ObjecTally: Template-Based Counting

*A **MAJOR** Project Report submitted in partial fulfillment of the requirements for the award of the degree of*

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE & ENGINEERING

With Specialization **AI & ML**

BY

SALMAN FAIZI UID: TNU2021053200018L

AYUB ALAM UID:TNU2021053200031L

AYAN PRAMANIK UID:TNU2020053100001

SUBARNA DAS UID: TNU2020053100011

MD ZUNNURAIN UID:TNU2020053100014

Under the supervision of

ARIJIT HAJRA

CEO

Think Again Lab, Kolkata, West Bengal, India

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

THE NEOTIA UNIVERSITY

D.H. ROAD, SOUTH 24-PARGANAS, WEST BENGAL, INDIA, PIN-743368 MAY 2024

Certificate

We hereby recommend that the Project entitled **"ObjecTally:Template-Based Counting"** worked under our guidance may please be accepted in the partial fulfillment of the requirement for the degree of "Bachelor in Technology" in the Computer Science and Engineering with specialization AI & ML of 'The Neotia University'. The project report in our opinion is worthy for its acceptance. During the work '**Salman Faizi**' was found to be sincere, regular, and hard-working and has successfully completed the thesis work assigned to him.

Arijit Hajra.

……………………..

(Project Guide)
Founder and CEO,
Think Again Lab

……………………….
(HOD, CSE)
The Neotia University

Certificate

We hereby recommend that the Project entitled **"ObjecTally:Template-Based Counting"** worked under our guidance may please be accepted in the partial fulfillment of the requirement for the degree of "Bachelor in Technology" in the Computer Science and Engineering with specialization AI & ML of 'The Neotia University'. The project report in our opinion is worthy for its acceptance. During the work '**Ayub Alam**' was found to be sincere, regular and hard working and has successfully completed the thesis work assigned to him.

Arijit Hajra.

……………………..

(Project Guide)
Founder and CEO,
Think Again Lab

……………………….

(HOD, CSE)
The Neotia University

Certificate

We hereby recommend that the Project entitled ***"ObjecTally:Template-Based Counting"*** worked under our guidance may please be accepted in the partial fulfillment of the requirement for the degree of "Bachelor in Technology" in the Computer Science and Engineering with specialization AI & ML of 'The Neotia University'. The project report in our opinion is worthy for its acceptance. During the work '**Ayan Pramanik**' was found to be sincere, regular and hard working and has successfully completed the thesis work assigned to him.

Arijit Hajra.

……………………..

(Project Guide)
Founder and CEO,
Think Again Lab

……………………….

(HOD, CSE)
The Neotia University

Certificate

We hereby recommend that the Project entitled **"ObjecTally:Template-Based Counting"** worked under our guidance may please be accepted in the partial fulfillment of the requirement for the degree of "Bachelor in Technology" in the Computer Science and Engineering with specialization AI & ML of 'The Neotia University'. The project report in our opinion is worthy for its acceptance. During the work '**Subarna Das**' was found to be sincere, regular and hard working and has successfully completed the thesis work assigned to him.

Arijit Hajra.

……………………..

(Project Guide)
Founder and CEO,
Think Again Lab

……………………….

(HOD, CSE)
The Neotia University

Certificate

We hereby recommend that the Project entitled **"ObjecTally:Template-Based Counting"** worked under our guidance may please be accepted in the partial fulfillment of the requirement for the degree of "Bachelor in Technology" in the Computer Science and Engineering with specialization AI & ML of 'The Neotia University'. The project report in our opinion is worthy for its acceptance. During the work '**Md Zunnurain**' was found to be sincere, regular and hard working and has successfully completed the thesis work assigned to him.

Arijit Hajra.

……………………..

(Project Guide)
Founder and CEO,
Think Again Lab

……………………….

(HOD, CSE)
The Neotia University

Declaration of Originality and Compliance of Academic Ethics

We hereby declare that this report contains literature survey and original research work done by the under signed candidates, as part of our **"Bachelor in Technology Studies"**.

All information in this document have been obtained and presented in accordance with academic rules and ethical conduct.

we also declare that, as required by these rules and conduct, We have cited and referenced all materials that are original to this work.

