Graduates Research Project Report Template

Name: Muhammad Zaid Kamil (Student Number: 211890395)
Class: CSC 582-01
Semester: Fall 2022

The whole report should be between 10 and 15 pages using double space and 12 points

Abstract

Describe the topic of your report, the content of this report.

In this report we will give a brief introduction of object detection, then we will showcased how we can integrate OO principles to reduce the complexity and provide reuse in the object detection model. After that we will represent the history of object detection which showed the improvement of Object detection field over the years. In section 3, we will define the current problem the object detection is facing. Where there is lack of relationships between object classes and does not perform analysis deeper from the parent class that define specific subclass. In the next section, we will represent two approaches of Object detection these approaches were retrieved from the research papers. We showcases the UML diagram and implementation. In section 5, we will perform our own proposed approach which will be using OO principles for the object detection. The approach involved five steps. In the final section 6 we will briefly provide reasons of the current approaches, identified the difficulties, and our prediction for the future research development of the field of Object detection.

Keywords: Object Detection, Object Oriented Principles, Feature Extraction, Classes, UML, CNN, YOLO, Subclasses, ConfidenceScore, Characterisitcs

1. Introduction

1.1 Brief introduction to the field/topic

What is Object Detection?

Object detection is the task of detecting instances of objects of a certain class within an image. The goal of object detection is to identify and locate objects of interest in an image or video, and to classify them into predefined categories.
it requires the model to accurately identify and locate objects which can be complex.

We use a convolutional neural network (CNN) is trained to learn features of an object from a large dataset of annotated images. We can perform these detections either through images or in real time An example of snapshot of real time object classification detection can be shown in the Figure 1 below: we can see the classification of different objects: such as zebras, giraffe.

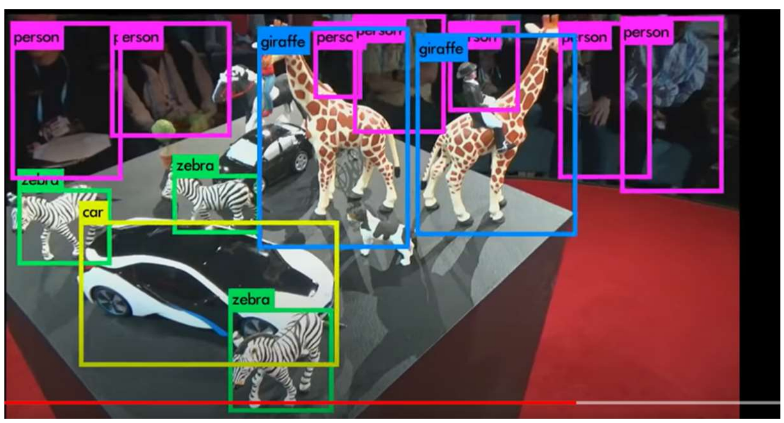


Figure 1: Object Classification using object detection method

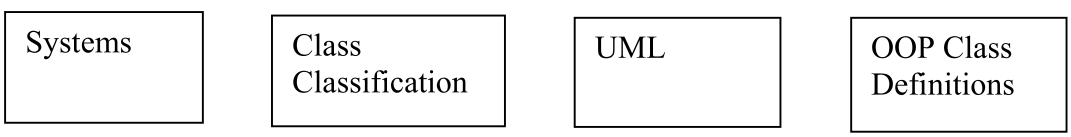
Integrating OOPS to Object Detection

The main features of OOPS include Classes, Objects, Encapsulation (binding data and functions), Inheritance (class acquire properties of another class), Polymorphism.

By organizing the object detection process into objects and methods, it can be easier to modify, maintain, and extend the object detection system. It also makes it easier to reuse code and incorporate object detection into larger software systems.

Each instance of class can be defined as a specific Object Oriented class in the OOP language. The This collection of classes models a fully functional Model. For example, We can integrate Object Oriented Class Definitions.
In this Research Project Report, we will examine the different system, classifying its entities with respect to their classes, convert that information to form a UML (Universal Modeling Language), and then create some class stub definitions in a specific Object Oriented Programming language.

