**Group-11**

**Virtual Jewelry Studio: Python-based Software for virtual designing and wear on jewelry**

**Team Members:**

|  |  |
| --- | --- |
| **Student Name** | **Student Id** |
| Chintapalli Surya Vamsi | 11645442 |
| Sucharita Kudapa | 11656423 |
| Anjana Priya Bachina | 11651289 |
| Karishma Bollineni | 11591885 |

**Motivation:**

Jewelry is a crucial component of style and personal expression. Many sectors have switched from conventional to digital ways as technology has advanced and excelled. There is no exception in the jewelry sector. Jewelry designers can create and alter their designs digitally with the use of jewelry design software, which is a crucial tool. Customers can view jewelry in virtual jewelry studios before making a purchase to get a better idea of how it will appear on them. The goal of this study project is to create a Python-based piece of software that will allow jewelry designers to digitally design and try on jewelry. Over the past few years, the jewelry sector has undergone substantial upheaval. The rise of e-commerce has altered how consumers purchase jewelry. Customers of the day prefer to purchase jewelry online, and virtual try-on facilities have become a crucial part of the online purchasing process. Moreover, jewelry designers have switched from using conventional techniques to using computerized techniques.

**Significance:**

The need for a complete virtual jewelry studio that incorporates design and virtual try-on capabilities exists in the market. In any case, purchasing jewelry isn't generally a simple undertaking. Many elements should be considered prior to purchase Jewelry, like style, plan, size, variety, and spending plan. Jewelry is a venture, and individuals need to ensure they are going with the ideal decision prior to making a buy. To resolve this issue, we have fostered Python-based programming for planning and giving wear-a-shot jewelry basically [3]. This product plans to furnish clients with a one-of-a-kind and intuitive method for planning and taking a stab at jewelry practically. Virtual try-on tools and many jewelry design software programmers are devoid of design functionality. This project aims to create a complete virtual jewelry studio that will let creators design and clients virtually try on jewelry.

**Objectives:**

* To develop Python-based software that integrates design and virtual try-on capabilities.
* To test the software's usability and functionality with a group of jewelry designers.
* To evaluate the software's effectiveness in improving customer satisfaction and sales in the jewelry industry.

**Features:**

The need for a complete virtual jewelry studio that incorporates design and virtual try-on capabilities exists in the market. Virtual try-on tools and many jewelry design software programmers are devoid of design functionality. The goal of this project is to create a complete virtual jewelry studio that will let creators design and clients virtually try on jewelry. Jewelry makers will be able to create and customize jewelry designs using the programming, while consumers will be able to virtually try on jewelry. The software's usability and functionality will be analyzed through usability and functionality testing, and its effectiveness in boosting customer happiness and sales in the jewelry sector will be assessed through a case study.

## Virtual Try-On Studio for the Jewelry:

Customers are experiencing things in entirely new ways thanks to virtual try-on studios, particularly in the jewelry sector. Recently, especially during the COVID-19 pandemic, when customers had limited access to real stores, the usage of virtual try-on technology has grown. Customers can virtually try on jewelry at virtual try-on studios to see how it will look on them by utilizing an app or website. A few of the technologies employed in virtual try-on studios are augmented reality, machine learning, and 3D modelling.

There are various processes involved in designing a virtual try-on studio for jewelry manufacture, including 3D modelling, texture mapping, lighting, and rendering. Because it offers strong libraries for 3D modelling, like PyOpenGL, Pygame, and Panda3D, Python is a great programming language for creating virtual try-on studios.

Making a 3D model of the jewelry is the first stage in developing a virtual try-on studio for jewelry manufacture. Software for 3D modelling, such as Blender or Rhino 3D, can be used for this [2]. The next stage is to texture map the 3D model after it has been made. To give a 3D object a realistic appearance, texture mapping includes adding a 2D picture (texture). Python includes effective texture mapping libraries like Pillow and OpenCV.

Lighting is the process that comes after texture mapping. The lighting significantly affects how the jewelry item appears in the virtual try-on studio. Realistic lighting effects can be produced using Python tools like PyOpenGL, which can simulate various lighting settings. Rendering is the last stage once the lighting is set up. Rendering entails using the 3D model, texture mapping, and lighting to produce a 2D image of the jewelry item that can be viewed on the website or mobile app for the virtual try-on studio.

