

**MINISTRY OF EDUCATION AND TRAINING**

**HCMC UNIVERSITY OF TECHNOLOGY AND EDUCATION**

**FACULTY OF INTERNATIONAL EDUCATION**

**\*\*\*\*\*\*🙠🕮🙢\*\*\*\*\*\***

****

**FINAL PROJECT REPORT**

**Subject: MACHINE VISION**

**Topic: IMAGE PROCESSING APPLICATION FOR INTERIOR DECOR STORES**

**LECTURER: Dr. Nguyễn Văn Thái**

**STUDENT:**

**Phạm Anh Kiệt - 20146220**

**Nguyễn Kim Long - 20146222**

**Trương Lê Duy Nhân - 20146263**

***HO CHI MINH CITY, 13th JANUARY, 2024***

**TABLE OF CONTENTS**

[**INTRODUCTION**](#_bookmark0) **3**

* 1. **CHAPTER 1: OVERVIEW** **4**

**1.1. Questions of the topic 4**

**1.2. Topic objectives 5**

**1.3. The applications of the topic 6**

* 1. **CHAPTER 2: THEORETICAL BASIS****7**

**2.1.** [**Python language**](#_bookmark6) **7**

**2.2.** [**Libraries in python**](#_bookmark6) **7**

**2.2.1.** [**OpenCV library**](#_bookmark6) **7**

**2.2.2.** [**Torchvision library**](#_bookmark6) **9**

**2.2.3.** [**Tkinter (GUI) library**](#_bookmark6) **10**

**2.3.** [**Control methods**](#_bookmark6) **12**

* 1. **CHAPTER 3:** [**DESIGN THE INTERFACE AND FEATURES**](#_bookmark1) **13**

**3.1. Import libraries 13**

**3.2. Design the interface 14**

**3.3. Design the features of the project 17**

* 1. [**CHAPTER 4: RESULTS, COMMENTS AND DIRECTIONS**](#_bookmark1) **19**

**4.1. Results of the project 19**

**4.2. Conclusion 22**

**4.3. Development directions 22**

[**REFERENCES**](#_bookmark0) **23**

**INTRODUCTION**

Machine vision, also known as computer vision, is a part of artificial intelligence and computer science. It helps computers understand visual information like humans do. In simple terms, it involves creating algorithms and systems that can get useful information from images or videos. The main goal of machine vision is to make machines capable of doing things like recognizing images, finding objects, and understanding scenes, similar to how humans do. This field combines knowledge from computer science, math, physics, and engineering to create systems that can pull out important information from visual data. Machine vision has different parts, including capturing images, processing them, picking out important features, recognizing patterns, and making decisions. Capturing images is about using devices like cameras to get visual data, and the quality of this data affects how well the applications work. Processing images means improving the raw visual data using techniques like filtering and segmentation. Feature extraction is about finding and separating important parts, like edges or textures, to help with further analysis. Pattern recognition uses machine learning to identify patterns or objects based on these features. Decision-making is the final step, where actions or decisions are made based on the analyzed visual data.

1. **CHAPTER 1: OVERVIEW**

**1.1. Questions of the topic**

In the world of decor stores, using image processing applications can tackle practical challenges. A key issue is the efficient search and categorization of products in the store, which image processing automates based on color, style, or other features, improving the shopping experience. Another important aspect involves online try-on applications (AR) that let customers visualize how a product would look in their space. Image processing seamlessly integrates product images into real environments, helping customers make informed decisions. Ensuring optimal image quality for products on websites or mobile apps is crucial for grabbing customer attention. Image processing tools optimize color, brightness, and filtering of product images, creating an appealing visual presentation. In-store challenges like object detection and interaction can be addressed by image processing. Recognizing objects in the store and enabling interactive experiences, such as providing detailed information or special offers when a customer shows interest, significantly enhances engagement.

Assessing how shelves and products are performing poses operational challenges. Image processing proves useful in gauging inventory levels, tracking storage, and optimizing product arrangement for an improved store layout. Determining product sizes and providing interactive details, such as dimensions and material information, can be accomplished through image processing. This assists customers in making informed choices and enhances their comprehension of the products. It is crucial to monitor customer behavior within the store for a personalized shopping experience. Image processing aids in tracking customer movements, including the duration spent at a product display, enabling the store to gain better insights into customer preferences.