The process can be represented below in a flowchart:



2. History and related work

2.1 Description of the origination and motivation of the field

The history of object detection can be represented in the timeline as shown below:

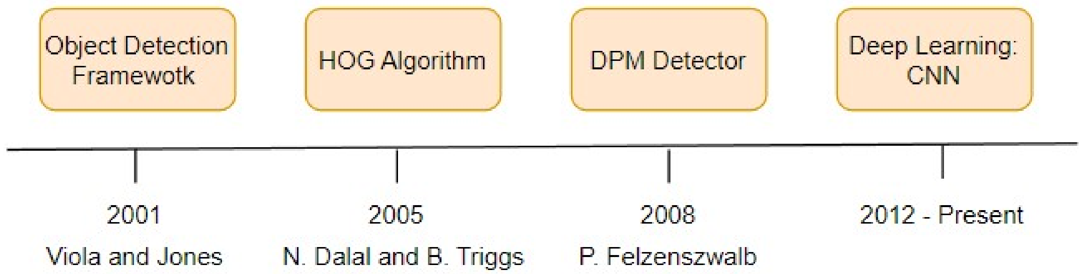


Figure 2: Timeline of object detection method

Object Detection Framework: One of the earliest approaches to object detection was the Object detection framework which allows the detection of human faces in real-time. This was developed by Paul Viola and Michael Jones in 2001.

HOG Algorithm: Was an improvement of feature transform. The basic idea of HOG is dividing the image into small connected cells. Then for each cell it computes histogram.

DPM: Deformable Part-based Model (DPM) is a Training process involves learning a proper way of decomposing an object using divide and conquer' strategy

Deep Learning: This is what we currently use which includes Use of convolutional networks and YOLO for learning high-level classification feature representations of an image or real time.

2.2 Survey literature on this field

For this topic we researched papers in the Object Detection field. The following table represents the research papers reviewed for this report:

Table 1: Literature Survey of Object Classification using object detection method

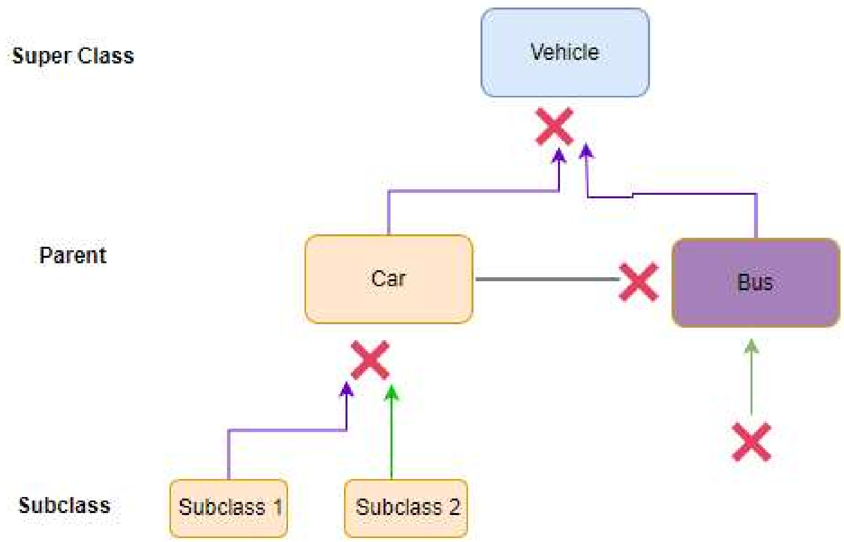
S.No	Title	Author	Concept
	Object Oriented Programming: Neural Networks, Part 1,2 and 3	Lance Dooley	¬Biological system classification ¬UML

			¬OOP Class Definitions ¬Object Inheritance
1	A Learning algorithm for model-based object detection	Chen Guodong, Zeyang Xia, Rongchuan Sun, Zhenhua Wang, Zhiwu Ren and Lining Sun	KEY STEPS: ¬ Object detection ¬Shape Matching ¬Image Segmentation Shape Fragment
	Evolution of Object Detection	Chinmoy Borah	KEY STEPS: ¬ Object detection era
	Object detection with discriminatively trained part based models	F. Felzenszwalb, R. B. Girshick, D. McAllester	KEY STEPS: ¬ Object detection algorithms
	"You Only Look Once: Unified, Real-Time Object Detection", Allen Institute of AI, 2022. Available: https://arxiv.org/pdf/1506.02640v5.pdf	Redmon, S. Divvala and R. Girshick,	¬ YOLO implementation