Project Title: ObjecTally:Template-Based Counting

Student 1
Name: Salman Faizi
UID: TNU2021053200018L
Signature with Date: Salman Faizi 20/06/24

Student 2
Name: Ayub Alam
UID: TNU2021053200031L
Signature with Date: AyubAlam 20/06/24

Student 3
Name: Ayan Pramanik
UID: TNU2020053100001
Signature with Date: Ayan Pramanik 20/06/24

Student 4
Name: Subarna Das
UID: TNU2020053100011
Signature with Date: Subarna Das 20/06/24

Student 5
Name: Md Zunnurain
UID: TNU2020053100014
Signature with Date: Md.Zunnurain 20/06/24

ACKNOWLEDGEMENT

We would like to express our sincere gratitude to all those who have contributed to the completion of this project. Firstly, We extend our heartfelt thanks to **Arijit Hajra** for his invaluable guidance, support, and encouragement throughout the duration of this project. His expertise and insights have been instrumental in shaping this work.

We are also thankful to **Think Again Lab** for providing the necessary resources and facilities for the successful completion of this project. The cooperation and assistance extended by the staff and faculty members have been immensely valuable.

Furthermore, We extend our appreciation to our colleagues and peers who have provided assistance and shared their knowledge during the course of this project. Their inputs and discussions have been enriching and have contributed to the quality of the work.

Lastly, we are grateful to our professors of our university for their unwavering support and understanding during this endeavor. Their encouragement has been a constant source of motivation.

Student 1

Name: Salman Faizi

UID: TNU2021053200018L

Student 2

Name:Ayub Alam

UID:TNU2021053200031L

Student 3

Name: Ayan Pramanik

UID:TNU2020053100001

Student 4

Name: Subarna Das

UID: TNU2020053100011

Student 5

Name: Md Zunnurain

UID: TNU2020053100014

Place: Think Again Lab

Date:

Contents

Abstract

ObjectTally is a new approach to counting objects in a given category using template-based counting. This project focuses on developing a robust method to accurately count occurrences of objects of a given class in images. The methodology involves creating a custom dataset consisting of images marked with dotted lines representing the location and number of objects. A deep learning model is then trained on this dataset to identify and locate instances of the target object class. In addition, a unique calculation mechanism based on combined component analysis was implemented to refine the number of objects. The proposed approach aims to achieve high accuracy for target counting tasks and potential applications in various fields such as retail inventory management, crowd counting and wildlife population monitoring. Through experimental validation and real-world applications, ObjectTally demonstrates its effectiveness in accurately counting certain classes of objects and provides a valuable tool for various counting tasks.

Objectives

The primary objective of the ObjecTally project is to develop a template-based counting methodology that achieves high accuracy in object enumeration within a specific category. To achieve this objective, the project aims to:

- Create a custom dataset comprising images annotated with dotted masks to indicate the location and count of objects within the target category.
- Train a deep learning model on the annotated dataset to detect and localize instances of the target object category with high precision.
- Implement a novel counting mechanism based on connected components analysis to refine the object count and improve accuracy.
- Evaluate the performance of ObjecTally on benchmark datasets and real-world scenarios to validate its effectiveness and reliability.

Introduction

In fields as diverse as business, surveillance, wildlife monitoring and environmental research, the ability to accurately count certain classes of objects is very important. Counting targets is a key task in these areas, helping with inventory management, crowd analysis, species population assessment, and more. Traditional item count methods often involve manual counting or simplified automated techniques that lack accuracy and scalability. However, recent advances in computer vision and deep learning have paved the way for more sophisticated and efficient object counting methods.In this context, ObjectTally is a promising solution that provides a model-based calculation method that uses the power of deep learning and connected component analysis. ObjectTally aims to solve the challenges of counting objects within a given category by providing a systematic and accurate approach. This project introduces the concept of ObjectTally and describes its methodology, implementation and possible applications.

Background Study

Object counting techniques:

- **Manual counting:** a traditional method involving manual counting, which is time-consuming, labor-intensive and error-prone, especially when dealing with large data sets.
- **Automated computing:** Previous automated computing methods are often based on simple approaches such as thresholding, clustering or simple object detection algorithms. While these methods offer some automation, they may not be accurate, especially in complex scenarios.
- **Model-based computing:** Model-based computing methods use predefined models or object models to identify and calculate patterns of occurrence in images. This approach offers advantages in terms of accuracy and robustness, especially when dealing with objects of a specific shape or appearance.

