**Jewellery Detection System using SVM and YoloV8**

## UIT2602 – Web Programming Project

### A PROJECT REPORT

***Submitted by***

**AASISH 3122215002002**

**BADRI NARAYANAN 3122215002018**

**HARISH K 3122215002034 MANOJ KUMAR S 3122215002302**

**SSN COLLEGE OF ENGINEERING, KALAVAKKAM**

**MAY 2024**

### Sri Sivasubramaniya Nadar College of Engineering (An Autonomous Institution, Affiliated to Anna University)

**BONAFIDE CERTIFICATE**

Certified that this project titled “**Jewellery Detection System using SVM and YoloV8**” is the bonafide work of “Aasish M - 3122215002002**,** Badri Narayanan- 3122215002018, Harish K 3122215002034, Manojkumar S - 3122215002302”, and is

submitted for project viva-voce examination held on 23-05-2024

#### Signature of examiner

**Submitted on** 23-05-2024

#### Internal Examiner External Examiner

**Abstract**

This project introduces an innovative approach to object recognition by integrating the YOLOv8 and Support Vector Machine (SVM) algorithms. Object recognition is vital across diverse domains such as security, inventory management, and quality control. Traditional methods often struggle to balance real-time detection and robust feature extraction. To tackle this challenge, our project combines YOLOv8 and SVM to develop a comprehensive object recognition solution.

YOLOv8 excels in real-time object detection, efficiently localizing objects within images. Its simultaneous detection of multiple objects makes it ideal for applications requiring rapid and accurate detection. Conversely, SVM is employed for robust feature extraction, enhancing object classification accuracy based on these features. The integration of YOLOv8 and SVM yields a hybrid model architecture that leverages the strengths of both algorithms. YOLOv8 focuses on real-time object detection, while SVM enhances classification through robust feature extraction. This hybrid model ensures efficient and accurate object recognition, suitable for various real-world applications.

The primary objective of our project is to accurately identify specific objects, including gold bars, mangal sutra, and gold coins, in images. This identification significantly contributes to security monitoring, inventory management, and quality control processes. The successful implementation of the hybrid model demonstrates its potential in addressing the challenges associated with object recognition, offering a reliable solution for diverse applications.

In summary, our project aims to harness the combined capabilities of YOLOv8 and SVM to develop a robust object recognition system. By integrating real-time detection and robust feature extraction, our solution addresses the limitations of traditional methods, offering enhanced accuracy and efficiency in object recognition tasks. Through this innovative approach, we strive to provide a valuable tool for various industries, empowering them with reliable object recognition capabilities.

# Introduction

#### Motivation:

In today's dynamic technological landscape, there's a growing demand for efficient object recognition systems across various industries. These systems play a crucial role in security monitoring, inventory management, and quality control processes. However, existing methods often struggle to balance the need for real-time detection with accurate classification, leading to inefficiencies and inaccuracies.

#### Problem Statement:

The challenge we address in this project is to develop an object recognition system capable of achieving both real-time detection and accurate classification of objects. Traditional techniques face limitations in this regard, hindering their effectiveness in practical applications. By integrating advanced algorithms like YOLOv8 and SVM, we aim to overcome these challenges and create a hybrid solution that combines speed and accuracy seamlessly.

#### Objectives:

Our primary objective is to develop a hybrid object recognition system that integrates YOLOv8 and SVM algorithms. This system will be capable of real-time detection and accurate classification of objects, including gold bars, mangal sutra, and gold coins. Additionally, we aim to enhance the efficiency and accuracy of object recognition processes across various industries. Our goal is to provide a scalable and adaptable solution that can be easily integrated into existing systems.

#### Deliverables:

* Fully developed hybrid object recognition system integrating YOLOv8 and SVM algorithms.
* Evaluation reports demonstrating the performance and accuracy of the system.
* Technical support and assistance for deploying the system in real-world applications.
* Continuous updates and improvements to the system based on feedback and requirements.