**1.2. Topic objectives**

The goals of the Image Processing Application for interior decor stores cover various aspects aimed at improving both customer experience and operational efficiency. One primary objective is to enhance the shopping experience by refining the search and categorization of products based on color, style, and other features, making it more user-friendly. Another significant aim is to introduce online try-on applications (AR) that enable customers to visualize products in their own spaces. This involves smoothly integrating product images into real environments, assisting customers in making well-informed decisions. Ensuring optimal image quality for products featured on websites or mobile apps is essential for grabbing customer attention. Image processing tools prove useful in optimizing color, brightness, and filtering of product images, resulting in an attractive visual presentation. In-store challenges, such as evaluating shelf and product performance, can be tackled through the application of image processing. This includes assessing inventory levels, tracking storage, and ensuring efficient product arrangement to improve the overall store layout.

Using image processing, it's possible to determine product sizes and offer interactive details like dimensions and material information. This feature assists customers in making well-informed decisions, enhancing their understanding of the products. The critical aspect of monitoring customer behavior in the store is vital for delivering a personalized shopping experience. Image processing aids in tracking customer movements, including the time spent at a product display, enabling the store to gain better insights into customer preferences.

Overall, the objectives revolve around harnessing the capabilities of image processing to enhance the overall shopping experience, streamline operational procedures, and provide customers with a more personalized and informed approach in interior decor stores.

**1.3. The applications of the topic**

The application of Image Processing in interior decor stores brings practical benefits to both customers and store management. It involves enhancing the search and categorization of products based on color, style, and features to simplify the shopping experience. Customers can virtually try out products in their own spaces through online try-on applications, gaining a better understanding of the products. Image processing plays a role in maintaining optimal image quality for products on websites or mobile apps, capturing customer attention. It also assists in evaluating shelf performance, monitoring inventory levels, and ensuring efficient product arrangement to maintain an organized store. Understanding customer preferences is crucial, and image processing systems that track how much time customers spend at specific product displays contribute to this insight.

Additionally, the analysis of image data can predict interior design and décor trends, allowing stores to stay current. Providing interactive information, such as product dimensions and material details, enhances customer understanding. Image processing also contributes to optimizing the online shopping experience by improving user interfaces and streamlining the product selection process.

In summary, these practical applications lead to an enhanced shopping experience, streamlined store management, and an overall improved interaction between customers and interior decor stores.

1. **CHAPTER 2: THEORETICAL BASIS**

**2.1. Python language**

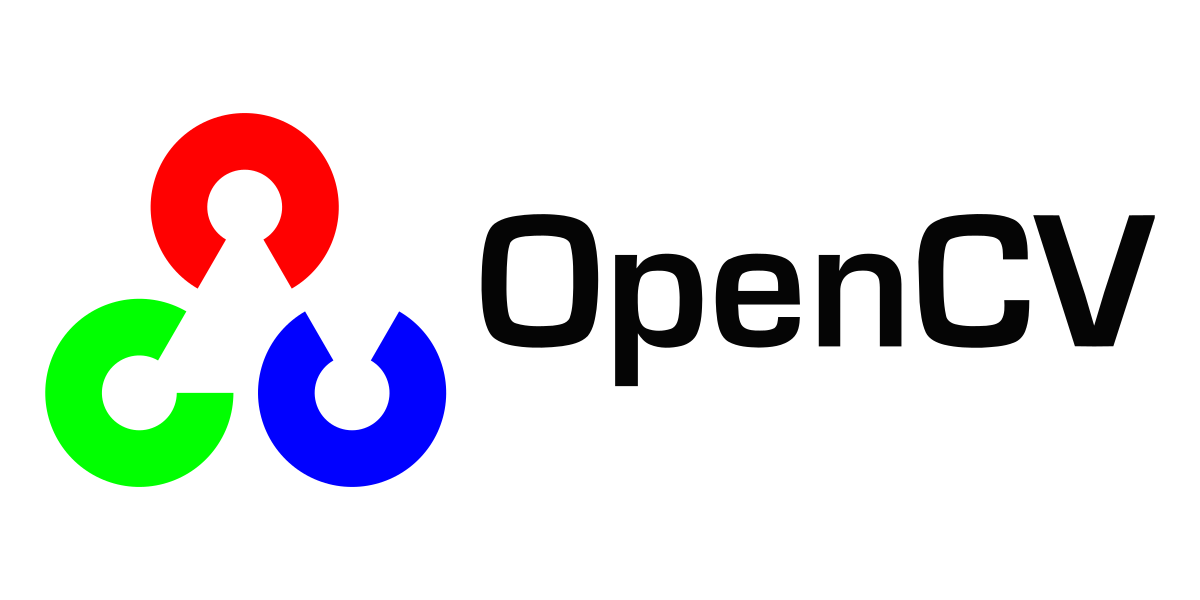
Python is a high-level programming language created by Guido van Rossum, first introduced in 1991. It is known for its versatility and is designed to be easy to read, learn, and remember. Python boasts a clean and clear syntax, making it convenient for beginners to grasp programming concepts with minimal keystrokes. Initially developed to run on Unix, Python has expanded its reach over time to various operating systems, including MS-DOS, Mac OS, OS/2, Windows, Linux, and other Unix-based systems. Despite the collaborative contributions to Python's development, Guido van Rossum remains the primary author and plays a key role in determining its direction. Python's structure allows users to write code with minimal typing, emphasizing simplicity and readability. Guido van Rossum continues to be the central figure in steering the development path of Python, holding a pivotal role in decision-making for the language's evolution.