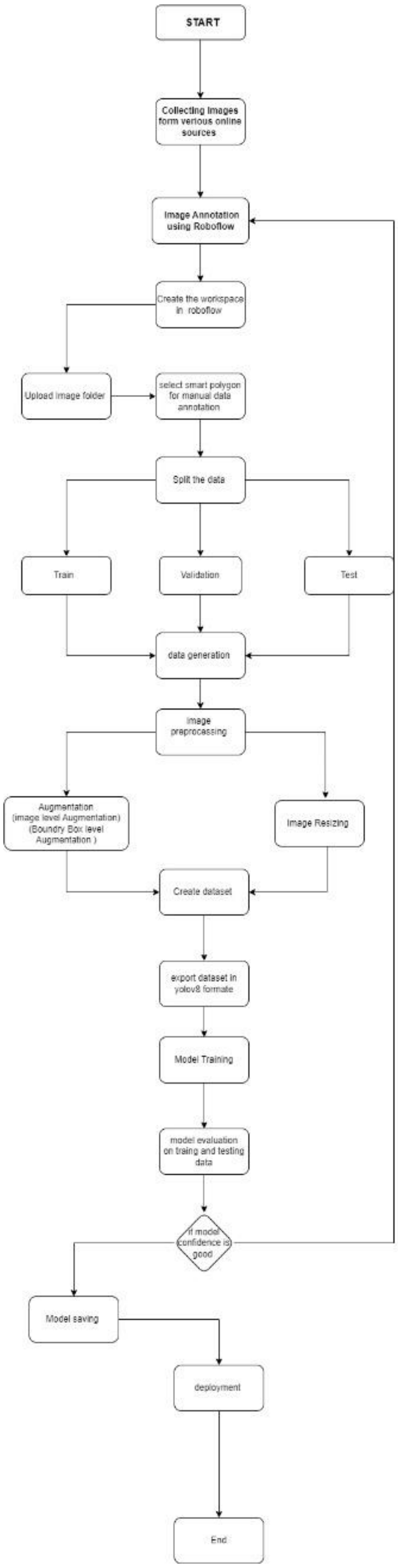
Deep Learning for Object Detection:

- **Transfer learning.** Transfer learning techniques, in which pre-trained models are refined on specific data sets, have been widely adopted to deal with data sparsity and improve generalization in object recognition tasks**.**
- **Data augmentation:** Techniques such as image rotation, scaling, and translation are commonly used to augment training data and improve the reliability of deep learning models.

Connected Components Analysis:

- **Image Segmentation:** Connected Component Analysis is a fundamental image processing and computer vision technique often used in image segmentation tasks. It identifies areas of interconnected pixels with similar characteristics that can be used to count and locate objects.
- **Deep Learning-Based Approaches:** Recent research has focused on leveraging deep learning for object counting tasks, with **CNNs** being the predominant architecture. These approaches offer advantages in terms of flexibility and scalability, enabling accurate counting across diverse datasets and object categories. the evolution of object detection techniques, from traditional methods to deep learning-based approaches like **Mask R-CNN**.

Flow Diagram



Methodology

1. Data Collection and Preparation:

- A custom dataset of images containing objects from a single category is created.
- Each image is paired with a corresponding segmentation mask, where each pixel is labeled to belong to the object or background. Tools like Roboflow is used for this purpose.

Object Category: Choose a well-defined and easily distinguishable object category for dataset. Examples include fruits (mangoes, apples, oranges), or manufactured items (bottles, screws, Steel bar).

Dataset Size: Aim for a dataset size that balances quality and training efficiency. A good starting point could be 100-200 images for a simple category, but more complex objects might require 500+ images.

Image Quality: Use high-resolution images with consistent lighting and minimal background clutter. Ensure the chosen category object occupies a significant portion of the image frame.

Segmentation Masks: Utilize annotation tools like Roboflow to create accurate segmentation masks. Each pixel in the mask should be labeled as either belong to the object or the background.

Data Augmentation: Explain the specific data augmentation techniques used (e.g., random cropping, flipping, brightness adjustments) and their purpose in improving model generalization.

2. Model Training:

- The YOLOv8n segment model is trained on the prepared dataset.

- The model configuration file (.yaml) is customized for segmentation by specifying the number of output channels and the appropriate loss function

YOLOv8 Configuration: Modify the YOLOv8 model configuration file to reflect segmentation. Set the number of output channels.

Training Parameters: Experiment with hyperparameters like learning rate, batch size, and the number of training epochs to achieve optimal performance. Techniques like data augmentation (e.g., random cropping, flipping,resizing ,rotation, shifting) can be used to improve model generalization and prevent overfitting.

Hardware and Software: Consider available hardware resources like GPUs for faster training. Utilize platforms like Google Colab or kaggle solutions offering pre-configured environments for YOLOv8 training.

Model Configuration Modifications: Detail the changes made to the YOLOv8 configuration file. Specify the number of output channels and the chosen loss function.

Hyperparameter Tuning: Explain the hyperparameters adjusted during training and the method used for tuning .