**Requirements Engineering**

**CLIENT DETAILS:**

**Name :** Dr. Srinivasan

**Profession :** Professor

**Email ID :** [srinivasanr@ssn.edu.in](mailto:srinivasanr@ssn.edu.in)

### List of all functional modules:

* + extract\_hog\_features
  + model\_train
  + classify\_image
  + classify\_images\_in\_folder
  + yolo\_model\_loading
  + predict\_yolo
  + index
  + about
  + upload\_file

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sprint #** | **Epic** | **Essential or Desirable** | **Description of the Requirement** | **Remarks** |
| 1 | load\_dataset | essential | Loads image | Permission denied |
|  |  |  | dataset from | for some files. |
|  |  |  | specified |  |
|  |  |  | directory for |  |
|  |  |  | training the |  |
|  |  |  | model. |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | extract\_hog\_features | essential | Extracts HOG (Histogram of Oriented Gradients) features from an image. | In extremely rare cases, if the image contains very low contrast or uniform texture, HOG feature extraction might not capture significant information. |
| 1 | model\_train | desirable | Initializes and trains an SVM classifier using extracted HOG features. | In rare instances of highly imbalanced classes or noisy data, consider implementing class weighting or data augmentation techniques to improve classifier performance. |
| 2 | classify\_image | Essential | Classifies a single image using the trained SVM classifier. | Enhances genetic diversity but may not always be necessary |
| 2 | classify\_images\_in\_f older | Essential | Classifies multiple images in a specified folder using the trained SVM classifier. | In rare cases of folder structure inconsistency or unexpected file formats within the folder. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 2 | yolo\_model\_loading | Essential | Loads the YOLO model from the specified file path. | Some model file will be corrupted, ensure proper error handling and validation of the model file before loading. |
| 3 | predict\_yolo | Essential | Runs the YOLO model to predict objects in images from the specified directory. | For complex images or a large number of images in a folder, YOLO may take a longer time for prediction. |
| 4 | time | Desirable | Measures and prints the execution time of the model prediction. | None |
| 4 | allowed\_file | Essential | Checks if the uploaded file has an allowed extension. | Incorrect file extensions, ensure users are notified properly. |
| 4 | index | Essential | Renders the main page of the web application. | None |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 4 | about | Desirable | Renders the "About" page of the web  application. | It has information how the models are working and classes used in the models. |
| 4 | upload\_file | Essential | Handles file upload from the user. | Notifies the users for the network issue during the file upload if occurred. |

**Implementation and Risk Management**

#### Individual contribution: Name: Aasish M

**Register Number:3122215002002 Role in the project: Developer**

#### Implementation

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sprint #** | **Epic** | **User Story** | **Requirement / User Story** | **Remarks on implementation** |
| 1 | Data Loading and Basic Model | Develop a  mechanism to introduce random | Implement a mechanism for introducing random | Initial data loading, SVM training, and basic mutation |
|  | Training | mutations in | mutations in | mechanism designed. |
|  |  | existing schedules. | existing schedules. |  |
| 2 | Model Optimization and Validation | Evaluate the impact of random mutations on model performance and fine-tune mutation mechanism. | Assess the effect of random mutations on model performance and optimize mutation mechanism. | Model evaluation, mutation mechanism adjustment, and optimization performed. |
| 3 | Model Deployment and  Integration | Integrate the optimized mutation mechanism into the  final model and | Integrate the optimized mutation mechanism into the final model and perform testing. | Integration into the final model, testing, and bug fixing completed. |
|  |  | conduct thorough |  |
|  |  | testing. |  |

1. **Risk Management**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Risk #** | **Risk Description** | **Probabil ity** | **Impact** | **Mitigation Plan** |
| 1 | Limited labeled data for training. | Medium | Reduced model accuracy and generalization. | Augment training data through data augmentation techniques. |
| 2 | Model overfitting during training. | High | Reduced model performance on unseen data. | Regularization techniques implementation and validation. |
| 3 | Software library compatibility issues. | Low | Delayed development and deployment timelines. | Regularly update and test software dependencies for compatibility. |