3. Problem definition

3.1 Define the problem of the field

Object detection is a challenging task, as it requires the model to accurately identify and locate objects of interest in images or videos that may be cluttered or contain multiple objects at different scales and orientations. The current object detection system does not classify every object. For example, if we take a transportation system such as a Car. The Neural networks are trained to only detect the object car. Analyzing deeper from the parent class Car that define specific subclass objects such as Tire, Engine, Steering wheel, Car body which are all subclasses to the parent object car are too complex for the neural network to be trained. This allows Object detection. This is one of the reasons we can use OO principles to represent different subclasses and form class relationships. Otherwise it will be complex for different objects.



As we can see from the above diagram represents the current state of the CNN neural network system for Object detection. Each of the entity act as independent modules, as there are no relationship between the parent class or it's subclasses. The neural network recognizes only one type of object which is the parent class itself such as Car, Bus. Each object is independent to each other this allows no opportunity for reuse of code.

As more objects are classified and trained, this makes the approach more heavy weight and requires more hardware resources which can cause lagging in the detection if use in real time.

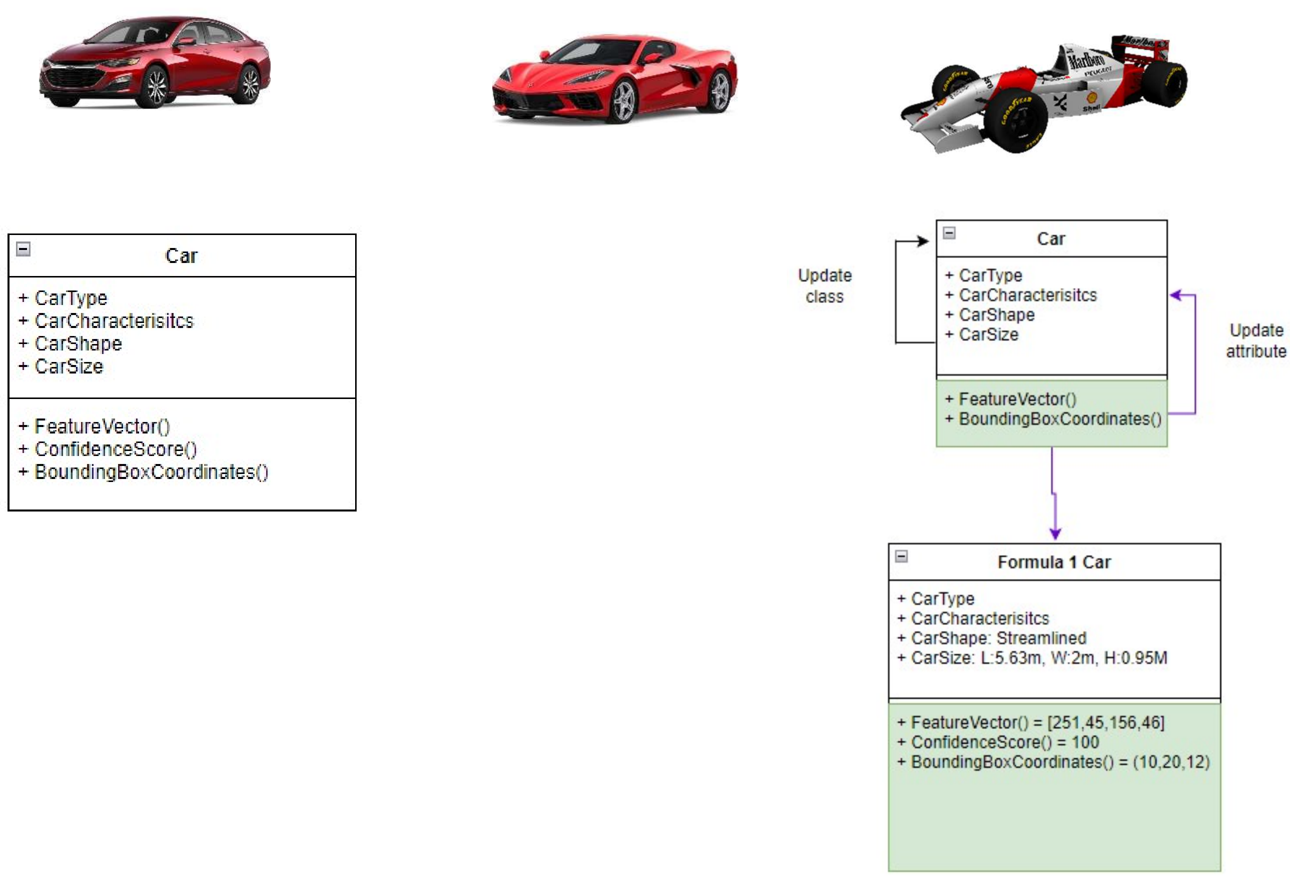
3.2 Define the importance/significance and goals of the field

