Vasantdada Patil Pratisthan's College of Engineering & Visual Arts,
Sion, Mumbai-22
Department of Computer Engineering

VISUAL INSPECTION FOR INDUSTRIES

VASANTDATA PATIL PRATHISHTHAN'S COLLEGE OF ENGINEERING

VISUAL INSPECTION FOR INDUSTRIES

GROUP MEMBERS

- CHINMAY GAIKAWAD
- SUYASH SALVI
- PRATIK MAURYA

OUTLINE OF PRESENTATION

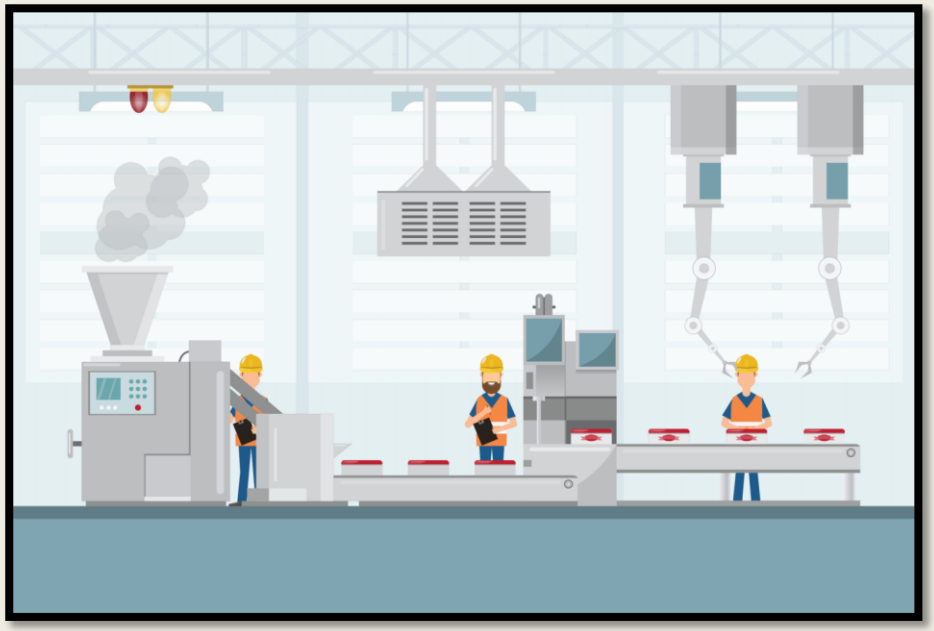
- Problem Definition
- Introduction
- Goals and Objectives
- Motivation
- Literature Review
- Proposed System
- Methodology
- Development Plan
- Software and Hardware Requirements
- Implementation
- Results
- Conclusion
- Paper Publication
- References

PROBLEM DEFINITION

- Problem: The expensive costs of visual inspection systems used in assembly lines.
- Problem Statement: Making systems extremely affordable and scalable according to the needs of the industry