**2.2. Libraries in python**

**2.2.1. OpenCV library**

**Definition of OpenCV:**

* OpenCV, or Open Source Computer Vision Library, is an open-source computer vision and machine learning software library. It provides a wide range of tools and algorithms for image and video analysis, including image processing, object detection, feature extraction, and machine learning.
* Originally developed by Intel, OpenCV is written in C and C++ and has interfaces for various programming languages, including Python, Java, and MATLAB. It is widely used in both academic and industrial settings for applications such as robotics, augmented reality, facial recognition, gesture recognition, and medical image analysis.
* OpenCV's modular structure makes it easy to use and integrate into different projects. The library includes over 2500 optimized algorithms, making it a valuable resource for developers and researchers working on computer vision and image processing tasks. OpenCV is released under the open-source BSD license, allowing users to freely use and modify the source code. The collaborative nature of the OpenCV community contributes to its continuous development and improvement.
* OpenCV has been utilized in a wide array of applications, from simple image processing tasks to complex computer vision systems. It has found applications in industries such as healthcare, automotive, robotics, and security. For example, OpenCV is often used for medical image analysis, lane detection in autonomous vehicles, and surveillance systems.



***Figure 2.1. OpenCV logo***

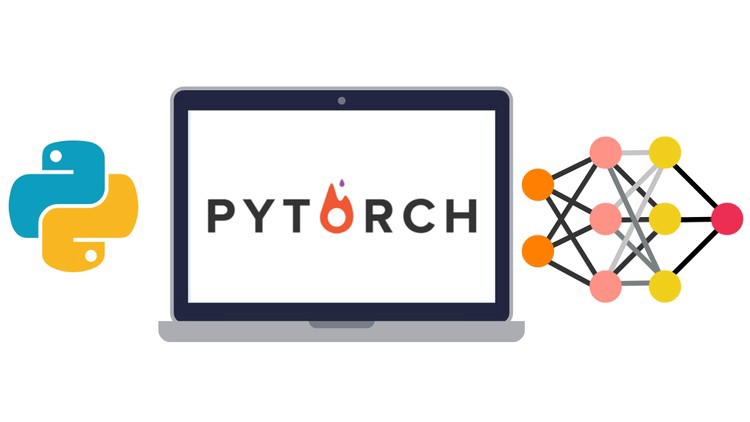
**Features of OpenCV:**

* Reading and saving videos.
* Image processing can involve noise filtering or image transformations.
* Identifying features of image shapes.
* Detecting predefined objects such as faces, eyes, etc., in videos or images.
* Analyzing videos, estimating the motion of objects, separating background and objects, etc.

**2.2.2. Torchvision library**

**Definition of Torchvision:**

* The torchvision library is an essential component of the PyTorch ecosystem, specifically designed for computer vision tasks. It serves as a comprehensive toolkit, providing convenient functions, datasets, and pre-trained models to streamline the development of computer vision applications in Python.
* One of the prominent features of torchvision is its inclusion of widely-used datasets like CIFAR-10, CIFAR-100, MNIST, and ImageNet. These datasets are pivotal for tasks such as image classification, object detection, and segmentation, making them readily accessible for research and benchmarking purposes.
* For image preprocessing, torchvision offers a range of transformation functions. These include resizing, cropping, normalization, and data augmentation, allowing users to efficiently prepare and augment their image datasets before training neural networks.