3. Object Counting:

- An inference script is developed to process new images with the trained model.
- The model is predict segmentation masks for each new image.
- Connected components analysis is applied to the masks to identify individual objects.
- The number of identified components will represent the final object count.

Inference Script: Develop a script using libraries like OpenCV to load the trained model and process new images. The script is perform inference on the image, generating a segmentation mask.

Connected Components Analysis: Apply connected components analysis to the predicted mask. This identifies connected regions in the mask, each representing a single instance of the object.

Counting Objects: The number of identified connected components in the mask corresponds to the final object count. The script will display or store this count for further analysis.

4. Challenges Faced:

- **Collecting Appropriate Datasets**: Ensuring the availability and quality of relevant and diverse datasets for training models.

- **GPU Session Time Limit**: Managing the constraints imposed by limited GPU session durations, which can hamper and extensive the training processes.

- **Poor Model Performance**: Addressing issues related to underperforming models, which may arise from insufficient data, or other factors.

- Faced difficulty to detecting small objects, dealing with background clutter, or object occlusions

5. Evaluation:

- The trained model is evaluated on a separate testing dataset.
- Counting accuracy is measured by comparing the predicted object count with the ground truth count in the testing images.

Testing Dataset: Prepare a separate testing dataset containing unseen images of the same object category. This dataset should not have been used during training to ensure unbiased evaluation.

Counting Accuracy: Compare the predicted object count with the actual number of objects present in each testing image. Calculate metrics like mean absolute error (MAE) or percentage error to quantify the accuracy of the counting process.

6. Results Interpretation:

- Analyze the achieved accuracy in the context of chosen object category and the challenges faced.
- Discuss scenarios where the model performed well and identify limitations where further improvement is needed.
- Highlight areas requiring further research to enhance model performance in challenging scenarios.

The achieved accuracy suggests that YOLOv8 segmentation is a promising approach for object counting within a specific category. The model performed well, but limitations are observed. These limitations highlight the need for further research.

7. Real-World Applications:

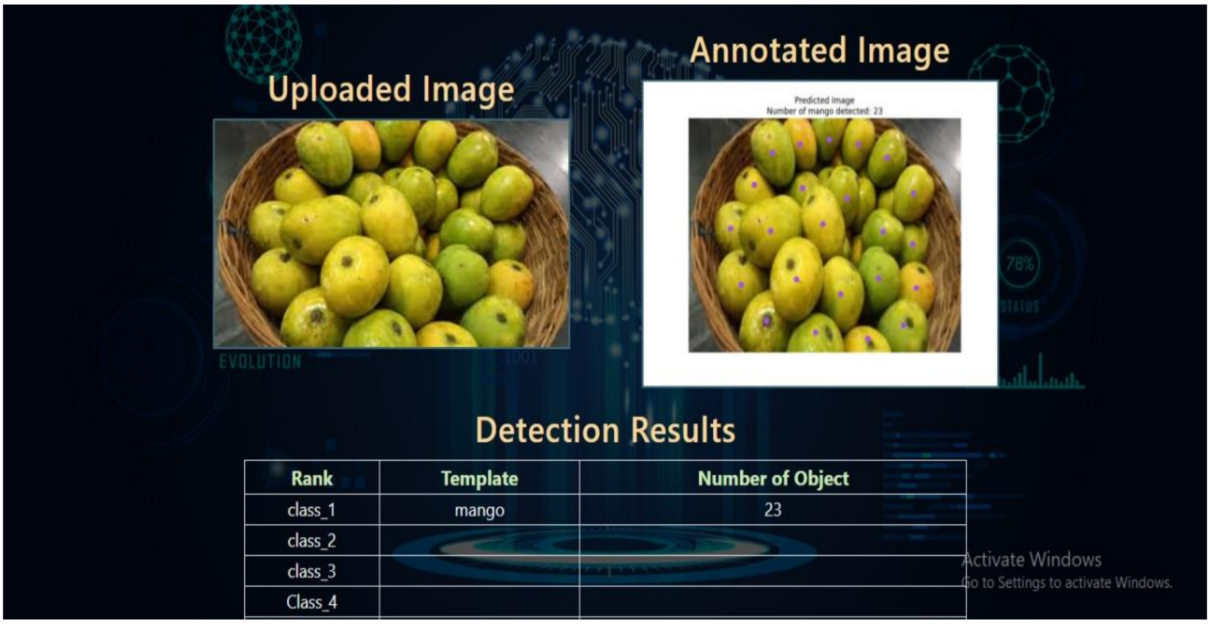
- focused on counting apples, applications could include automated fruit inventory management in orchards or monitoring fruit yield in agricultural settings.