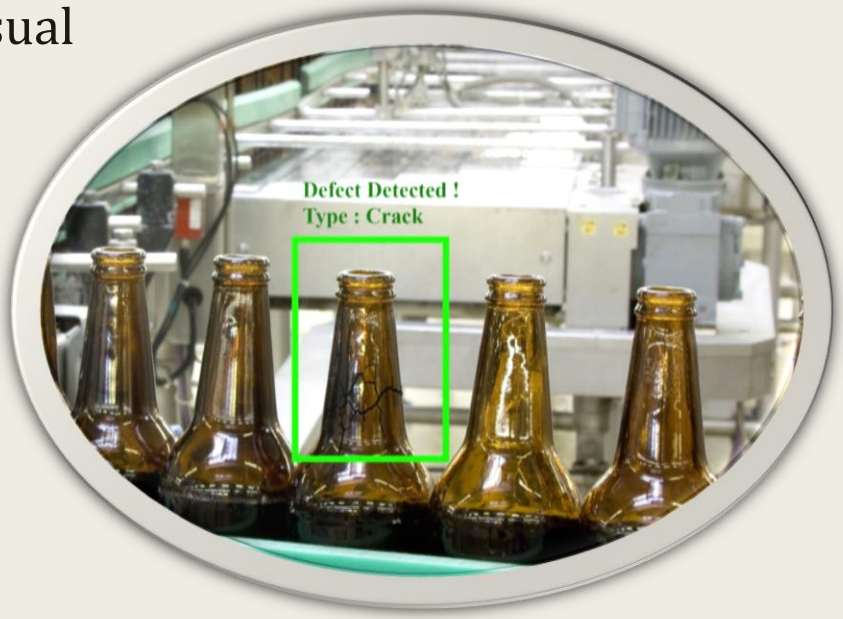
INTRODUCTION

- Vision inspection systems provide image based inspection for your convenience for a variety of industrial applications
- Systems are capable of measuring parts, verifying parts in correct position
- Systems are intelligent to differentiate and take actions on its own



AIM AND OBJECTIVES

- To use Machine Learning for Visual Inspection of assembly line
- Make an accurate model for standard product



MOTIVATION

Infinity Automation Systems PVT LTD	5 lakhs
Rabro Systems	2 lakhs
Analte Control	3 lakhs
Nexgen Robotic Automation Private Limited	4 lakhs
Radheykrishna Technology	3 lakhs
Mayura Automation & Robotic Systems Pvt. Ltd	60k

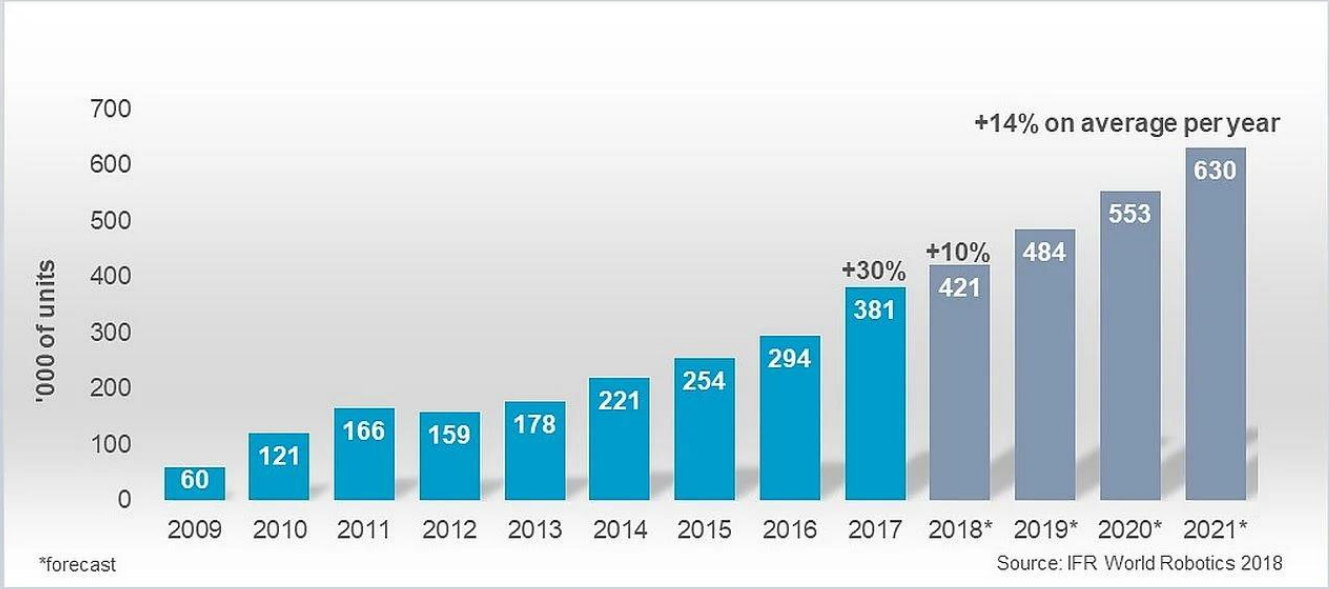
- The visual supervision systems available by the Indian manufacturers are priced at around 60K - 5 lakhs.
- This makes a large investment for small-scale industry. Moreover, the systems are not dynamic if at all the industry wants to scale its production.
- There is a need for a cheaper yet accurate visual supervision system that would look after one of the most important aspects of the product.