****

***Figure 2.2. Torchvision logo***

**Features of Torchvision:**

* The library simplifies data preprocessing with transformation functions, including resizing and augmentation.
* Torchvision includes pre-trained models for various tasks, aiding in image classification and object detection.
* Utilities for image loading and display, along with support for object detection and segmentation, make it a versatile tool.
* Torchvision seamlessly integrates with PyTorch, enhancing the development of computer vision solutions.

**2.2.3. Tkinter (GUI) library**

**Definition of Tkinter (GUI) library:**

* Tkinter is a standard Python library used for creating Graphical User Interfaces (GUIs) in desktop applications. It acts as a binding for the Tk GUI toolkit, which originated from the Tcl language. Tkinter enables developers to build interactive applications with buttons, labels, entry fields, and other graphical elements.
* Key features of Tkinter include built-in widgets such as buttons, labels, and entry fields, providing the necessary tools for designing the user interface. The library follows an event-driven programming paradigm, where user actions trigger events, and developers define functions or methods to respond to these events.
* Tkinter incorporates geometry managers like pack, grid, and place, facilitating the organization and arrangement of widgets within the application window. It is cross-platform, ensuring that applications developed using Tkinter can run seamlessly on Windows, macOS, and Linux.



***Figure 2.3. Tkinter (GUI) logo***

**Features of Tkinter (GUI):**

* Tkinter provides a collection of pre-built widgets like buttons, labels, checkboxes, and entry fields, simplifying the process of creating interactive interfaces.
* Tkinter adopts an event-driven approach, where user actions such as button clicks trigger events. Developers can easily define functions to respond to these events, adding interactivity to the interface.
* The library includes geometry managers such as pack, grid, and place, enabling developers to organize and position widgets within the application window effortlessly.
* Tkinter is designed to be cross-platform, ensuring consistent behavior across operating systems like Windows, macOS, and Linux.
* Tkinter permits developers to customize the appearance of widgets, including colors, fonts, and styles, allowing for visually appealing and personalized interfaces.
* As part of the Python Standard Library, Tkinter seamlessly integrates with Python, allowing developers to create GUI applications using Python code without external dependencies.

**2.3. Control methods**

* Mark and detect objects that need to be removed.
* Calculate the pixels of the object.
* Set up an interface to remove objects that the customer does not want and an interface to add objects that the customer is interested in.
* Then add the items that the customer or we feel most satisfied with.
* To exit, we can click on the X on the right corner of the interface.

1. **CHAPTER 3: DESIGN THE INTERFACE AND FEATURES**

**3.1. Import libraries**

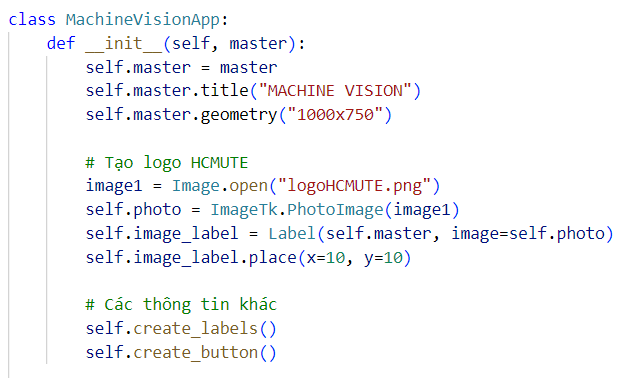
* Import the Tkinter library, which is a standard GUI (Graphical User Interface) toolkit for Python.
* Import the Python Imaging Library (PIL), now known as Pillow, which is used for handling images.
* Import the NumPy library, which is a powerful library for numerical operations in Python.
* Import the PyTorch library, a popular deep learning framework. It is used for building and training neural networks.
* Import the Mask R-CNN model from the torchvision library. Mask R-CNN is a deep learning model used for object detection and instance segmentation.
* Import the “read\_image” function from torchvision.io, which is used to read an image from a file into a PyTorch tensor.
* Imports the Matplotlib library, a popular plotting library in Python.
* Import the os module, which provides a way of interacting with the operating system.
* Import the OpenCV library, which is used for computer vision tasks, including image and video processing.
* Import specific widgets (Listbox, Button) and a messagebox from the Tkinter library.