#### Name: Manojkumar S Reg no: 3122215002302

**Role in the project: Developer**

#### Implementation

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sprint#** | **Epic** | **User story** | **Requirement** | **Remarks** |
| 1 | Data | Develop a | Implement a | Initial data loading, |
|  | Preprocessing and | mechanism to | mechanism for | SVM training, and |
|  | Feature Extraction | introduce random | introducing | basic mutation |
|  |  | mutations in | random mutations | mechanism |
|  |  | existing schedules. | in existing | designed. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  | schedules. |  |
| 2 | Model Evaluation | Evaluate the impact | Assess the effect | Model evaluation, |
|  | and Fine-tuning | of random | of random | mutation |
|  |  | mutations on model | mutations on | mechanism |
|  |  | performance and | model | adjustment, and |
|  |  | fine-tune mutation | performance and | optimization |
|  |  | mechanism. | optimize mutation | performed. |
|  |  |  | mechanism. |  |
| 3 | User Interface | Integrate the | Integrate the | Integration into the |
|  | Integration and | optimized mutation | optimized | final model, |
|  | Testing | mechanism into the | mutation | testing, and bug |
|  |  | final model and | mechanism into | fixing completed. |
|  |  | conduct thorough | the final model |  |
|  |  | testing. | and perform |  |
|  |  |  | testing. |  |

1. **Risk Management**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Risk #** | **Risk Description** | **Probability** | **Impact** | **Mitigation Plan** |
| 1 | Limited domain knowledge in image processing. | Medium | Decreased efficiency in feature extraction and model optimization. | Collaborate with domain experts or undergo training sessions. |
| 2 | Insufficient computational resources for model training. | High | Extended model training time and delays in project completion. | Utilize cloud-based computing resources or optimize model architecture. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 3 | Data privacy and security concerns. | Low | Legal and ethical issues, damage to reputation. | Implement strict data access controls and encryption measures. |

#### Name : V T Badri Narayanan Reg No:3122215002018

**Role in the project: Developer**

#### Implementation

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sprint#** | **Epic** | **User story** | **Requirement** | **Remarks** |
| 1 | Model Training | As a user I want a system that detects and classifies the object in the image as quick as possible. | A fast model to predict the image. | Keep latency as low as possible. |
| 2 | Integrating with Web | As a user I want a web application that is powered by the model | An efficient way to connect the backend and frontend. | Handle the events of image uploading and passing to backend safely. |

1. **Risk Management**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Risk #** | **Risk Description** | **Probability** | **Impact** | **Mitigation Plan** |
| 1 | Model will become overfitting. | high | Model will not detect objects. | Have a diverse set of dataset. Perform upsampling or downsampling of images of each class appropriately. |
| 2 | Server will fail collecting the image and perform detection. | low | Web application will not show the detection at all | Conduct regular tests to ensure correct collection and processing of the image. |

#### Name : Harish K

**Reg No:3122215002034**

#### Role in the project: Developer

1. **Implementation**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sprint#** | **Epic** | **User story** | **Requirement** | **Remarks** |
| 1 | Model Training | Train the model to classify the images using YOLOv8 | Classifying the images | Training the model to classify the given input image. |
| 2 | Model Fine-tuning | Remove unnecessary images and train the model with correct epochs | To minimize the misclassification of images | Minimized the misclassification of images |
| 3 | Interface to upload the image. | Develop a website to use the model to classify the images | Website for the user to input the image and get the results | A website to use the model. |

#### Risk Management

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Risk #** | **Risk Description** | **Probability** | **Impact** | **Mitigation Plan** |
| 1 | Model will become overfitting. | high | Model will not detect objects. | Remove images that can decrease the training accuracy of the model |
| 2 | Image uploading might not work | low | Web application will not show the detection at all | Conduct regular tests to ensure correct collection and processing of the image. |

1. **Test Log report**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TC id** | **Test case description/ condition** | **Test case input** | **Expected Output** | **Result (PASS/ FAIL)** |
| TC 1 | Single gold bar image. | File Name: "1.jpg" Format: JPEG Size: 85 KB Dimensions: 700X700 pixels Content: Image of a single gold bar | Bar | PASSED |