QUALITY ASSURANCE!!!

LITERATURE REVIEW

Estimated annual worldwide supply of industrial robots 2009-2017 and 2018*-2021*

IFR International Federation of Robotics



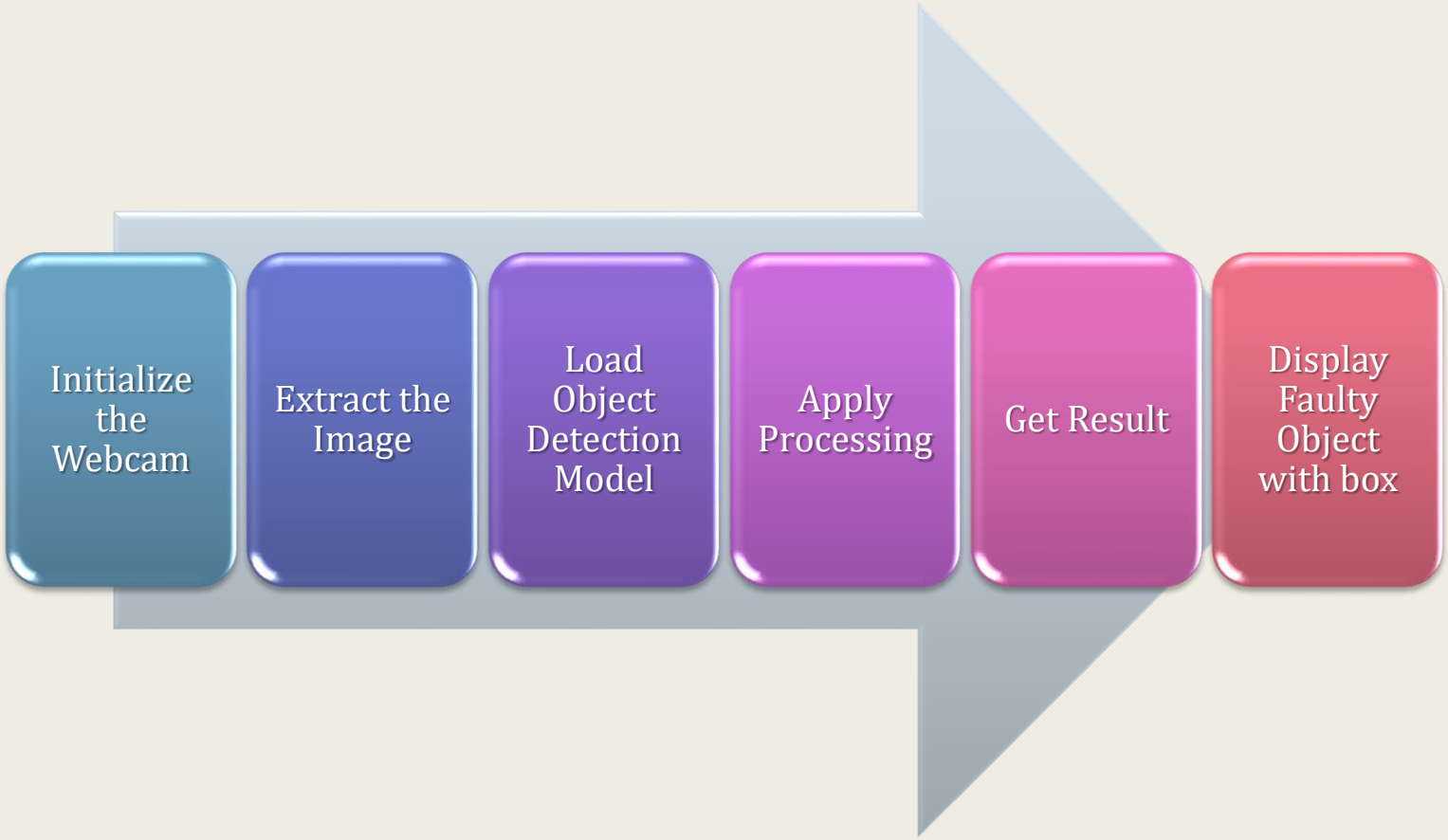
LITERATURE REVIEW

- The industries have evolved, the production speed of the assembly line has changed over time.
- The manual inspectors were inefficient when it came to maintaining the quality of the dispatched products.
- Hence, there was a surge in adaptation of automation in industries for such intricate tasks.

