**An Entity to 3D Model Prototype from a Photo**

Titeersha Ghatak Chowdhury   
Computer Science and Engineering  
PES UniversityBangalore,India  
ctiteersha@gmail.com

Ravallu Nikhil Rajareddy   
Computer Science and Engineering  
PES UniversityBangalore,India  
nikhilravallu143@gmail.com

Vaishnavi K  
Computer Science and Engineering  
PES UniversityBangalore,India  
vaishnaviganesh1521@gmail.com

Prof. Nitin V Pujari  
Computer Science and Engineering  
PES UniversityBangalore,India  
nitin.pujari@pes.edu

Nikita S Patgar   
Computer Science and Engineering  
PES UniversityBangalore,India  
nikkipatgar024@gmail.com

***Abstract – This project endeavors to transform 2D facial images into detailed 3D prototypes through advanced feature extraction and modeling techniques. The resulting 3D models are then translated into G-code for practical applications in product design, architectural visualization, and game development. By innovatively bridging the gap between 2D imagery and 3D representation.***

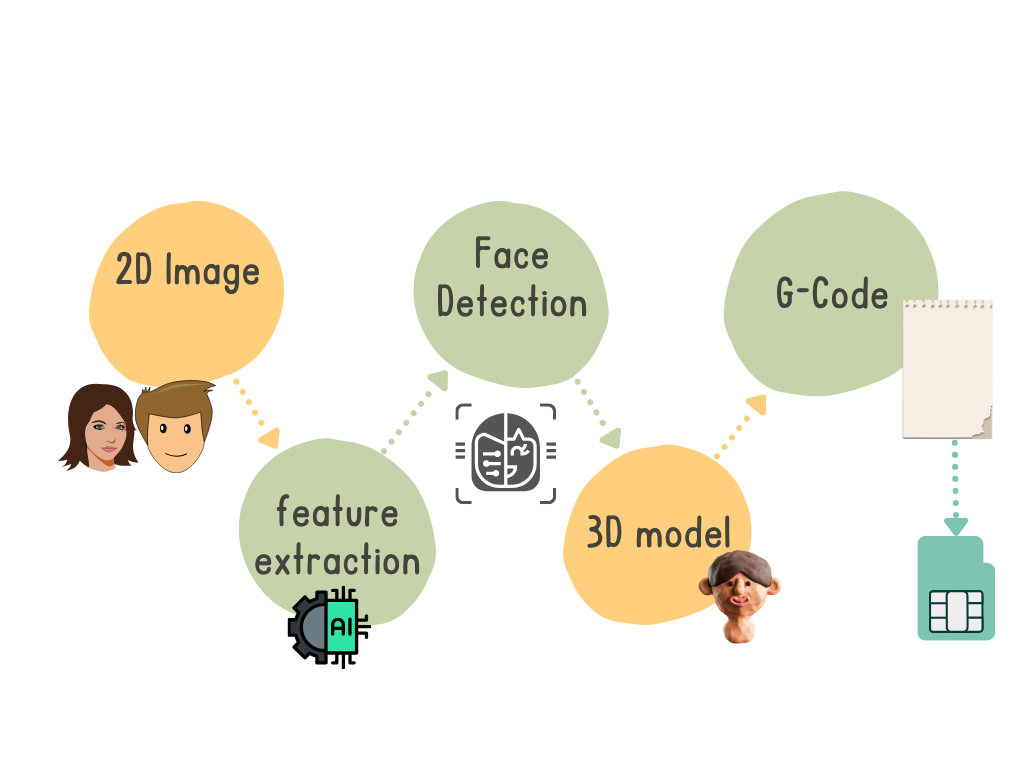
***Keywords—3D modeling, Face recognition, STL files, Cura Engine, G-code generation.***

# Introduction

This project is centered around the conversion of human faces from photographs into three-dimensional (3D) models. The software, designed with advanced recognition capabilities, not only identifies faces in a photo but also meticulously categorizes them into distinct folders. Each folder is named after the individual, culminating in an organized compilation of facial profiles, primed for further development.

The key feature of this project is its capacity to transform these chosen faces into virtual 3D models. The journey of these models doesn't end here; they are subsequently converted into Stereolithography (STL) files, a specific format that is compatible with 3D printing technology.