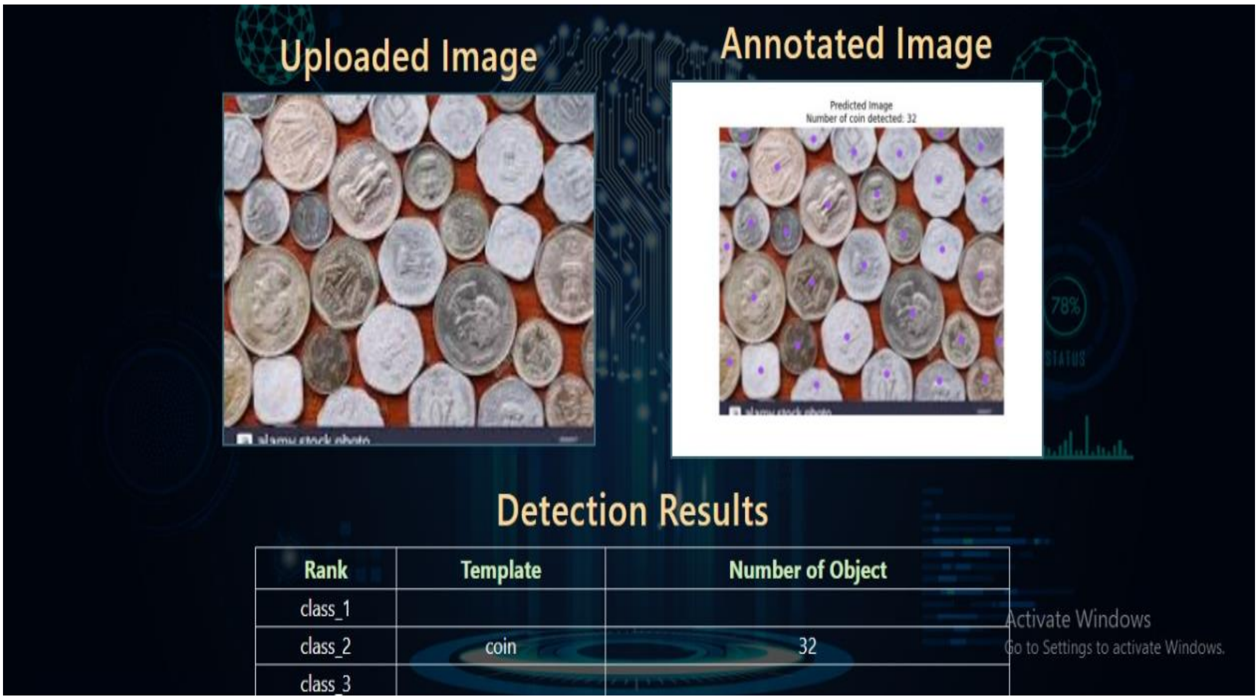
<u>This model has potential applications in areas such as:</u>