****

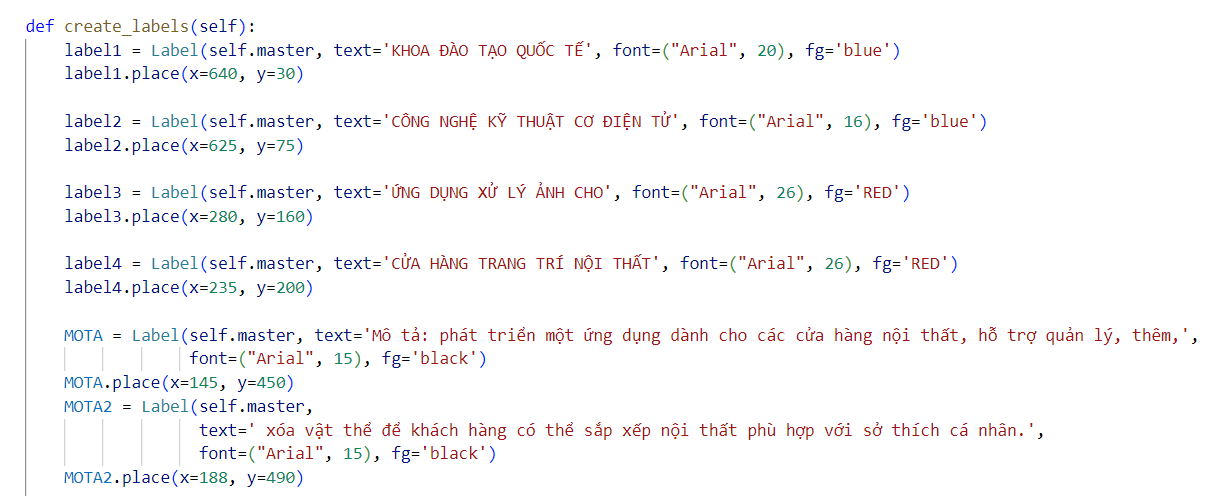
***Figure 3.1. Import the libraries***

**3.2. Design the interface**

* Defines the ”\_\_init\_\_” method for a class, presumably a Tkinter GUI application. Let's go through each line to understand its purpose.
* defines the constructor method “\_\_init\_\_” for the class. It takes a parameter master, which is expected to be a Tkinter master (main) window.
* Initializes an instance variable “self.master” with the provided master window. This allows the class to have access to the main Tkinter window. This line sets the title of the main window to "MACHINE VISION". Set the initial size of the main window to a width of 1000 pixels and a height of 750 pixels.
* Load an image from the file "logoHCMUTE.png" using the PIL (Pillow) library. It then converts the image to a Tkinter PhotoImage object “(self.photo)”. Finally, a Tkinter Label widget “(self.image\_label)” is created and placed at coordinates “(x=10, y=10)” within the main window to display the logo.
* Creates a Label widget (label1) with the text 'KHOA ĐÀO TẠO QUỐC TẾ', a font size of 20, and a blue foreground color. It is placed at coordinates (640, 30) within the main window.
* Creates another Label (label2) with a different text 'CÔNG NGHỆ KỸ THUẬT CƠ ĐIỆN TỬ', font size, and position.
* Label (label3) is created with a different text 'ỨNG DỤNG XỬ LÝ ẢNH CHO' and position. The font size is larger, and the text color is red.
* Create another Label (label4) with different text 'CỬA HÀNG TRANG TRÍ NỘI THẤT' and position.
* Label (MOTA) is created to display a description text with a smaller font size and different position.
* Label (label5) is created for the name of the supervisor 'GVHD: TS. Nguyễn Văn Thái' with a larger font size and different position.
* Another Label (label6) is created for the label 'SVTH:' (Sinh viên thực hiện) with a different position.
* Labels (label7, label8, label9) are created for the names and the IDs of the students who implemented the project, each with different positions.



***Figure 3.2. First part of the interface’s code***



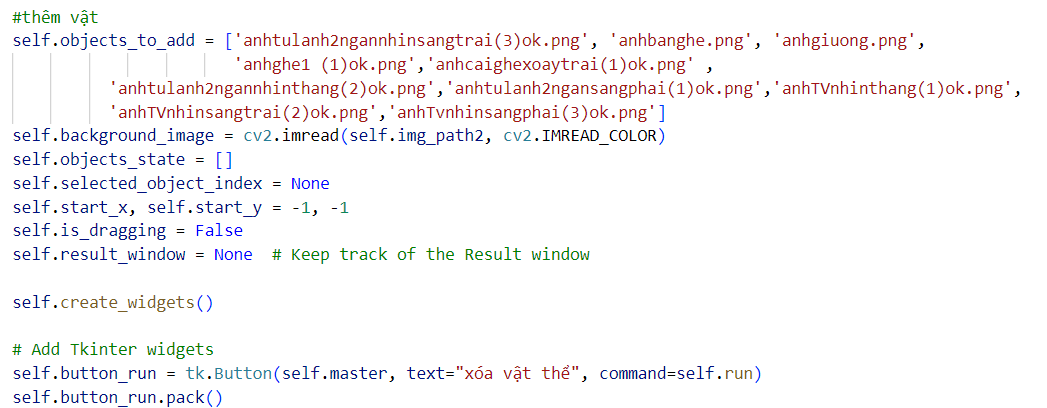
*Figure 3.3. Second part of the interface’s code*