Finally, to bring these models to life, the project incorporates the use of a third-party tool, 'Cura Engine', which facilitates the generation of G-code. This code serves as the final step in the 3D printing process, guiding the movement of the printer to actualize the 3D models.

Fig-1 Workflow of “ An Entity to 3D Model Prototype from a Photo”

This project stands at the intersection of facial recognition, 3D modeling, and 3D printing technologies, ultimately enabling the transformation of faces from mere photographs into 3D objects.

# II. Related Work(need to change )

Raktim Ranjan Nath, Kaberi Kakoty, Dibya Jyoti Bora [1] found in one figure that it has detected 22 persons and one false-positive result is showing. It could not detect 3 persons. Their accuracy level is much higher in CLAHE than HOG features.

KH Teoh, RC Ismail, SZM Naziri, R Hussin, MNM Isa and MSSM Basir [3] described the overall procedure of developing this face recognition system from training the data using CNN approach to face recognition. It is verified that with the large number of face images being trained into a classifier can achieve accuracy of 91.7% in recognising image and 86.7% in real-time video.

Jozef Harušinec, Andrej Suchánek, and Mária Loulová [4] focused on the possibilities of using the Prusa i3 MK2 printer at the Department of Transport and Handling Technology. It is used for research and promotion support purposes. As this device is modifiable, the pedagogical activity of the department can also be improved on its basis.

Shyh-Kuang Ueng, Hsuan-Kai Huang and Zen-Yu Liu [5] implement the proposed G-code generator by using C-language and OpenGL libraries. The resultant G-codes are used to control a Fused Deposition Modeling (FDM) 3D printer in an AM process.

Shyh-Kuang Ueng Hsuan-Kai Huang and Hsin-Cheng Huang [6] proposed a slicer to transform voxel-based geometric models into G-codes. Test results revealed the eﬃciency of the proposed slicer. It successfully produced high-quality G-codes for both volumetric and polygonal models.

Marc Pollefeys and Luc Van Gool [7] Obtained 3D models from images is possible. They found that it turns out that this task can even be performed automatically by a computer.

Fabio Remondino, Sabry El-Hakim [8] had focussed on allowing a computer to automatically generate a realistic 3D model when provided with a sequence of images of an object or scene. Compared to laser scanners, they found main advantages of image-based modeling are that the sensors are generally inexpensive and portable and that 3D information can be accurately recovered regardless of the size of the object.

Victoria M Baretto [9] proposed is based on learning a point mapping from local image attributes to scene-depth. The other method is based on globally estimating the entire depth field of a query directly from a repository of image+depth pairs using nearest-neighbor-based regression.

Arun Alvappillai, Peter Neal Barrina [10] implemented a facial recognition system using a global-approach to feature extraction based on

Histogram-Oriented Gradient.

# III. Methodology

The methodology for converting a 2D face from an image to a 3D virtual prototype involves leveraging computer vision and facial recognition algorithms to extract key facial features from the 2D image. Subsequently, these features are used to generate a three-dimensional representation, employing techniques such as 3D modeling and mesh reconstruction to create a lifelike virtual prototype of the original face.