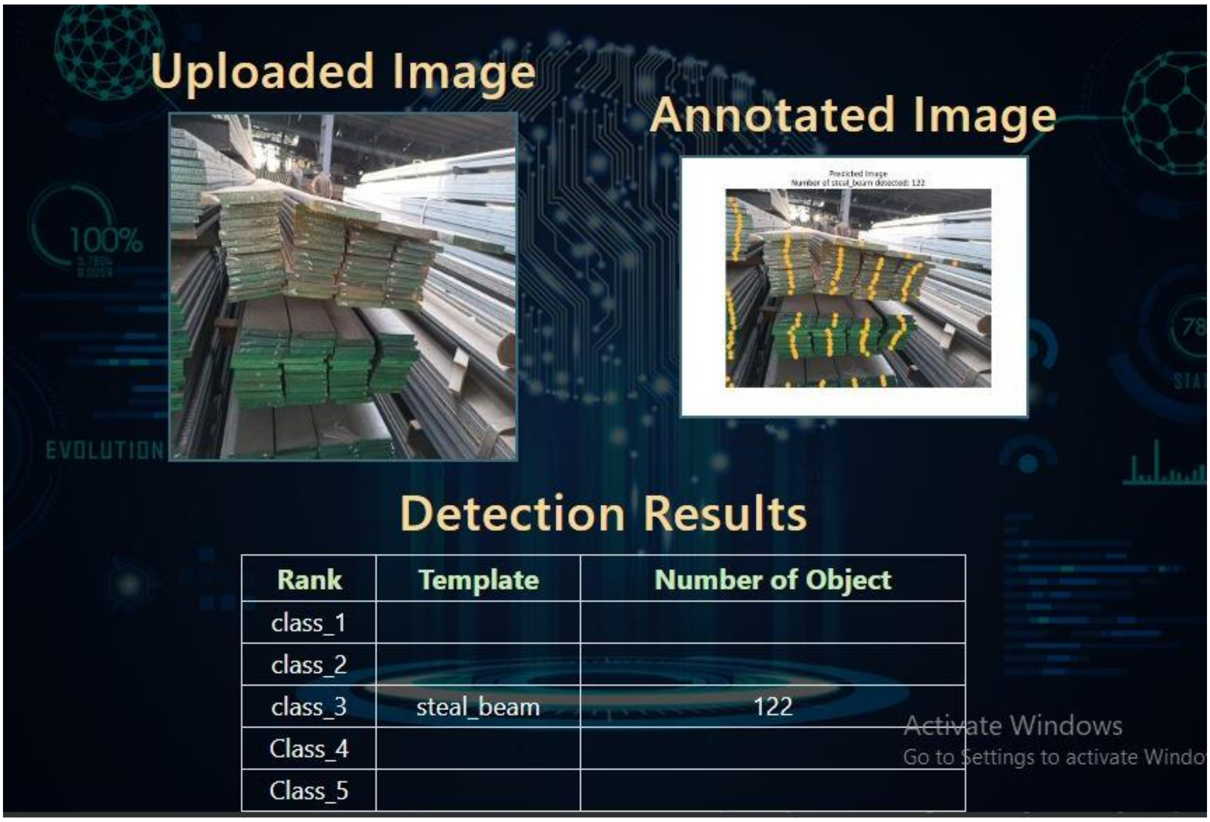
- traffic flow analysis
- automated inventory management
- ecological monitoring

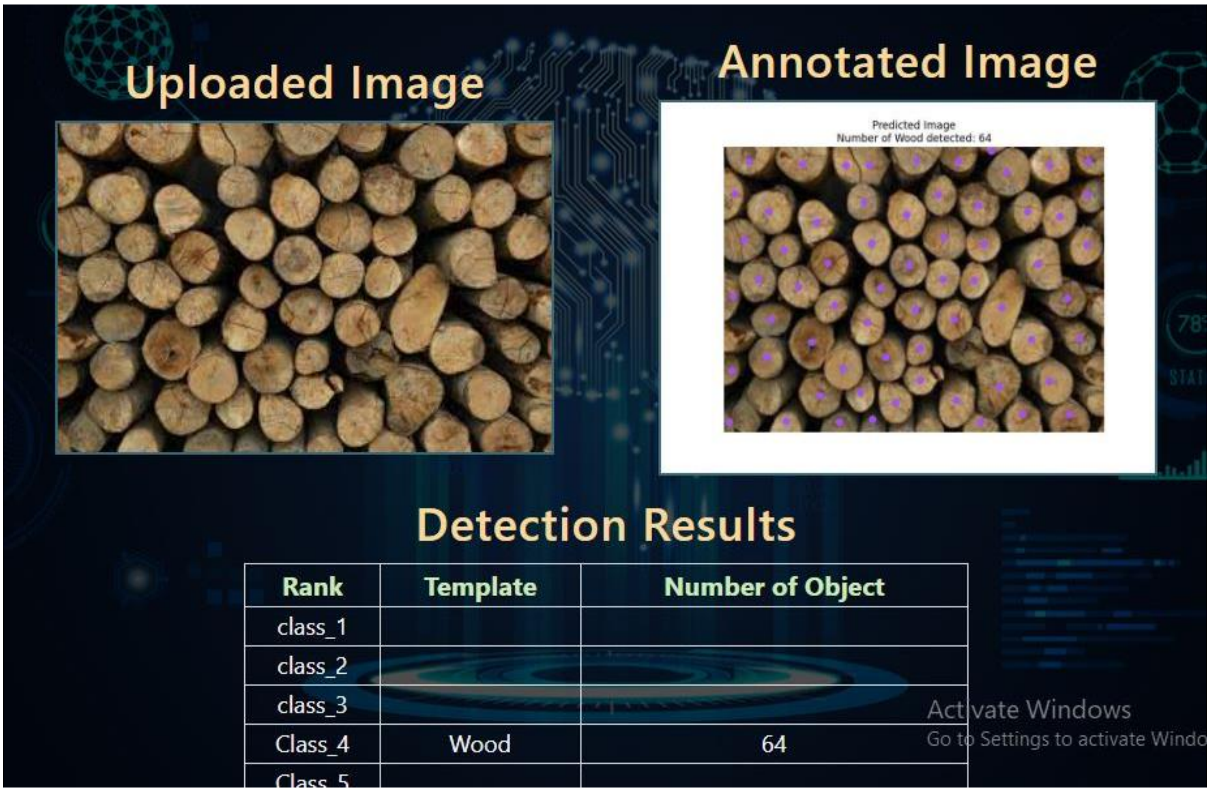
etc.