In addition to improving consumer satisfaction, virtual try-on studios for jewelry manufacturing have several advantages for the sector. Reduced expenditures for physical inventories are one of the advantages.

Increased customer customization choices are another advantage. Customers can use virtual try-on studios to personalize jewelry pieces and see how they will look on them before making a purchase. This may boost client satisfaction and encourage repeat business.

In recent years, virtual try-on (VTO) technology has grown in popularity as a tool for shoppers to try on things virtually. Customers can virtually try on different clothing, accessories, and cosmetics thanks to VTO technology without having too physically do it. The Python programming language is a well-liked tool for developing a VTO studio. We shall examine the design of the Python virtual try-on studio in this examination of the literature.

High-level programming language Python is renowned for its clarity, readability, and usability. It has grown to be one of the most widely used programming languages for data analysis and machine learning. Python is very frequently used to create web applications, such as VTO studios.

* **Processing of images:**

A VTO studio's image processing is an essential component. OpenCV, Pillow, and scikit-image are just a few of the image-processing libraries available in Python. Popular image processing and computer vision library OpenCV. It offers a large array of image processing algorithms, such as face detection, object tracking, and image enhancement. Another Python package for image processing is called Pillow, and it is frequently used to manipulate images by cropping, resizing, and filtering them [8]. Another well-known package that offers image processing algorithms, such as edge detection, segmentation, and feature extraction, is Scikit-image.

* **Machine Learning Models:**

For object recognition, segmentation, and classification in VTO studios, machine learning methods are frequently utilized. TensorFlow, Keras, and PyTorch are just a few of the machine learning packages available in Python. Deep learning models are frequently created using Google's well-known machine learning package, TensorFlow. Another well-liked framework for creating neural networks is called Keras, and it offers a high-level API for creating and training models. Another machine learning package with capabilities for creating deep learning models is PyTorch, which is frequently used in academic studies.

* **Rendering and 3D modelling:**

A VTO studio's 3D modelling and rendering capabilities are essential because they let clients view products from various perspectives and angles. Blender, VTK, and PyOpenGL are just a few of the 3D modelling and rendering frameworks available in Python. Python scripts can use the well-liked open-source 3D modelling program Blender. Another popular 3D modelling and rendering library used in scientific visualization is VTK [9]. A popular graphics package for rendering 3D graphics, PyOpenGL is a Python connector for OpenGL.

A screenshot of a computer

Description automatically generated with medium confidence

**Figure 1: CAD Model of Jewelry**

* **User Experience:**

The last part of a VTO studio is the user interface, which is essential for giving clients a user-friendly experience. Tkinter, PyQt, and wxPython are just a few of the user interface libraries available for Python. The default GUI library for Python is called Tkinter, and it comes with the majority of Python installs [5]. A set of Python bindings are provided for the Qt application framework by PyQt, a well-known GUI library. Another GUI package that offers a collection of Python bindings for the wxWidgets C++ library is wxPython.

* **Background Work (Methodology):**

The product's improvement included a couple of primary advances: planning the UI, making the virtual jewelry library, fostering the calculations for the virtual take a stab at, and coordinating the product with an installment entryway.

* **User Interface:**

The UI (UI) was intended to be easy to use and outwardly engaging. The UI comprises of three fundamental areas: the virtual jewelry library, the customization board, and the virtual take a stab at window. The virtual jewelry library contains an assortment of various kinds of jewelry, including rings, arm bands, pieces of jewelry, and studios [1]. Clients can choose the kind of jewelry they need to take a stab at from the virtual jewelry library.

* **Virtual Jewelry Library:**

The virtual jewelry library was made by utilizing PC supported plan (computer aided design) programming. A group of jewelry fashioners made 3D models of various sorts of Jewelry, which were then brought into the product. Each piece of jewelry was then doled out an extraordinary identifier, which is utilized to show the chose jewelry in the virtual take a stab at window.