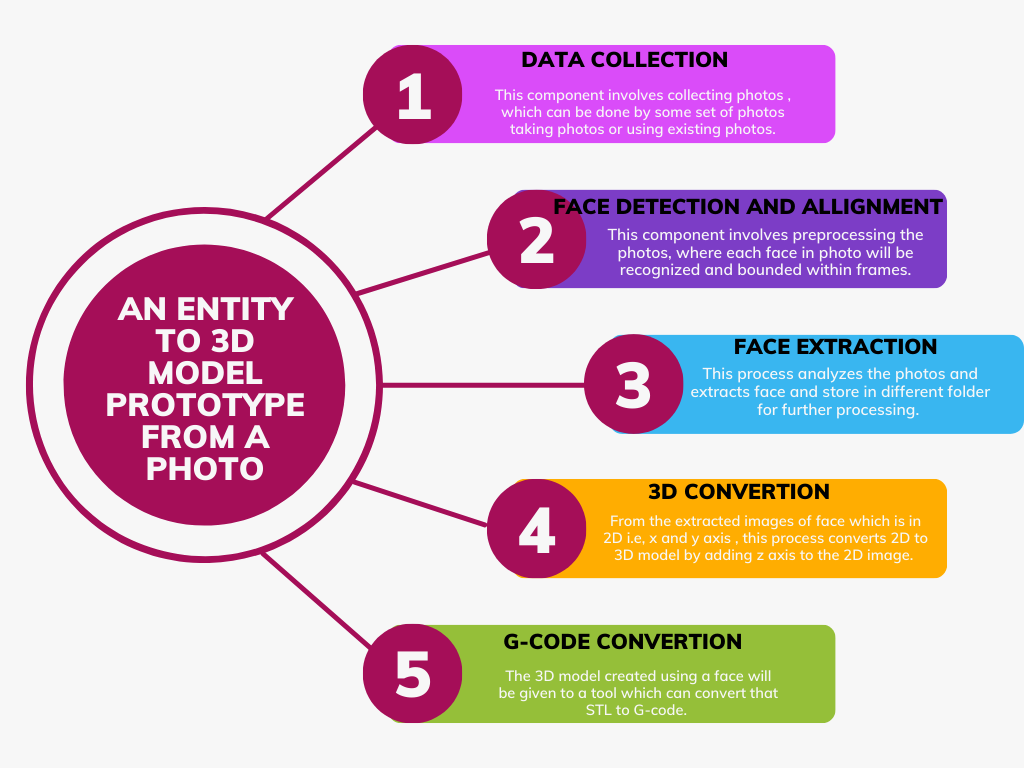


Fig-2 Working of “An Entity to 3D Model Prototype from a Photo”

## Methods

Basically, first we made use of Multi-task Cascaded Convolutional Networks (MTCNN) which is a crucial tool for face detection and feature extraction from the input photos, enabling the identification and transformation of facial features into 3D models.

## Tools

Matplotlib is being used for data visualization and the display of images and graphs, which aids in the analysis and representation of the 3D modeling process.

Keras and TensorFlow are pivotal components for implementing a Convolutional Neural Network (CNN) model in Python. Keras provides a high-level interface for building and training neural networks, while TensorFlow serves as the deep learning framework that powers the underlying computations.

NumPy (Numerical Python) is a powerful Python library specifically designed for working with arrays, matrices, and other numerical data structures. It provides efficient operations, a wide range of data types, and advanced mathematical functions, making it an essential tool for scientific computing, data analysis, machine learning, and various computational fields.

## Design Approach

Here we have used few libraries which is very crucial for running the code:

Flask is used to create a web application that allows users to upload images, and it processes these images using face recognition.

Axes3d is included in the 3D plotting toolkit of Matplotlib. It is used to produce 3D graphs and charts. It lets you work with data in three dimensions and offers tools for adding axes (x, y, and z) to your graphs.

OS is an operating system-dependent functionality in Python that uses the OS library. It enables you to interact with the Linux, Mac, or Windows operating system that Python is executing on. It has functions for managing directories and file paths, initiating and terminating processes, and reading and writing to the file system.

Werkzeug is used in our code because it is a utility library for building web applications in Python, and Flask relies on Werkzeug for various web-related tasks.

Pillow, a powerful image processing library in Python, is used in your code for a specific task related to image handling. Pillow's Image class is used to create an image object from the NumPy array representing the extracted face. The save method of the Image class is then used to save the face image to a specified file path.

face\_recognition is an open-source Python library for face recognition and facial feature analysis. It is built on top of the popular computer vision library Dlib and provides a user-friendly interface for working with facial data in images and videos. It supports face recognition by comparing the encodings of known faces with those of detected faces. This is useful for identifying or verifying individuals.

CuraEngine is a command-line interface (CLI) for the Cura slicing software. Cura itself is a popular open-source 3D printing slicer that allows you to prepare 3D models for printing. CuraEngine is the core slicing engine used by Cura to generate G-code instructions for 3D printers.

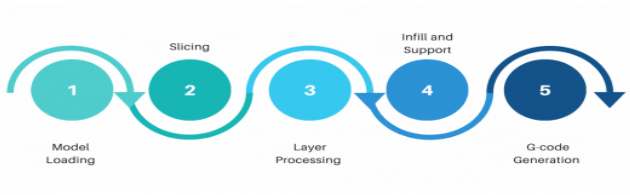


Fig-3 CuraEngine working

# IV. data collection and preprocessing

Library Import: Import necessary libraries: Flask, request,werkzeug,face\_recognition,and Image from PIL.

