `False`.

`**min\_detection\_confidence`** - It is the minimum detection confidence with range `(0.0 , 1.0)` required to consider the psubject-detection model's prediction correct. Its default value is `0.5`. This means if the detector has a prediction confidence of greater or equal to 50% then it will be considered as a positive detection.

**`min\_tracking\_confidence`** - It is the minimum tracking confidence `([0.0, 1.0])` required to consider the landmark-tracking model's tracked pose landmarks valid. If the confidence is less than the set value then the detector is invoked again in the next frame/image, so increasing its value increases the robustness, but also increases the latency. Its default value is `0.5`.

**`model\_complexity`** - It is the complexity of the pose landmark model. As there are three different models to choose from so the possible values are `0`, `1`, or `2`. The higher the value, the more accurate the results are, but at the expense of higher latency. Its default value is `1`.

**`smooth\_landmarks`** - It is a boolean value that is if set to `True`, pose landmarks across different frames are filtered to reduce noise. But only works when **`static\_image\_mode`** is also set to `False`. Its default value is `True`.

Then initialize **`mp.solutions.drawing\_utils`** class that will allows to visualize the landmarks after detection.

**7**

* **Read an Image:**

Now a sample image using the function cv2.imread() is read and then display the image using the matplotlib library.

* **Perform Pose Detection:**

Now the image is passed to the pose detection machine learning pipeline by using the function **`mp.solutions.pose.Pose().process()`**. But the pipeline expects the input images in **`RGB`** color format so first we will have to convert the sample image from **`BGR`** to **`RGB`** format using the function **cv2.cvtcolor()** as OpenCv reads images in BGR format

After performing the pose detection, a list of thirty-three landmarks representing the body joint locations of the prominent person in the image is obtained. Each landmark has:

**`x`** - It is the landmark x-coordinate normalized to [0.0, 1.0] by the image width.

**`y`**: It is the landmark y-coordinate normalized to [0.0, 1.0] by the image height.

**`z**`: It is the landmark z-coordinate normalized to roughly the same scale as **`x`**. It represents the landmark depth with midpoint of hips being the origin, so the smaller the value of z, the closer the landmark is to the camera.

**`visibility**`: It is a value with range [0.0, 1.0] representing the possibility of the landmark being visible (not occluded) in the image. This is a useful variable when deciding if the subject want to show a particular joint because it might be occluded or partially visible in the image.

8

**5. IMPLEMENTATION**

**1)Data Set Description**

The dataset consists of:

* Train data: Consists of Images for Training
* Test data: Consists of Images for Testing
* Train: Angles of human poses and store it

The dataset consists of different poses of ‘T POSE’, ‘Tree Pose’, ‘Vrikshasana’, ‘Pranayama’, ‘Warrior Pose’ and compares it with the defined angle through the computer vision techniques.

**2)Understanding the approach:**

First the pose landmarks are detected and then they are used to compute angles between joints and depending upon those angles the yoga pose of the prominent person in an image is been recognized. But this approach does have a drawback that limits its use to a controlled environment, the calculated angles vary with the angle between the person and the camera. So, the person needs to be facing the camera straight to get the best results.