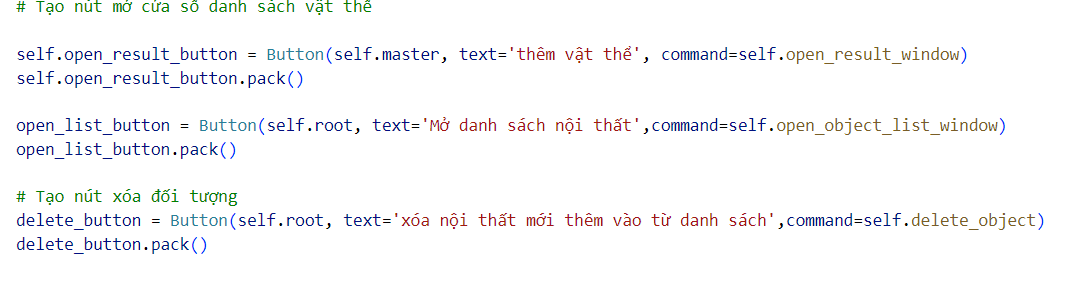
*Figure 3.3. Final part of the interface’s code*

**3.3. Design the features of the project**

* Initializes a list “objects\_to\_add” containing filenames of images representing objects that can be added to the scene.
* Reads an image from the file specified by “self.img\_path2” using OpenCV. It appears to be the background image of the scene.
* Initialize various variables for managing the state of the objects, the selected object index, starting coordinates for dragging, and a variable for keeping track of the Result window.
* Call a method “create\_widgets”, which is assumed to create some Tkinter widgets for the GUI. The exact implementation of this method is not provided.
* Create a button “(button\_run)” with the text "xóa vật thể" (delete object) and associates it with the run method when clicked.
* Create a button “(open\_result\_button)” with the text "thêm vật thể" (add object) and associates it with the open\_result\_window method when clicked.
* Create a button “(open\_list\_button)” with the text "Mở danh sách nội thất" (Open furniture list) and associates it with the open\_object\_list\_window method when clicked.
* Creates a button “(delete\_button)” with the text "xóa nội thất mới thêm vào từ danh sách" (delete newly added furniture from the list) and associates it with the “delete\_object” method when clicked.