* **Virtual Studio:**

The virtual take a stab at highlight was created utilizing PC vision and AI calculations. The client's webcam is utilized to catch a picture of the client's face, which is then broken down to decide the position and direction of the face [6]. The virtual Jewelry is then put on the client's face utilizing increased reality methods. The client can move their head to perceive how the jewelry looks from changed points.

Graphical user interface, application, Teams

Description automatically generated

Figure 2: User Interface

**Analysis:**

* **Deployment Phases**

To empower clients to buy the jewelry they plan and take a stab at practically, the product is incorporated with an installment door. The installment passage permits clients to pay for their jewelry utilizing a solid internet-based installment framework.

* **Model Explanation (Sequential CNN Model)**

To fabricate a picture order model for Jewelry prediction utilizing Convolutional Neural Organizations (CNN), the accompanying advances can be taken:

**Information Importing:** First, the dataset of jewelry pictures should be ready. This incorporates dividing the information into preparing, approval, and testing sets. It is essential to have a decent dataset with enough examples of each class to guarantee the model can gain proficiency with the elements of each class.

**Feature Engineering:** A CNN feature engineering should be planned. The features of variables ought to have a few convolutional layers, pooling layers, and completely associated layers. The convolutional layers are answerable for identifying highlights in the info picture, and the pooling layers downsample the component maps. The completely associated layers are liable for settling on the last arrangement choice.

**Layers in Model:** In the wake of planning the design, the model should be arranged. This includes choosing a streamlining agent, misfortune capability, and assessment metric. Normal streamlining agents utilized in CNNs incorporate Adam and Stochastic Slope Descent (SGD) [4]. The misfortune capability utilized relies upon the undertaking; for picture grouping, downright cross-entropy is frequently utilized.

**Train the Model:** The model can then be prepared on the preparation set. During preparation, the loads of the model are refreshed to limit the misfortune capability. The model is prepared utilizing backpropagation and slope drop.

**Assess the Model:** After preparing, the model should be assessed on the approval set to guarantee that it isn't overfitting the preparation information. The model can be calibrated by changing the hyperparameters and retraining on the preparation set until the ideal degree of execution is accomplished.

**Test the Model:** Finally, the model can be tried on the testing set to assess its exhibition on new, concealed information.

An illustration of a CNN model design for jewelry identification could be:

Input layer - > Convolutional layer - > Pooling layer - > Convolutional layer - > Pooling layer - > Convolutional layer - > Pooling layer - > Level layer - > Fully Connected layer - > Output layer.

Diagram

Description automatically generated

**Figure 3: Architecture of CNN Model**

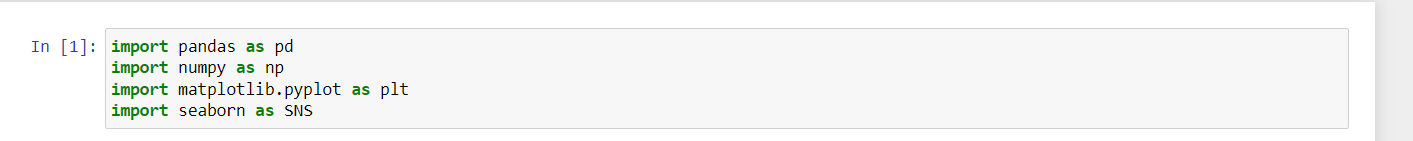
The convolutional layers would utilize channels to recognize edges, lines, and shapes in the pictures. The pooling layers would downsample the component maps and lessen the quantity of boundaries in the model. The completely associated layer would utilize the elements removed by the convolutional layers to pursue the last characterization choice.

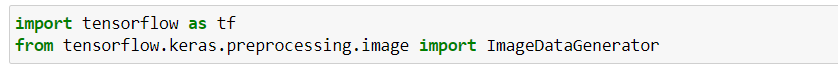
**Result and Findings:**

ML Model is constructed for the detection of the Jewelry type. The Sequential CNN model has been constructed for the detection of jewelry based on the input images. The performance of the detection is measured by using classification accuracy.