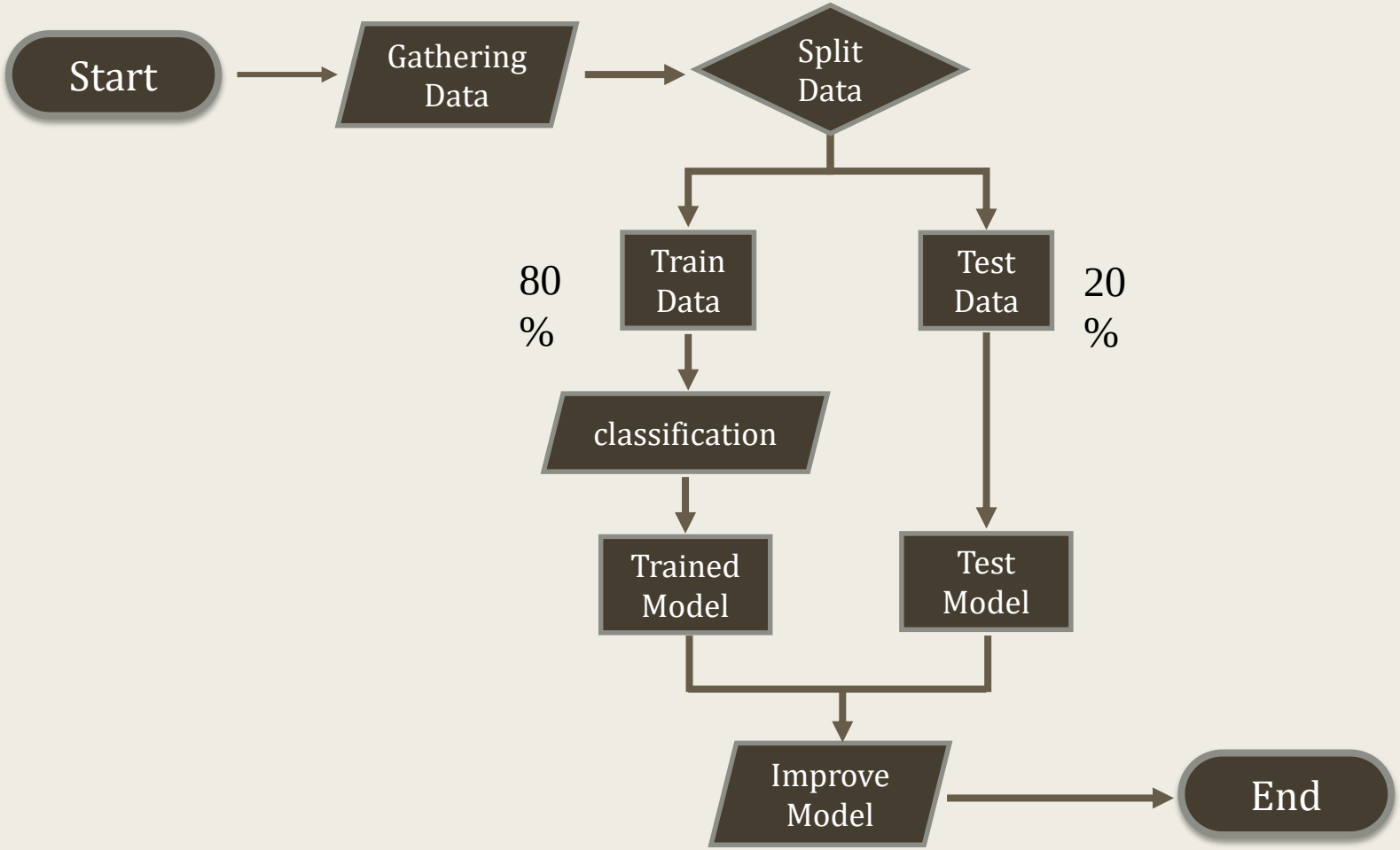
PROPOSED SYSTEM

- To create a visual inspection system
- It could be used by all types of industries(Small, Medium, Large)
- Make it affordable
- The quality control on the assembly line is as automized and accurate

METHODOLOGY



DEVELOPMENT PLAN



Flowchart

DEVELOPMENT PLAN

1. Gathering Data

- The first step in our Project is gathering data.
- This process depends on the project and data type.
- We can also use data from internet repositories sites such as Kaggle and others but for this project we try to make our own data set.