*Figure 3.4. First part of the features interface code*



*Figure 3.5. Final part of the features interface code*

1. **CHAPTER 4: RESULTS, COMMENTS AND DIRECTIONS**

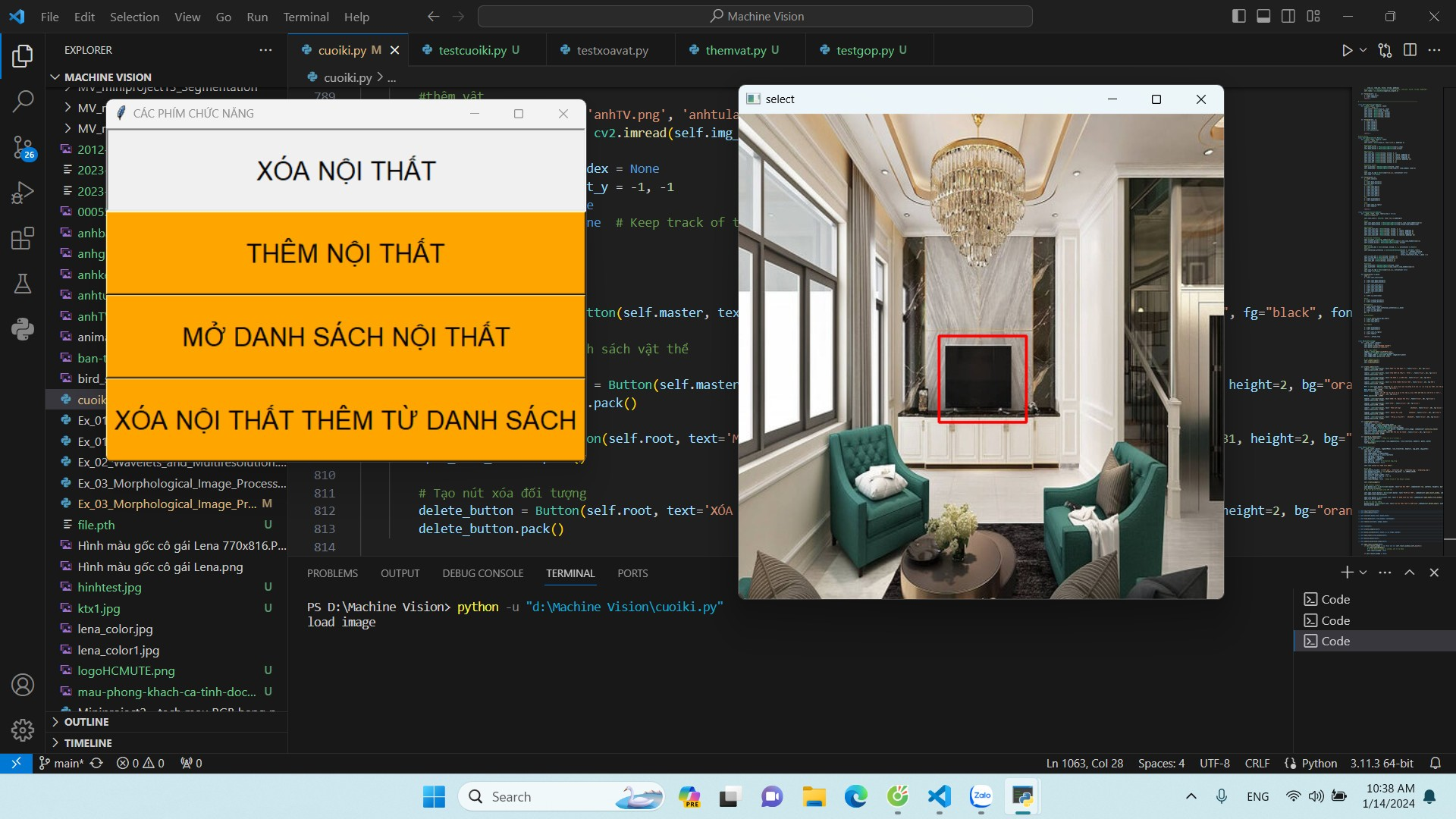
**4.1. Results of the project**

****

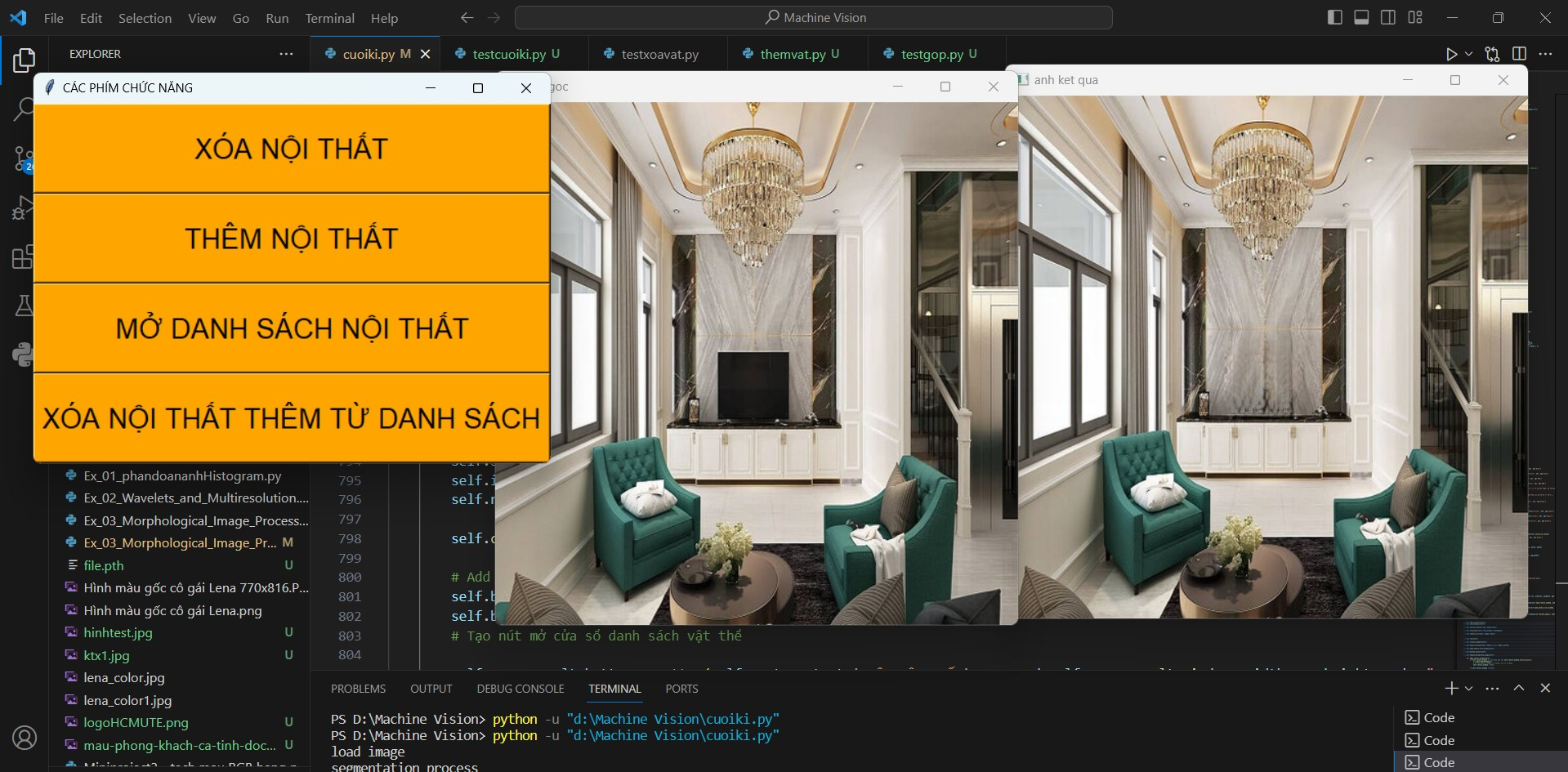
***Figure 4.1. The completed interface of the project***