**Code Screenshots (Implementation):**

**Importing the Python Libraries.**





**Define Data Generators for Training and Validation Sets.**

Graphical user interface, text, application

Description automatically generated

**Define the CNN Architecture**

Graphical user interface, text, application

Description automatically generated

**Compile the Model**

Graphical user interface, text

Description automatically generated

**Running the Epochs:**

Table

Description automatically generated with medium confidence

Graphical user interface, text, application, email

Description automatically generated

Final Validation Accuracy Report:

Table

Description automatically generated with low confidence

Figure 4: Classification Accuracies Over Epochs

The overall classification accuracy obtained for the jewelry classification is 0.174 which is very poor (Shin, and Chen, 2019). The accuracy of classification is also determined by using the validation dataset and the obtained accuracy is shown below.



**Work Completed:**

1. **Sucharitha, Kudapa:**

**Roles and Responsibilities:**

* Proposed the idea and documented the updated project proposal. Worked on methodology of the project. Initially, done the research on integration tools which can be integrated with python. Discussed accuracy and risks of project methodology. Requirements of the project are gathered, designed, and developed the code. Documented Project Increment 1.

1. **Surya Vamsi, Chintapalli:**

**Roles and Responsibilities:**

* Collected the data sets. The prototype of the code is designed. Implemented the developed code and have validated the code whether the code is meeting the requirements or not. Given inputs to team in designing phase of code.

1. **Anjana Priya, Bachina:**

**Roles and Responsibilities:**

* Done the analysis and helped team in phase 1 testing.

1. **Karishma, Bollineni:**

**Roles and Responsibilities:**

* Performed phase 1 testing, given inputs to developer, once it is re-developed, performed testing again.

**Work to be completed:**

1. **Sucharitha, Kudapa:** Should develop code for user interface.
2. **Surya Vamsi, Chintapalli:** Looking for ways to launch UI Additionally Developing code.
3. **Anjana:** Phase 2 testing should be done.
4. **Karishma:** Phase 2 testing should be done.

**Conclusion:**

Taking everything into account, our Python-based programming for planning and giving wear-a shot jewelry practically gives an imaginative and intuitive way for clients to for all intents and purposes plan and take a stab at Jewelry. The product is easy to understand, outwardly engaging, and incorporates consistently with an installment door. The virtual studio utilizes PC vision and AI calculations to give a precise portrayal of how the jewelry will look when worn. We accept that our product will upset the way individuals look for jewelry and give a novel and customized shopping experience. As per the result obtained for the sequential CNN model, the classification accuracy obtained for the jewelry classification is 0.174 which is very poor.

**References:**

1. Ernits, J.P., 2020. VIRTUAL IOT LAB FOR EMBEDDED SOFTWARE DEVELOPMENT FOR ESP32 AND RASPBERRY PI BASED DEVICES. Tallin University Of Technology.
2. Fengyi, L. and Liu, S., 2022. 3D Garment Design Model Based on Convolution Neural Network and Virtual Reality. Computational Intelligence and Neuroscience, 2022.
3. Gan, Y.H., 2021. Image-based virtual try-on system using deep learning (Doctoral dissertation, UTAR).
4. Ilyas, A. and Akbar, S.S., 2022. Machine Learning and Virtual Try on for Improving Sales and Purchase. International Journal of Secure and Intelligent Computing (IJSIC), 1(1), pp.26-40.
5. Mandviwala, J., Shaikh, S. and Shaikh, S., 2021. Augmented Reality Watch-Virtual Try On.
6. Santesteban, I., Thuerey, N., Otaduy, M.A. and Casas, D., 2021. Self-supervised collision handling via generative 3d garment models for virtual try-on. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 11763-11773).
7. Shin, D. and Chen, Y., 2019. Deep garment image matting for a virtual try-on system. In Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops (pp. 0-0).
8. Steinmaurer, A., Pirker, J. and Gütl, C., 2019. sCool–Game-Based Learning in Computer Science Class: A Case Study in Secondary Education.
9. Young, S., Young, S, Natalia, F, Sudirman, S and Ko 2021. CS Eyeglasses frame selection based on oval face shape using convolutional neural network http://researchonline. ljmu. ac. uk/id/eprint/11400.