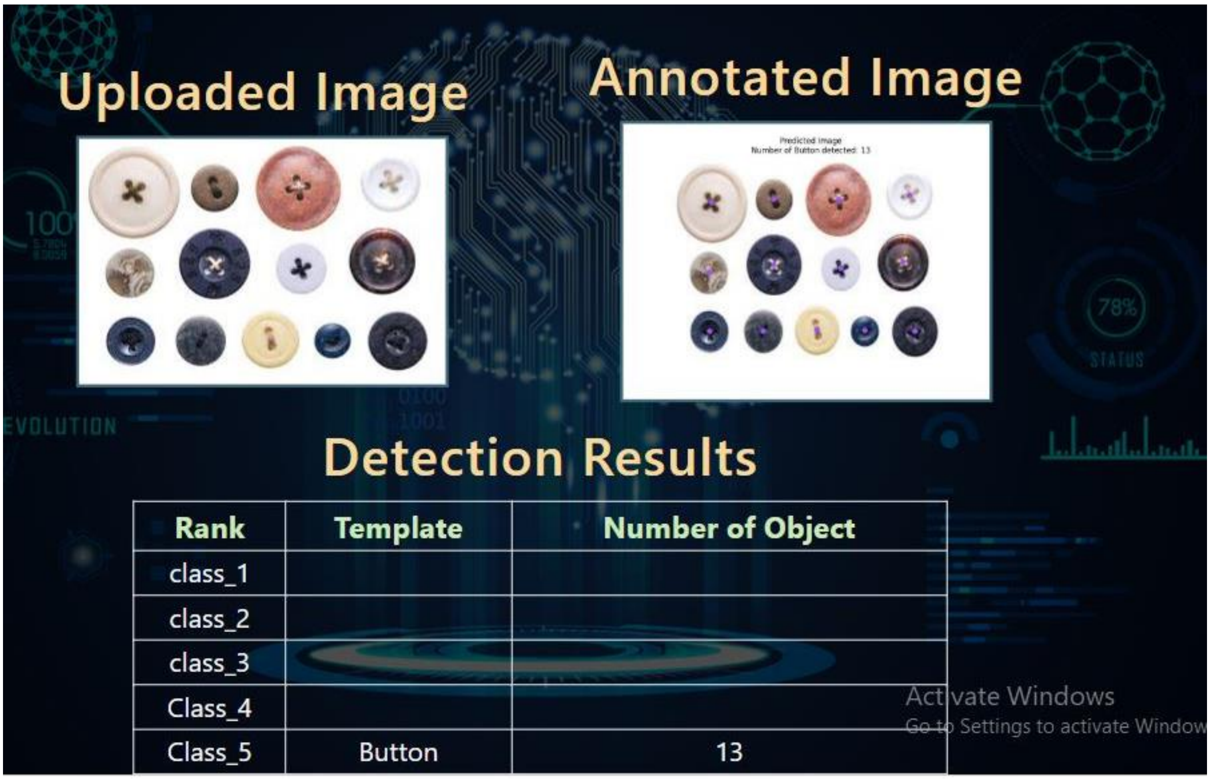
Project Highlights




Uploaded Image
Annotated Image
Predicted Image
Number of coin detected: 32
Detection Results
Rank | Template | Number of Object
class_1
class_2 | coin | 32
class_3
Activate Windows
Go to Settings to activate Windows.


Uploaded Image
Annotated Image
100%
EVOLUTION
Detection Results
Rank | Template | Number of Object
class_1
class_2
class_3 | steal_beam | 122
Class_4
Class_5
Activate Windows
Go to Settings to activate Windo


Uploaded Image
Annotated Image
Predicted Image
Number of Wood detected: 64
Detection Results
Rank
Template
Number of Object
class_1
class_2
class_3
Class_4
Wood
64
Activate Windows


Uploaded Image
Annotated Image
Detection Results
Rank
Template
Number of Object
class_1
class_2
class_3
Class_4
Class_5
Button
13
Activate Windows

Discussion: ObjectTally uses a model-based approach where calculations use predefined models that represent the objects of interest. These models capture the most important features of objects, facilitating effective identification and recognition.