***Figure 4.2. The completed features interface of the project***

******

***Figure 4.3. When you delete the TV in the original image***

******

***Figure 4.4. The TV disappear in the resulted image***

******

***Figure 4.5. Add another TV to the resulted image***

**4.2. Conclusion**

The "Image Processing Application for Interior Decoration Store" is a user-friendly tool designed for interior decoration purposes. Using a Tkinter-based graphical interface and OpenCV for image handling, the application allows users to add and remove furniture objects from a background image. Key features include the ability to select, drag, and drop predefined furniture items onto the scene. The application also facilitates object deletion and provides a window for browsing a list of available furniture objects. Team information, including academic department details and project contributors, is displayed in the GUI.

Overall, this application is practical for decorators and customers to visualize and plan furniture arrangements in a given space.

**4.3. Development directions**

* Make the app simpler with clear buttons and helpful tips.
* Add pics or tips to show what each button does.
* Allow users to change furniture size, color, or look.
* Let them pick colors or patterns for the furniture.
* Make a version that works well on phones.
* Adjust how things look for smaller screens and touch.
* Make the app work faster, especially with big pics or lots of furniture.
* Load things only when needed for a quicker app.

**REFERENCES**

[1] “Rafael C. Gonzalez, Richard E. Woods - Digital Image Processing (2008, Prentice Hall).

[2] “Badrulhisham, Nur Alia Syahirah, and Nur Nabilah Abu Mangshor. "Emotion Recognition Using Convolutional Neural Network (CNN)". Journal Of Physics: Conference Series, vol 1962, no. 1, 2021, p. 012040. IOP Publishing, doi:10.1088/1742-6596/1962/1/012040. Accessed 28 Jan 2022.

[3] “When to use Dense, and all the other layers? and Kiritee Gak. "When To Use Dense, Conv1/2D, Dropout, Flatten, And All The Other Layers?". Data Science Stack Exchange, 2019.

[4] “Segmentation”. From: The scikit-image contributors, 2020.  
Link: <https://scikit-image.org/skimage-tutorials/lectures/4_segmentation.html>.

[5] “Application of Computer Image Processing Software in Interior Design”. From: Qing Li, 10 December 2021.  
Link: <https://link.springer.com/chapter/10.1007/978-981-16-7466-2_90>.

[6] “Restaurant Interior Design under Digital Image Processing Based on Visual Sensing Technology”. From: Yan Wan,corresponding author Can Cui, and Guanqiang Wang, 8 Jun 2022.  
Link: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9200518/>.