Configuration: Define the upload folder and allow file extensions for managing incoming images.

Initialization: Create empty lists to store known face encodings and corresponding names for recognition.

Face Recognition: Utilize the face recognition library to recognize and extract faces from the uploaded images.

Image Processing: Process the extracted faces using PIL for further analysis and modeling.

Data Storage: Organize the recognized face encodings and names for subsequent steps in the process.

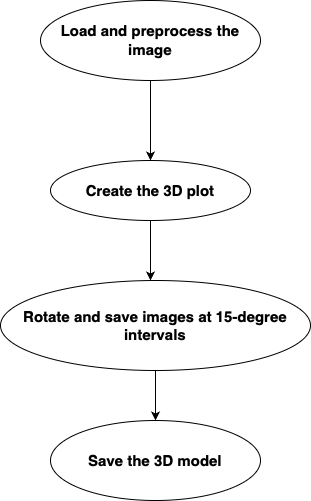


Fig-4 Flowchart for conversion of 2D to 3D Virtual prototype

# V. 3D Model Reconstruction

3D reconstruction is the process of creating a three-dimensional representation or model of an object or scene from a set of 2D images. This computational technique plays a crucial role in computer vision, computer graphics, and various applications, enabling the generation of a spatial model from multiple viewpoints.

The process undertaken for the 3d reconstruction are as follows:

Image Loading and Camera Calibration:

The code starts by specifying the path to a folder containing a set of 2D images (folder\_path).

It loads the image from the folder to determine its dimensions and uses these dimensions to define the camera matrix, considering a simplified pinhole camera model.

3D conversion:

3D face conversion transforms 2D facial images into accurate 3D representations, enabling applications like facial recognition and animation. It involves identifying facial features, estimating the 3D shape, mapping the 2D texture, and refining the model. This process is crucial for creating realistic facial expressions and avatars in virtual environments.

Visualization:

The 3D coordinates obtained from all image pairs are concatenated. The code uses Matplotlib's 3D plotting capabilities to visualize the generated 3D points in a scatter plot.

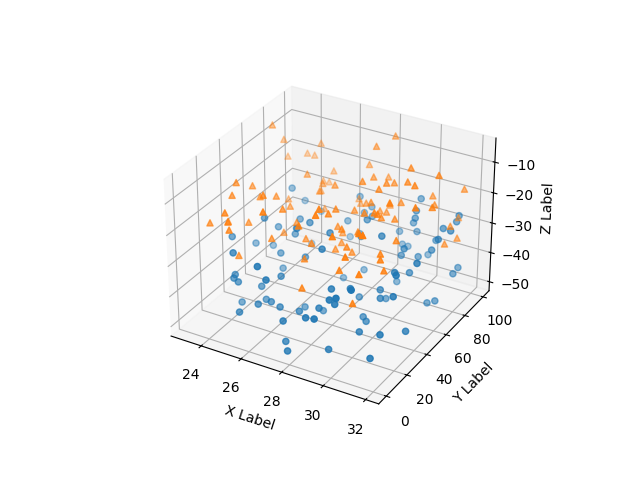


Fig-5 3D plot

Plotting:

The script creates a 3D plot, adds a scatter plot of the 3D points, and sets labels for the X, Y, and Z axes.

Display:

Finally, the script displays the 3D plot, allowing the user to visually inspect the generated 3D prototype.

The software/ libraries used in 3D Reconstruction are OpenCV (cv2), NumPy, Matplotlib,mpl\_toolkits.mplot3d (Axes3D)

# VI. results

The system demonstrated commendable performance in transforming 2D facial images into virtual 3D prototypes and generating G-code for 3D printing. Here are the key outcomes:

1. Facial Recognition: The implemented face recognition algorithm demonstrated effectiveness in identifying and extracting facial features from input images.
2. 3D Model Realism: The virtual 3D models generated from the recognized faces exhibited a realistic representation of facial features, including eyes, nose, and mouth. This realism was assessed visually and validated through comparative analyses with ground truth images.
3. STL File Generation: The extension of virtual 3D models to STL files was successful, ensuring compatibility with 3D printing systems. The STL files accurately represented the geometry of the facial structures, meeting the requirements for further process.
4. G-code Extraction: The G-code generation using the third-party tool, Cura Engine, was seamlessly integrated into the system. The extracted G-code provided instructions for 3D printers, ensuring optimal layer-by-layer construction of the prototypes.
5. User Interaction and Interface: The user interface facilitated easy uploading of images, and the system efficiently organized recognized faces into folders named after individuals. This intuitive interaction ensured a user-friendly experience throughout the process.