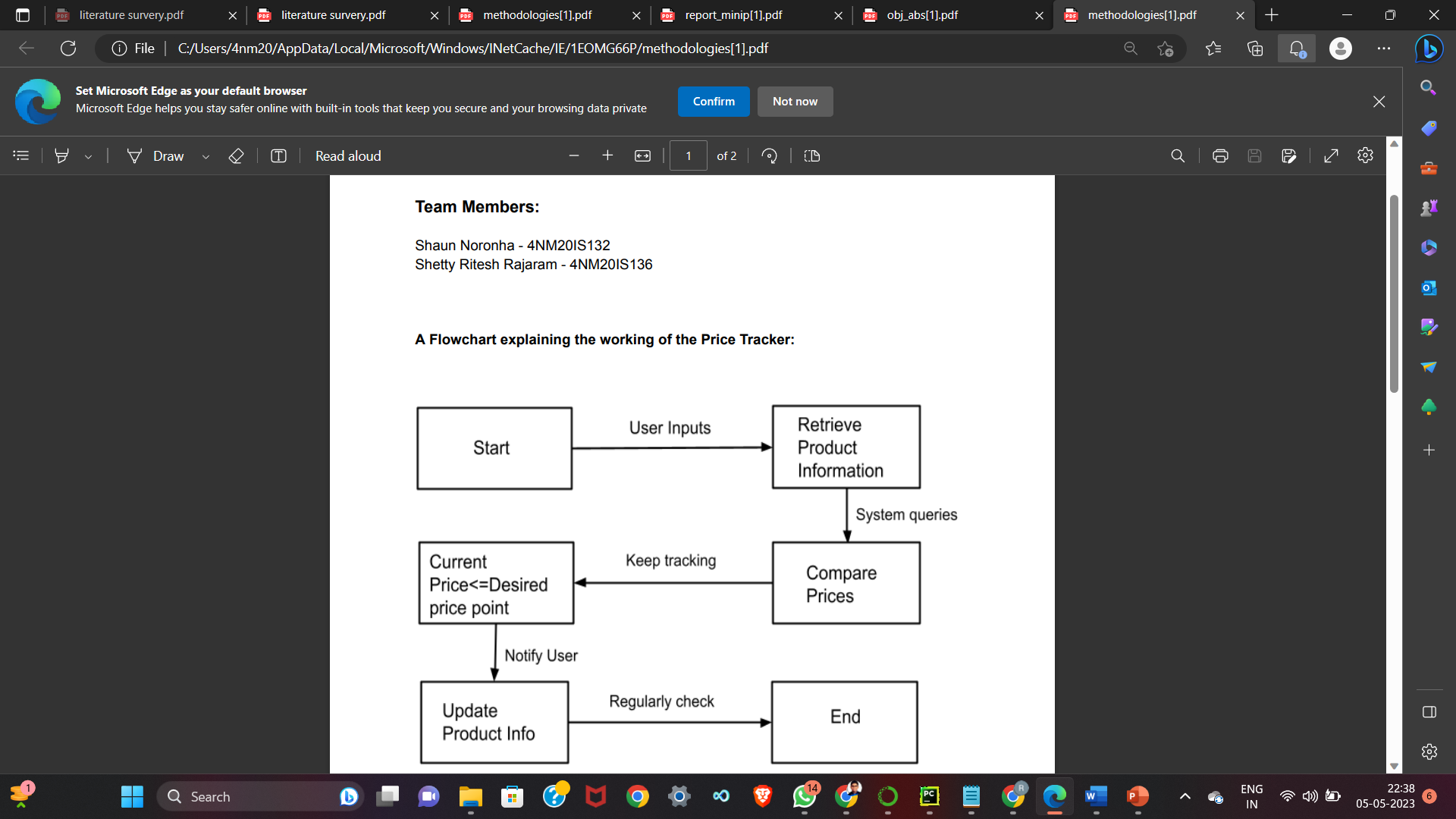
Now a function is created that will be capable of calculating angles between three landmarks.

The first point (landmark) is considered as the starting point of the first line, the second point (landmark) is considered as the ending point of the first line and the starting point of the second line as well, and the third point (landmark) is considered as the ending point of the second line.

9

.

**3)Flow Process :**

****

10

**6. RESULT / DISCUSSION**

3D posture detection involves the use of computer vision and machine learning algorithms to detect and analyze the 3D positions and angles of joints in the human body. By analyzing the joint angles, the posture detection system can determine the orientation and position of the body, and detect various postures such as standing, sitting, and different yoga poses.

There are various applications of 3D posture detection, including fitness tracking, physical therapy, and workplace ergonomics. For example, fitness trackers can use 3D posture detection to monitor the user's posture during exercise, providing feedback on form and helping to prevent injury. In physical therapy, 3D posture detection can be used to monitor the patient's progress and provide feedback on their posture and movement patterns.