2. Clean, Prepare & Manipulate Data

- Real-world data often has unorganized, missing, or noisy elements. Therefore, for Machine Learning success, after we chose our data, we need to clean, prepare, and manipulate the data.
- This process will be a critical step to our project, and Having a clean data set will help with our model’s accuracy down the road.
- After getting the data to a state we required, we need to convert the data sets into valid formats for our Project.
- Finally, we will split our data into training and test data sets. The training set is used to train the model in the next step, while the test data is used to validate the model in the fourth step. We will split as 80:20 between training and test sets

DEVELOPMENT PLAN

3. Train Model

- This step is the most important step in our project. In this step, We will connect The data set to an algorithm, and the algorithm leverages sophisticated mathematical modeling to learn and develop predictions.
- These algorithms commonly fall into one of following categories:
  Classification – Classify into many categories
  Regression – Predict a numeric
- We will use Classification Algorithm for our project.

4. Test Model

- Now, it's time to validate our trained model. Using the test data from Step 3, we will check the model's accuracy.
- If the results are not satisfactory, we need to improve and retrain our ML model.

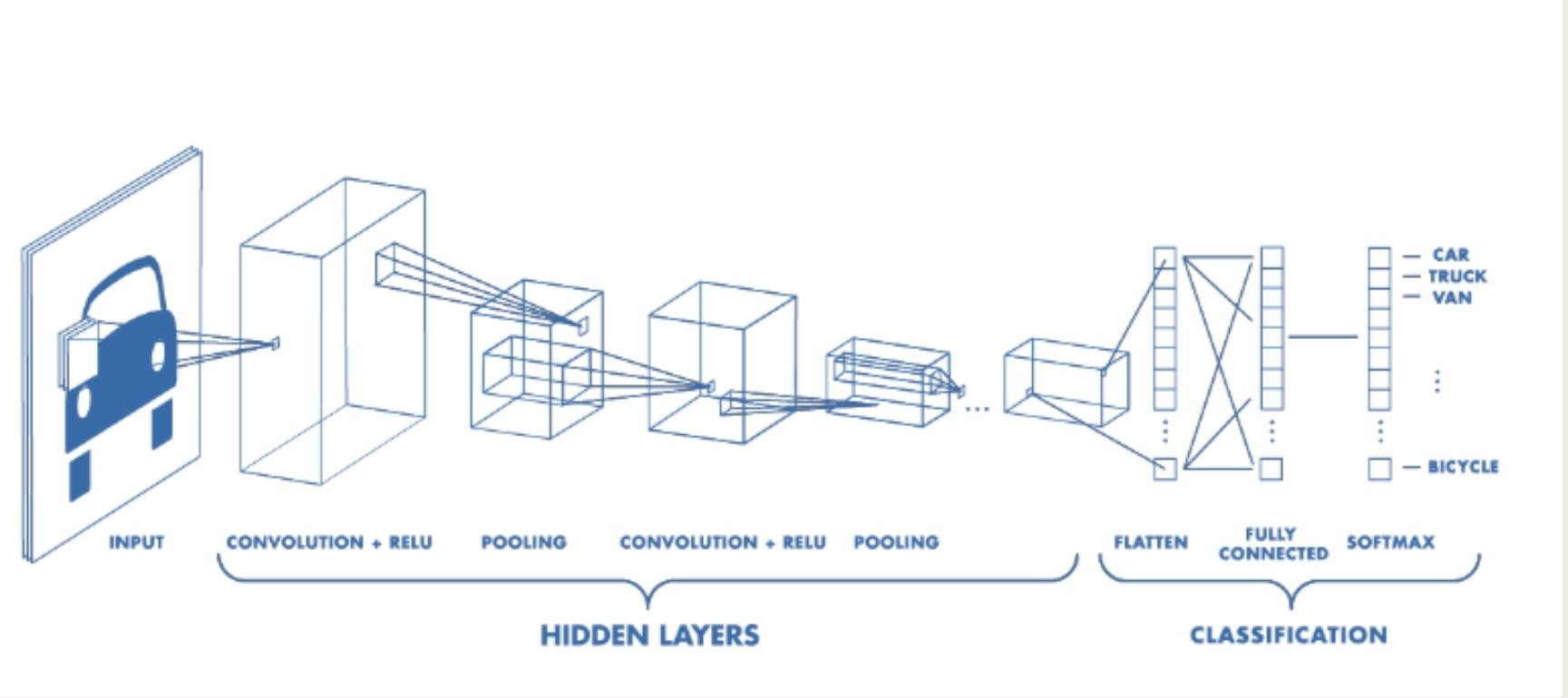
5.Improve

- In this step, we will review our project. To improve accuracy, we need reconsider the algorithm choice.