The goal is to form relationships among different entity classes which improve reusability of code. This allows for faster detection and improves throughput of the program.

We also want to analyze deeper for image analysis. For example the current Object detection system are trained to only detect the object car. By using OO principles, this arise the opportunity to form relationships from parent class to sublasses which allows to define more deeper object detections such as Tire, Engine, Steering wheel, Car body.

Since we define each object as a class which contain attributes. As the CNN model is trained more, the attributes can be updated based on based on the ML dataset.

3.3 Identify characteristics/features of the field



We can see in the above figure that when the neural network is trained for different types of cars: Sedan, Sports and Formula 1. By using OO principles, for each type of car, the attributes of the class can be updated such as the characteristics of the car like the shape, size of the car. Which can in turn update the class name.

If we use OO principles, we can form class relationships between the parent entities, and each parent entity can have subclasses which can be defined. This allows for more precise and detailed object detection image. By using classes we can also update the characteristics and size of the objects which are the Class Attributes. This development of an object detection system more organized and efficient, as it allows you to reuse and extend existing code and abstract common functionality into shared classes.

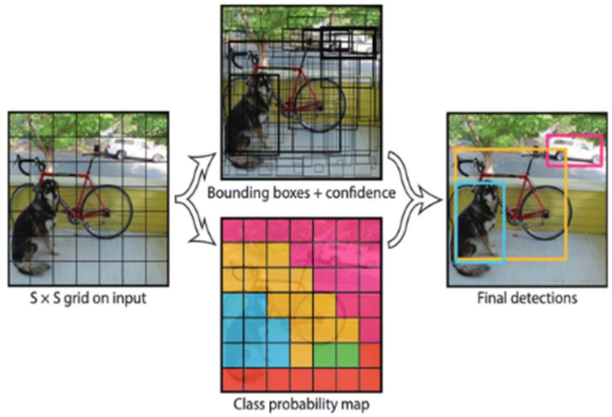
By organizing the object detection code using object-oriented principles, we can make it easier to add new object classes and to maintain and modify the code over time. Objects can be used to represent the objects being detected in the image or video. Each object could be represented as an instance of a class, with attributes such as object type, bounding box coordinates, and confidence score.

4. Existing approaches to solving the problem

4.1 Survey approaches to solving the problem of the field

Approach 1: Object Detection using Neural network and YOLO

The basic idea is dividing the image into small connected cells. Then for each cell it computes the probability of different classification of object. Then replaced with the bounding box that has the greatest class probability in all the bounding boxes under consideration histogram This process is shown in the Figure below which uses YOLO tool to form the classification.



Process of class probability using YOLO[3]

4.2 Identify scope and limitations of each approach

One of the limitations is to create an object-oriented training dataset, the first step is to gather a large collection of images or videos that are relevant to the task at hand.

They have a wide range of applications, including image and video search, security and surveillance, driver assistance, and robotics. For example, Amazon use Robot arms in warehouses that is trained using neural network to separate different object products: ex: electronics, accessories etc

One limitation is that object detection algorithms are only as accurate as the training data they are based on. If the training data is limited or biased, the object detection model may not generalize well to new data. For example if the training data only include Person and Car, then the model would not be able to classify other object other than Person and car object.

4.3 Compare pros and cons of different approaches, if any

There are advantages such as using YOLO tool which detects a variety of objects simultaneously. Provide fast object detection methods and can provide real time detection

Disadvantages: Larger dataset and tools required to train the model require can be challenging for resource-constrained devices such as smartphones or embedded systems.