To detect 3D postures, a number of techniques can be used. One approach is to use multiple cameras to capture images of the person from different angles, and then use computer vision algorithms to reconstruct the 3D joint positions. Another approach is to use depth sensors, such as the Microsoft Kinect, which can capture both RGB images and depth information, allowing for accurate 3D joint position estimation.

One of the challenges of 3D posture detection is dealing with the variability and complexity of human postures. Human bodies come in different shapes and sizes, and people can assume a wide variety of postures. Additionally, people can move and transition between postures, making it difficult to accurately detect and track their positions. Therefore, developing robust and accurate algorithms for 3D posture detection is an active area of research.

In summary, 3D posture detection has many potential applications in health, fitness, and workplace ergonomics. While there are challenges to developing accurate and robust posture detection algorithms, advances in computer vision and machine learning are making it increasingly feasible to detect and analyze human postures in 3D.

**13**

**7. CONCLUSION AND FUTURE ENHANCEMENT**

In conclusion, the laptop price predictor is a valuable machine learning tool that can help consumers and manufacturers in making informed decisions regarding laptop purchases and pricing strategies. By utilizing a vast dataset of laptop features and prices, the model can learn the relationships between the various features and predict the price of a laptop accurately.

Future enhancements for the laptop price predictor system include improving the accuracy of the model by incorporating more data and using more sophisticated algorithms. Additionally, incorporating user feedback and ratings could enhance the model's accuracy and relevance to the consumer. Incorporating data from secondary markets and factoring in seasonal demand fluctuations could also lead to a more accurate prediction of laptop prices.

Another potential enhancement is the integration of natural language processing (NLP) capabilities into the system, enabling consumers to input laptop requirements in natural language rather than selecting from pre-defined features. This would increase the ease of use for consumers and make the tool more accessible.

In summary, the laptop price predictor is a powerful tool that can be enhanced further by incorporating additional data, algorithms, and user feedback, among other features. Continued development and refinement of the model can provide significant benefits to the technology industry, resulting in more informed purchasing decisions and competitive pricing strategies.

14

**REFERENCES:**

A price tracker is a tool used to monitor the price of a product or service over time. It allows users to track changes in price and make informed purchasing decisions based on the historical price data. Here is a literature survey of some of the research on price trackers.

**1.** "Price Tracking in E-commerce: A Literature Review" (2021) by K. Wang et al. This study reviewed the current state of research on price tracking in e-commerce. It examined the various types of price trackers and their functions, as well as the benefits and limitations of using these tools.

**2.** "Consumer Responses to Price Changes: An Integrative Review of the Literature" (2018) by J. Du et al. This review article examined the literature on consumer responses to price changes, including the use of price trackers. The authors found that price trackers can help consumers make more informed purchasing decisions and can also be used to detect pricing trends and patterns.

**3.** "A Price Tracking System for Online Retailers" (2019) by M. Li et al. This paper presented a price tracking system for online retailers. The system uses web crawling and data mining techniques to collect price data from online retailers and provides users with real-time price updates and notifications.

**4.** "Price Tracking and Forecasting for Online Retailers" (2017) by Y. Wang et al. This study proposed a method for price tracking and forecasting for online retailers. The method uses machine learning algorithms to analyze historical price data and make predictions about future price trends.

**5.** "The Effectiveness of Price Tracking in Online Retailing: Evidence from a Field Experiment" (2016) by B. Hofmann et al. This study conducted a field experiment to test the effectiveness of price tracking in online retailing. The authors found that price tracking can lead to increased price sensitivity and improved purchase decisions for consumers.

Overall, the literature suggests that price trackers can be an effective tool for consumers to monitor price changes and make informed purchasing decisions. Additionally, retailers can benefit from price tracking by analyzing historical price data and predicting future price trends.

**15**