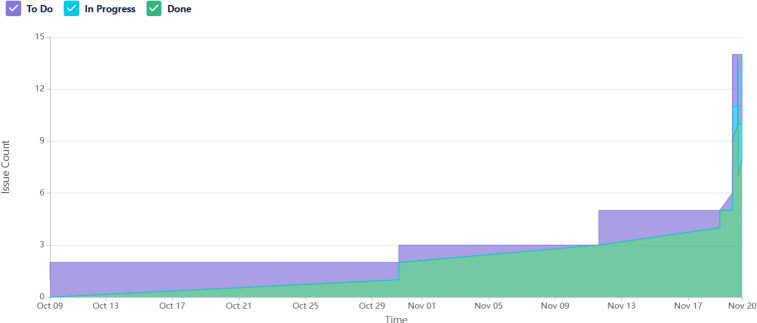
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TC 2 | Single silver bar image | File Name: "2.jpg" Format: JPEG Size: 85 KB Dimensions: 700X700 pixels Content: Image of a single silver bar | Bar | PASSED |
| TC 3 | Single gold coin image | File Name: "3.jpg" Format: JPEG Size: 85 KB Dimensions: 700X700 pixels Content: Image of a single gold coin | Coin | PASSED |
| TC 4 | Image of a single silver coin | File Name: "4.jpg" Format: JPEG Size: 85 KB Dimensions: 700X700 pixels Content: Image of a single silver coin | Coin | PASSED |
| TC 5 | Image of a single Mangal sutra with black beads | File Name: "5.jpg" Format: JPEG Size: 200 KB Dimensions: 1000X1000 pixels | Mangal sutra | PASSED |

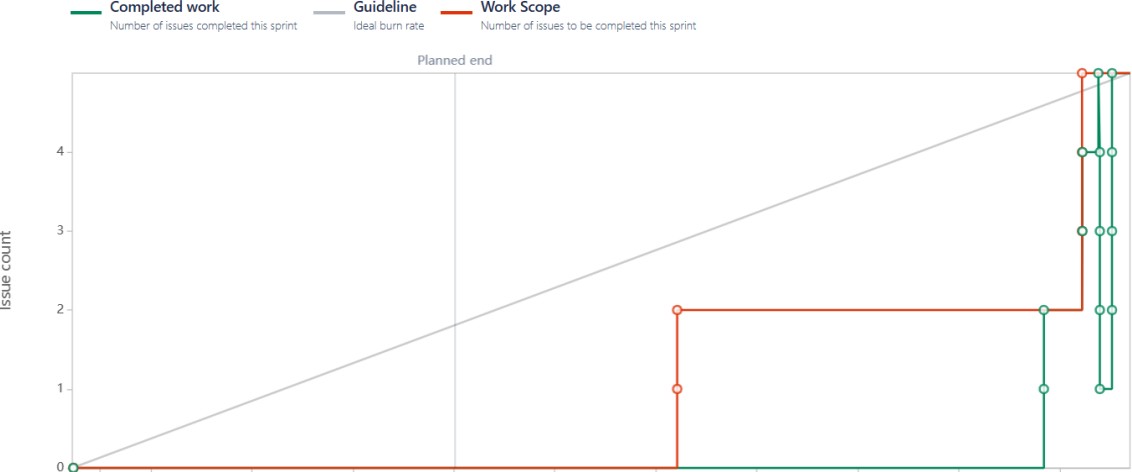
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | Content: Image of a single Mangal sutra with black beads. |  |  |
| TC 6 | Image of a single thali without black beads | File Name: "6.jpg" Format: JPEG Size: 150 KB Dimensions: 700X700 pixels Content: Image of a single thali without black beads | Thali | PASSED |
| TC 7 | Image of a single thali with one or two black beads | File Name: "7.jpg" Format: JPEG Size: 120 KB Dimensions: 800X750 pixels Content: Image of a single thali with black beads | Thali | PASSED |
| TC 8 | Image of a single thali without chain attached | File Name: "8.jpg" Format: JPEG Size: 130 KB Dimensions: 700X700 pixels Content: Image of a single thali without chain attached | Thali | PASSED |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TC 9 | Image of a single thali with coin- like object attached. | File Name: "9.jpg" Format: JPEG Size: 140 KB Dimensions: 600X600 pixels Content: Image of a single thali with coin-like object attached | Thali | PASSED |
| TC 10 | Image of a  Mangal sutra having both thali and black beads | File Name: "10.jpg"  Format: JPEG Size: 250 KB Dimensions: 800X800 pixels Content: Image of a single Mangal sutra with black beads and thali | Mangal sutra | PASSED |
| TC11 | Image of a single ring | File Name:"11.jpg" Format: JPEG Size: 140 KB Dimensions: 600X600 pixels Content: Image of a single gold ring | Others | PASSED |
| TC12 | Image of a single gold chain | File Name: "12.jpg"  Format: JPEG Size: 140 KB | Others | PASSED |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | Dimensions: 600X600 pixels Content: Image of a single gold chain |  |  |
| TC13 | Image of a single necklace | File Name: "13.jpg"  Format: JPEG Size: 140 KB Dimensions: 600X600 pixels Content: Image of a single gold necklace | Others | PASSED |
| TC14 | Image of a bangle | File Name: "14.jpg"  Format: JPEG Size: 140 KB Dimensions: 600X600 pixels Content: Image of a single gold bangle | Others | PASSED |