- Within each class of algorithm, there will be dozens of algorithm choices. A different algorithm may perform better for our project.
- Sometimes small adjustments will have a significant impact.

SOFTWARE AND HARDWARE REQUIREMENTS

SOFTWARE	HARDWARE
• Python • Tensorflow • OpenCV • Jupyter Notebook • Sklearn • MobileNet • CNN	• Camera • 8,10,12,14 bit selectable ouput • 24-bit RGB

How CNN works ?



IMPLEMENTATION

Implementation done :

- Gathering Data,
- Clean, Prepare & Manipulate Data
- Train Model
- Test Model
- Improve

For our project , after discussion with project mentor and with group members, we have selected a common product, which a **Biscuit**. We will make model for Biscuit, Train according to it and Test for it. Our Model will show Biscuit is broken or not and ready for packing. After that the model can be applied for other product by doing respective changes.

1. Gathering Data

we have made data set of product we are going to use, Biscuit. We have collected 100+ photos of Biscuit for making our project dataset.

IMPLEMENTATION

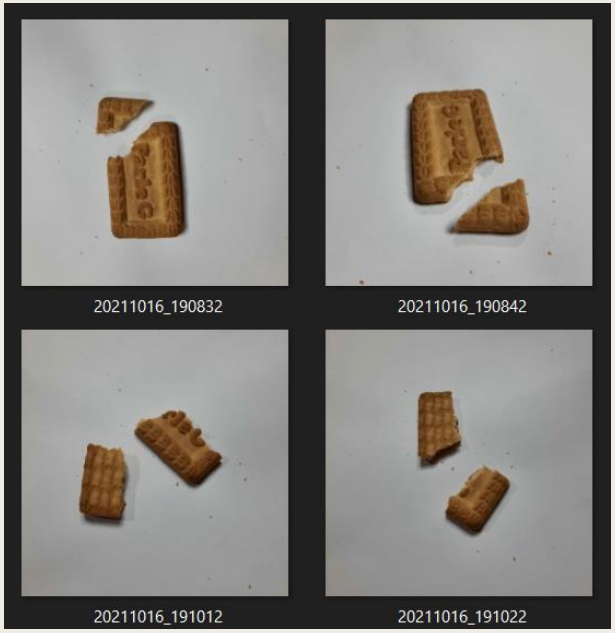
2. Clean, Prepare & Manipulate Data

In this process, we have cleaned our data set by removing unwanted photos. After that we have converted all data set in standard format.

Then, we have split our data in two sets, i.e. perfect items and faulty items.

We also split our 20% of data for testing purpose which is to be done in next sem.

Perfect Items



Faulty Items

IMPLEMENTATION

3. Train Model

In this process, we have chosen classification algorithm for our model. We have connected our processed data set to the algorithm and trained our model