Creating a template:The first step to using ObjectTally is to create a template. Models are created for each object class to be calculated. This involves collecting sample images or frames containing target objects and manually marking their boundaries or key features. Machine learning techniques such as clustering or deep learning can also be used to automate the creation of models.

Pattern Matching: After the patterns are created, ObjectTally performs pattern matching on the input images or video frames. This involves scanning the image to look for areas that closely resemble the models. Various matching algorithms such as normalized cross-correlation or feature-based matching can be used to identify candidate regions.

Counting and Aggregation: After matching the model, ObjectTally aggregates the identified regions to produce a final count for each object class. This aggregation process may involve grouping close observations or applying heuristic rules to handle overlapping items.

8. Conclusion

The object detection web application was successfully developed and deployed using Flask and various machine learning models. The application allows users to upload images and receive predictions on the presence and count of specific objects such as mangoes, coins, steel beams, wood, and buttons. This solution leverages models trained using the YOLO algorithm, ensuring efficient and accurate detection.

The key accomplishments of this project include:

- **Integration of Multiple Models:** The application integrates five distinct object detection models, each specialized for different objects. This modular approach enhances the flexibility and scalability of the system.
- **User-Friendly Interface:** The web interface developed using Flask allows users to upload images easily and view the results, including annotated images and the count of detected objects.
- **Image Handling and Processing:** The implementation efficiently handles image uploads, processing, and saving, ensuring a smooth user experience.

9. Future Work

Despite the successful implementation, there are several areas for potential improvement and future development:

1. **Model Optimization and Training:**
    - **Expand Object Classes:** Train models to detect additional objects, enhancing the versatility of the application.

- **Improved Accuracy:** Continue to refine and retrain models with larger and more diverse datasets to improve accuracy and robustness.
- **Transfer Learning:** Explore transfer learning techniques by pre-training models on a generic segmentation dataset. This approach can help improve model performance by leveraging knowledge from a broader dataset before fine-tuning on specific objects.
- **Post-Processing Methods:** Implement post-processing methods to refine segmentation masks and enhance the quality of detection results.

2. **Real-Time Detection:**
   - Develop functionality for real-time object detection using video feeds or live camera input, which can be useful for various applications such as surveillance or quality control in manufacturing.
3. **User Experience Enhancements:**
   - **Batch Processing:** Enable users to upload and process multiple images simultaneously.
   - **Mobile Compatibility:** Optimize the application for mobile devices to allow users to capture and upload images directly from their smartphones.
4. **Advanced Analytics and Reporting:**
   - Implement features for detailed analytics and reporting, such as generating summaries of detection results over multiple images or time periods.
   - **Visualization Tools:** Provide more sophisticated tools for visualizing and analyzing detection results.
5. **Scalability and Performance:**
   - **Cloud Deployment:** Deploy the application on cloud platforms to handle more significant traffic and larger datasets.
   - **Parallel Processing:** Utilize parallel processing techniques to speed up image processing and model inference.
6. **Security and Privacy:**
   - Enhance security measures to protect user data and uploaded images.
   - Implement privacy-preserving techniques to ensure that sensitive information in the images is handled appropriately.
7. **Community and Collaboration:**
   - Open source the project to invite contributions from the community, fostering collaboration and further development.
   - Create a platform for users to share their custom-trained models and datasets, promoting a collaborative approach to object detection challenges.
8. **Multi-Category Counting:**
   - **Model Extension:** Extend the current models to handle multi-category counting, enabling simultaneous detection and counting of different objects within the same image.
   - **Necessary Modifications:** This extension will require modifications to the model architecture to handle multiple output categories and more complex annotations in the training dataset.
   - **Potential Challenges:** Potential challenges include managing increased computational complexity and ensuring the accuracy of each object category's detection and count.

10. References:

- https://github.com/gpmorales/Face-Mask-Detector-YOLO-Faster-R-CNN
- https://github.com/OmriSR/Transfer-Learning-MASK-R-CNN
- https://docs.ultralytics.com/modes/
- https://docs.ultralytics.com/tasks/detect/
- https://docs.ultralytics.com/tasks/segment/
- https://github.com/Kazuhito00/sahi-yolox-onnx-sample/tree/main/yolox/model
- https://github.com/MeryemBenSalem/PySide6-GUI/tree/main/models
- https://github.com/roboflow/supervision/tree/develop/.github
- https://github.com/google-research/simclr?tab=readme-ov-file
- https://arxiv.org/pdf/2104.08391.pdf
- https://github.com/cvlab-stonybrook/LearningToCountEverything