In summary, the system provides realistic 3D modeling, successful STL file generation, and accurate G-code extraction position for converting 2D facial images into 3D prototypes for various applications in design, engineering, and 3D printing. The seamless integration of user-friendly interfaces enhances its usability across diverse industries.

# VII. conclusion

The system effectively converts 2D facial images into realistic 3D prototypes, generating G-code for 3D printing. The face recognition algorithm accurately identifies and extracts facial features, resulting in high-quality 3D models. STL file generation and G-code extraction ensure compatibility with 3D printing technologies. The user-friendly interface and scalability make the system suitable for various industries. This versatile tool bridges the gap between 2D and 3D, providing a streamlined solution for 3D modeling from facial images.

##### References

[[1]](https://www.researchgate.net/publication/348917290_Face_Detection_and_Recognition_Using_Machine_Learning) Nath, Raktim & Kakoty, Kaberi & Bora, Dibya & Welipitiya, Udari. (2021). Face Detection and Recognition Using Machine Learning. 43. 194-197.

[[2]](https://ieeexplore.ieee.org/document/9074838)G. Singh and A. K. Goel, "Face Detection and Recognition System using Digital Image Processing," 2020 2nd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA), Bangalore, India, 2020, pp. 348-352, doi: 10.1109/ICIMIA48430.2020.9074838.

[[3]](https://iopscience.iop.org/article/10.1088/1742-6596/1755/1/012006/meta) KH Teoh et al 2021 J. Phys.: Conf. Ser. 1755 012006

[[4]](https://www.researchgate.net/publication/330391341_Creation_of_prototype_3D_models_using_RAPID_PROTOTYPING) Harušinec, Jozef & Suchánek, Andrej & Loulová, Mária. (2019). Creation of prototype 3D models using RAPID PROTOTYPING. MATEC Web of Conferences. 254. 01013. 10.1051/matecconf/201925401013.

[[5]](https://www.researchgate.net/publication/331800424_Image-based_Contouring_and_G-code_Generation_for_Additive_Manufacturing) Ueng, Shyh-Kuang & Huang, Hsuan-Kai & Liu, Zen-Yu. (2019). Image-based Contouring and G-code Generation for Additive Manufacturing.

[[6]](https://www.researchgate.net/publication/335847757_A_G-Code_Generator_for_Volumetric_Models) Ueng, Shyh-Kuang & Huang, Hsuan-Kai & Huang, Hsin-Cheng. (2019). A G-Code Generator for Volumetric Models. Applied Sciences. 9. 3868. 10.3390/app9183868.

[[7]](https://ieeexplore.ieee.org/document/8370990) A. Khleif and A. A. Shnawa, "Vision system aided 3D object reconstruction and machining using CNC milling machine," 2018 International Conference on Advance of Sustainable Engineering and its Application (ICASEA), Wasit - Kut, Iraq, 2018, pp. 249-254, doi: 10.1109/ICASEA.2018.8370990.

[[8]](https://ieeexplore.ieee.org/document/6376677) P. Ji, L. Wang, D. -X. Li and M. Zhang, "An automatic 2D to 3D conversion algorithm using multi-depth cues," 2012 International Conference on Audio, Language and Image Processing, Shanghai, China, 2012, pp. 546-550, doi: 10.1109/ICALIP.2012.6376677.

[[9]](https://ieeexplore.ieee.org/document/7087681) Manbae Kim, "2D-to-3D conversion using color and edge," 2014 International SoC Design Conference (ISOCC), Jeju, Korea (South), 2014, pp. 171-172, doi: 10.1109/ISOCC.2014.7087681.

[[10]](https://ieeexplore.ieee.org/document/5156848) K. Yamada and Y. Suzuki, "Real-time 2D-to-3D conversion at full HD 1080P resolution," 2009 IEEE 13th International Symposium on Consumer Electronics, Kyoto, Japan, 2009, pp. 103-106, doi: 10.1109/ISCE.2009.5156848.