The data is being trained with the 2 distinct outputs individually.

```
1/1 [==============================] - ETA: 0s - loss: 0.7032 - accuracy: 0.56
1/1 [==============================] - 1s 878ms/step - loss: 0.7032 - accuracy: 0.5625
Epoch 13/20

1/1 [==============================] - ETA: 0s - loss: 0.7136 - accuracy: 0.71
1/1 [==============================] - 0s 401ms/step - loss: 0.7136 - accuracy: 0.7143
Epoch 14/20

1/1 [==============================] - ETA: 0s - loss: 0.6598 - accuracy: 0.68
1/1 [==============================] - 1s 868ms/step - loss: 0.6598 - accuracy: 0.6875
Epoch 15/20

1/1 [==============================] - ETA: 0s - loss: 0.6099 - accuracy: 0.68
1/1 [==============================] - 1s 886ms/step - loss: 0.6099 - accuracy: 0.6875
Epoch 16/20
```

	precision	recall	f1-score	support
faulty_item	0.78	1.00	0.88	7
perfect_item	1.00	0.60	0.75	5
accuracy			0.83	12
macro avg	0.89	0.80	0.81	12
weighted avg	0.87	0.83	0.82	12

The trained model is displayed with its distinct features and accuracy. Apparently, the accuracy is 89%

IMPLEMENTATION

As we are suggested to look for other algorithms for our project by our Project Guide and by our Project Coordinator, we list out some algorithms that can be used for image detection.

1. CNN
2. YOLO (You Only Look Once)
3. Spatial Pyramid Pooling
4. Single Shot Detector
5. Histogram Oriented Gradients

We found that CNN works best for real time object detection.

Also, we have extended our dataset, we have cleaned our data and converted in standard format.

We did this step again to improve our models accuracy.

IMPLEMENTATION

Our previously trained model had accuracy of 89% and we successfully improved our accuracy from 89% to 96%.


```
[INFO] evaluating network...
              precision    recall  f1-score   support

 faulty_item       0.92      1.00      0.96        12
perfect_item       1.00      0.92      0.96        12

    accuracy                           0.96        24
   macro avg       0.96      0.96      0.96        24
weighted avg       0.96      0.96      0.96        24
```

IMPLEMENTATION

4. Test Model

In this process, we had completed The object detection model and was capable of detecting our target object. It was a python program for detecting object using created model. Here we integrated all the outputs and produced the required output for the project.

The faulty objects were separated from the perfect ones.

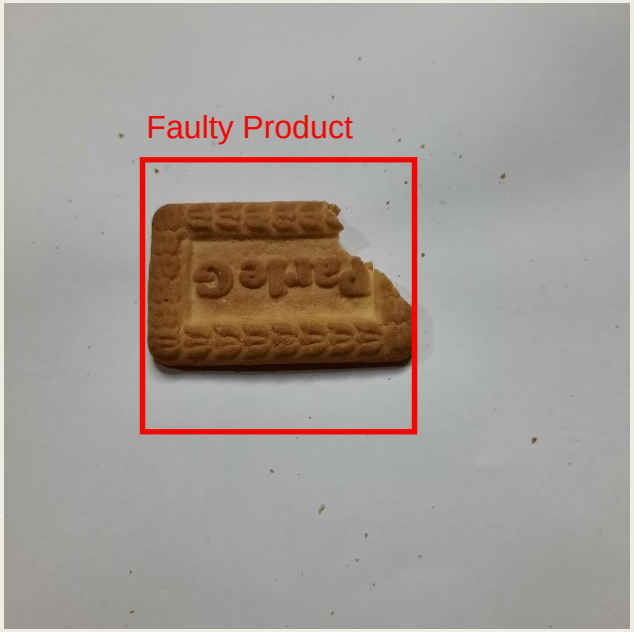
5. Improve

In this step, we have reviewed our project. To improve accuracy, we reconsidered the algorithm choice.