4.4 Survey approaches to solving the problem of the field

Approach 2: Classification of Biological Objects

A similar approach is used in the Research paper to represent biological systems such as a Neural network. For this problem, the paper represented the relationship between a SensoryNeuron, MotorNeuron, and InterNeuron [1]. They form inheritance to the parent type Neuron super class which is have a strong aggregation to the cell. The class relationship can be showcased in the diagram below where each rectangle represents an entity type called a class

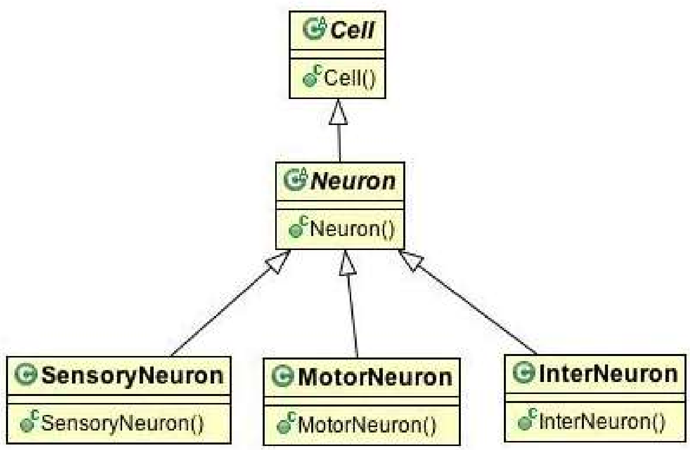


Figure : Relationship of Neuron level classes

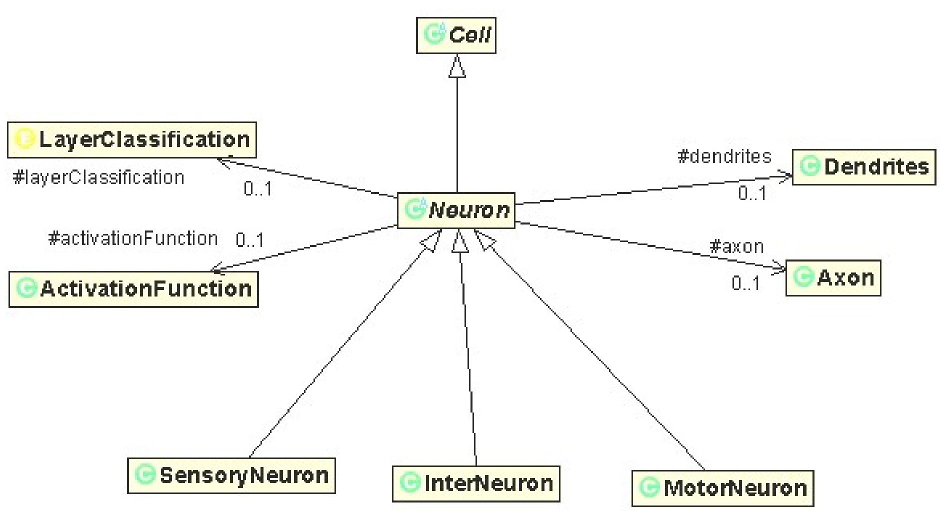
The SensoryNeuron inherits all the attributes of Neuron, which inherits all the attributes of Cell. The more downward we go in a hierarchy, the more specific we become.
In UML and OOP talk, it is legal to say that SensoryNeuron is a Cell.

We can implement the classes in Java or C++ which both support Object oriented programming languages. An example of the five classes can be represented below.

```
public abstract class Cell {
    // fields and functions …
}
public abstract class Neuron extends Cell {
    // fields and functions …
}
public class SensoryNeuron extends Neuron {
    // fields and functions …
}
public class MotorNeuron extends Neuron {
    // fields and functions …
}
public class InterNeuron extends Neuron {
    // fields and functions …
 }[1]
```

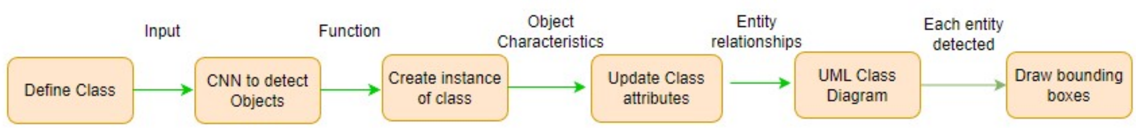
As we can see from the above pseudocode, we use the extend statement for the subclass to gain the same attributes of the parent class. This prevents defining the functions and attributes for each class. We know that the three different types of neurons (SensoryNeuron, InterNeuron, and MotorNeuron) are children of class type Neuron. [1]

A UML Class Diagram for Neuron can be represented as shown below.