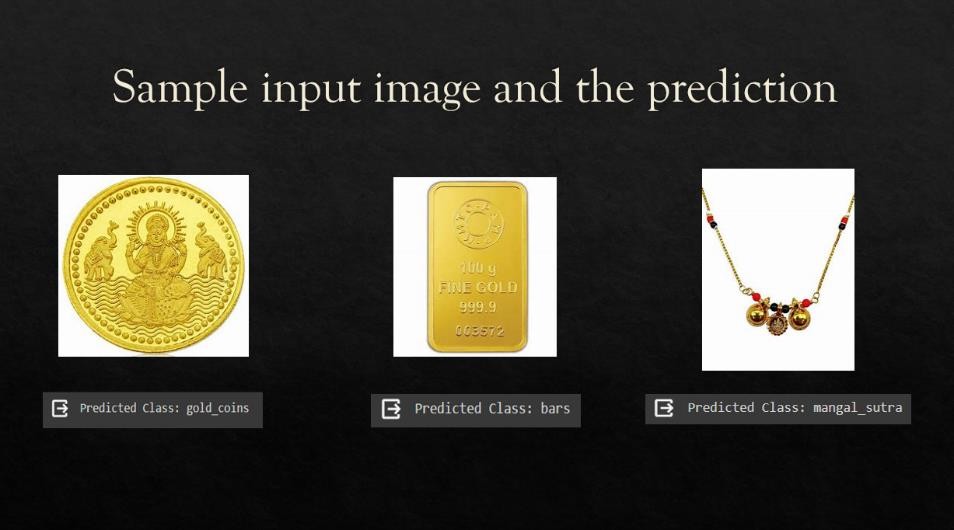
# Project Managementsss





**Project Outcomes**

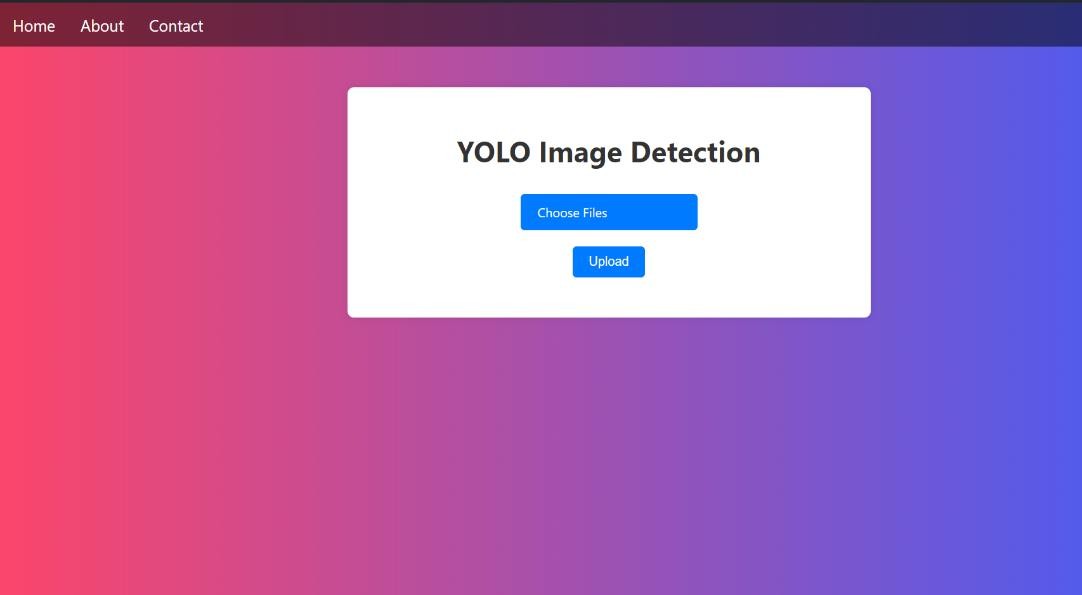
SVM prediction:



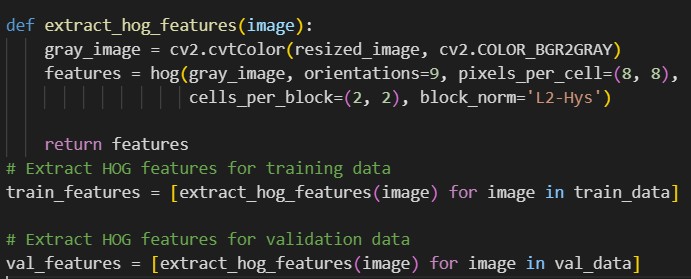
YOLO prediction:

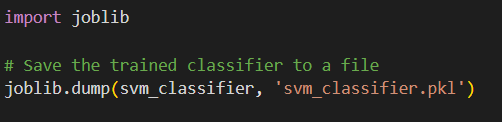


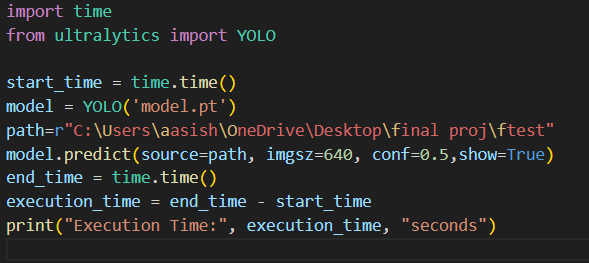
Home page



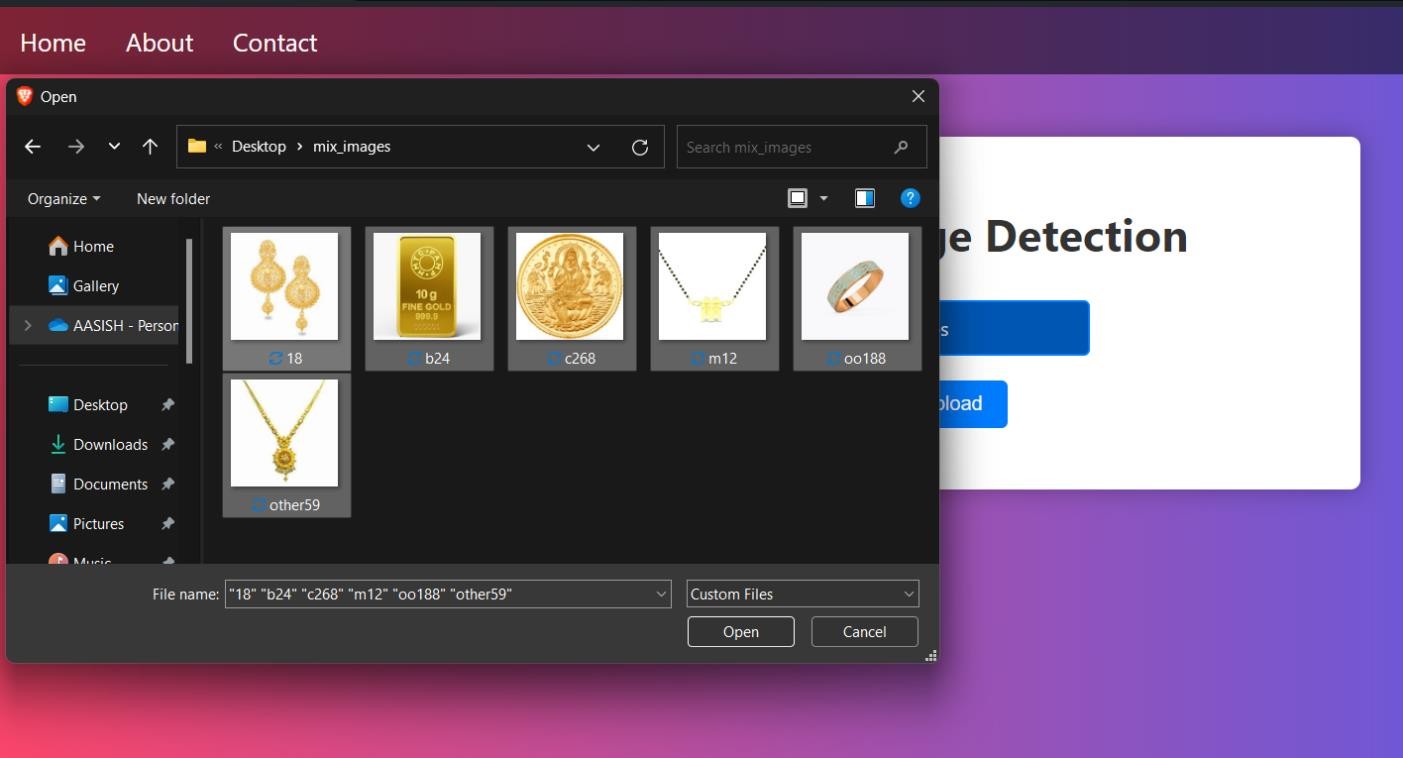
## Code snippets:



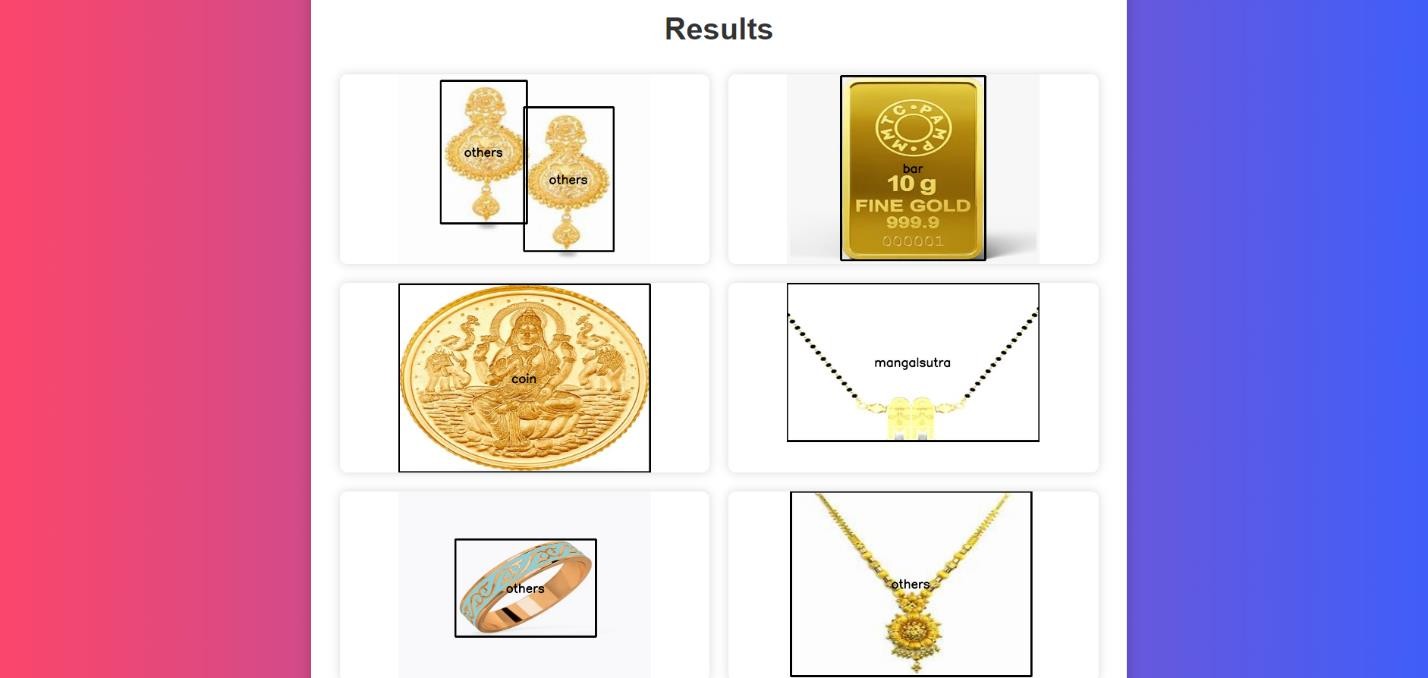




Selecting multiple images**:**



Yolo predictions:



# Conclusion

Our project successfully integrates YOLOv8 and SVM for a powerful and efficient object recognition system. By combining the real-time detection capabilities of YOLOv8 with the robust feature extraction of SVM, we have created a hybrid model that enhances both accuracy and efficiency in object recognition tasks.

The strength of YOLOv8 lies in its ability to quickly and accurately detect objects in real-time, making it ideal for dynamic environments where speed is crucial. On the other hand, SVM provides a reliable method for feature extraction and classification, ensuring that the system can accurately identify various objects, even in challenging conditions. This synergy allows for precise identification of items such as gold bars, mangal sutra, and gold coins, making it suitable for diverse applications.

This integrated approach offers significant advancements in the field of object recognition technology. The system's ability to handle a wide range of images and accurately identify specific objects meets the demands of various sectors, including security and inventory management. Its potential applications extend to monitoring and managing valuable items in secure environments, enhancing inventory accuracy, and improving overall operational efficiency.

In summary, the development of this hybrid model marks a notable step forward in object recognition technology. By leveraging the unique strengths of YOLOv8 and SVM, we have created a comprehensive solution that addresses real-world needs with high accuracy and efficiency. This project demonstrates the potential for advanced object recognition systems to make significant contributions across multiple domains.

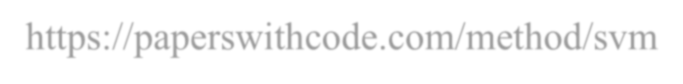
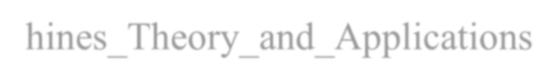
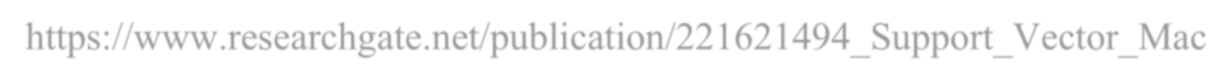
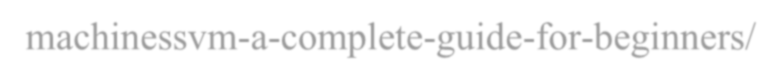
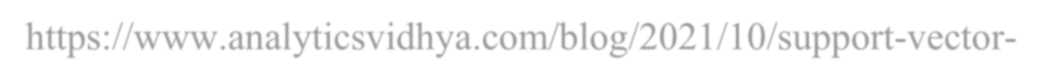
# Future Directions

In the future, enhancing our jewellery detection system could involve integrating advanced machine learning techniques such as Convolutional Neural Networks (CNNs) for improved accuracy and robustness. Exploring transfer learning, where

pre-trained models are adapted to our specific task, could also significantly boost performance with less computational effort.

Another promising direction is the development of a cloud-based solution for real- time image processing and analysis. This would allow users to upload images from anywhere and receive instant feedback. Enhancing the user interface with detailed feedback on predictions, including confidence scores and detected object annotations, would further improve the user experience. Implementing continuous learning capabilities would ensure the system stays effective and up-to-date with new data over time.

# References



* [https://www.analyticsvidhya.com/blog/2021/10/support-vector-](https://www.analyticsvidhya.com/blog/2021/10/support-vector-machinessvm-a-complete-guide-for-beginners/) [machinessvm-a-complete-guide-for-beginners/](https://www.analyticsvidhya.com/blog/2021/10/support-vector-machinessvm-a-complete-guide-for-beginners/)
* [https://www.researchgate.net/publication/221621494\_Support\_Vector\_Mac](https://www.researchgate.net/publication/221621494_Support_Vector_Machines_Theory_and_Applications) [hines\_Theory\_and\_Applications](https://www.researchgate.net/publication/221621494_Support_Vector_Machines_Theory_and_Applications)
* <https://paperswithcode.com/method/svm>