Some small adjustments were done to achieve desired output

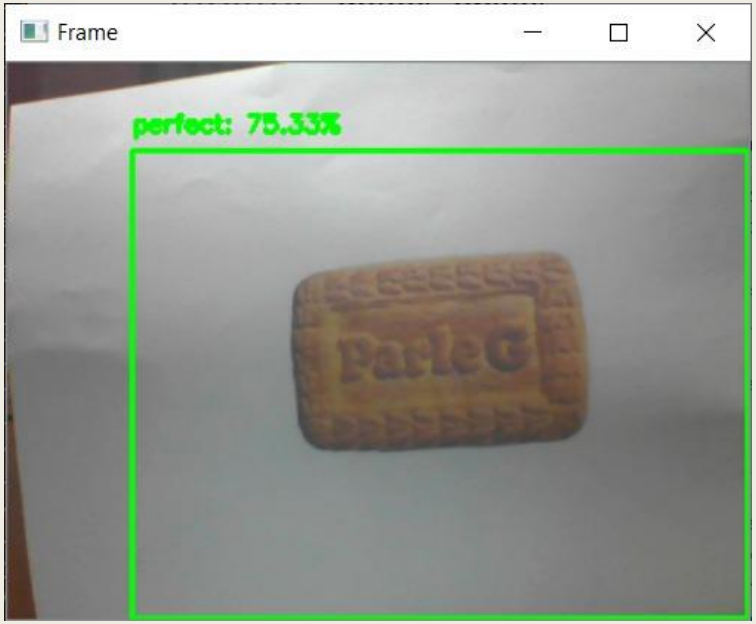
EXPECTED OUTPUT

Perfect Items

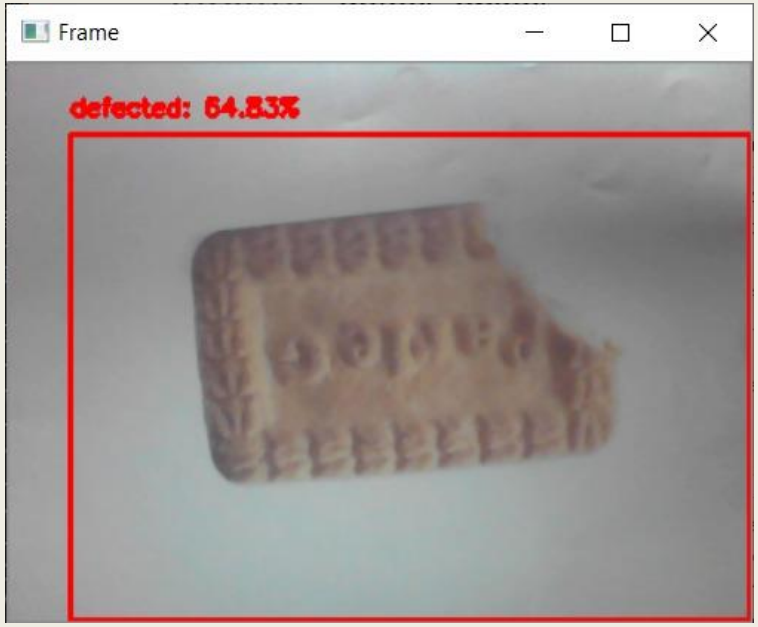


Faulty Items

RESULTS



Perfect Items



Faulty Items

CONCLUSION

- Hence, we created an affordable Visual Inspection System which could be easily used by small scale industrial and large scale industries according to their needs.
- This system also provide accurate result.

PAPER PUBLICATION

in **IJARSCT, Volume 2, Issue 2, March 2022**

International Journal of Advanced Research in Science, Communication and Technology

DOI: 10.48175/IJARSCT-2827

REFERENCES

- Articles
  - AI-BASED VISUAL INSPECTION FOR DEFECT DETECTION by Serhii Maksymenko
  - EVERYTHING YOU NEED TO KNOW ABOUT VISUAL INSPECTION WITH AI by Jamshed Khan
  - THE USE OF VISUAL INSPECTION IN ASSET MANAGEMENT AND QUALITY CONTROL
  - OPENCV FOR COMPUTER VISION APPLICATION by Naveenkumar Mahamkali
  - MACHINE LEARNING IN COMPUTER VISION by Ashraf I. Khan, Salim Al-Habsib
  - OPTIMAL SPEED AND ACCURACY OF OBJECT DETECTION by Alexey Bochkovsiky, Chien-Yao Wang, Hong-Yuan Liao
  - OBJECT TRACKING AND FAILURE RECOVERY by Saravanakumar Soman
  - IMAGE BASED SURFACE DEFECT DETECTION by Prahar Bhatt, Rishi Malhan, Brual Shah
  - UNDERSTANDING OF CONVOLUTION NEURAN NETWORK by Saad Albawi, Saad Al-Zawi
- Book
  - Python for data analyst

THANK YOU