The following relationships are implied ...
Neuron is a Cell.
InterNeuron is a Neuron.
MotorNeuron is a Neuron.
SensoryNeuron is a Neuron.
Neuron has a LayerClassification.
Neuron has an ActivationFunction.
Neuron has a Dendrites.
Neuron has an Axon [1]

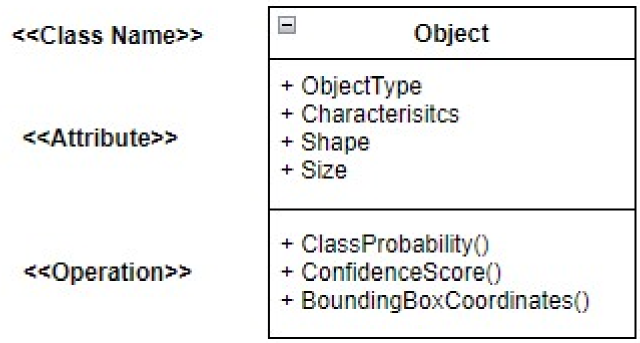
5. Proposed approach, methodology and contribution



Step 1: Define a class for each type of object you want to detect, as described above.
Step 2: Implement a function that takes an input image or video frame and processes it to identify the objects of interest. This function might use techniques such as feature extraction, edge detection, and machine learning algorithms to identify the objects in the image.
Step 3: For each object that is detected, create an instance of the appropriate class and initialize it with the object's characteristics.
Step 4: Update the object's attributes and class name as the object is tracked over time. Form class relationship between object entities
Step 5: Form a UML class diagram to represent the entity relationship between classes
Step 6: Draw bounding boxes to represent the boundaries of the detected entity objects.

Step 1:Define Class:

We can define a base "Object that can be used to represent the objects being detected in the image or video. Each object could be represented as an instance of a class, with attributes such as object type, bounding box coordinates, and confidence score.



From the above Object class, In the Attribute section represent the detected ObjectType, Characteristics, Object shape and size. In the operation section we can represent the ClassProbability this means the probability of the object being detected in the image. The confidence score determines the model shows the probability of the image being detected correctly by the algorithm and is given as a percentage. And the bounding to determine the coordinates for the bounding box (rectangular box) that contains an object.

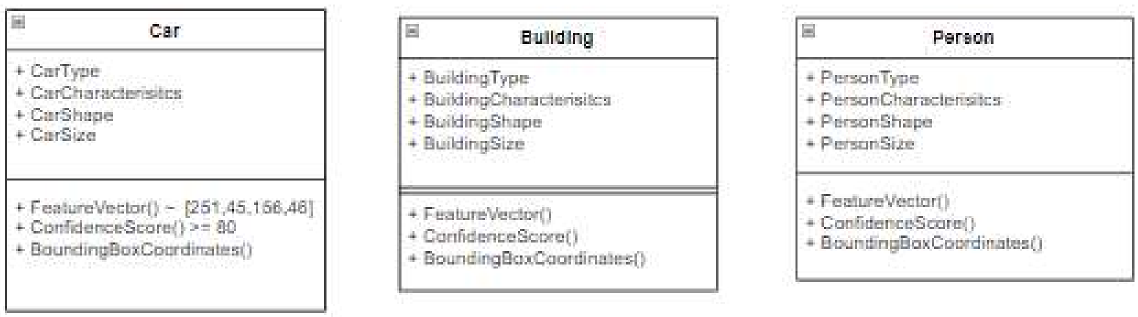
Step 2:Implement HOG Algorithm to extract Image features

The basic idea of HOG is dividing the image into small connected cells. Then for each cell it computes histogram. Then it combines all histograms to form a final feature vector. Where the vector is unique for each face.

Fig. 3. (a) grid cell for image[1] (b) computes histogram for each cell[2]

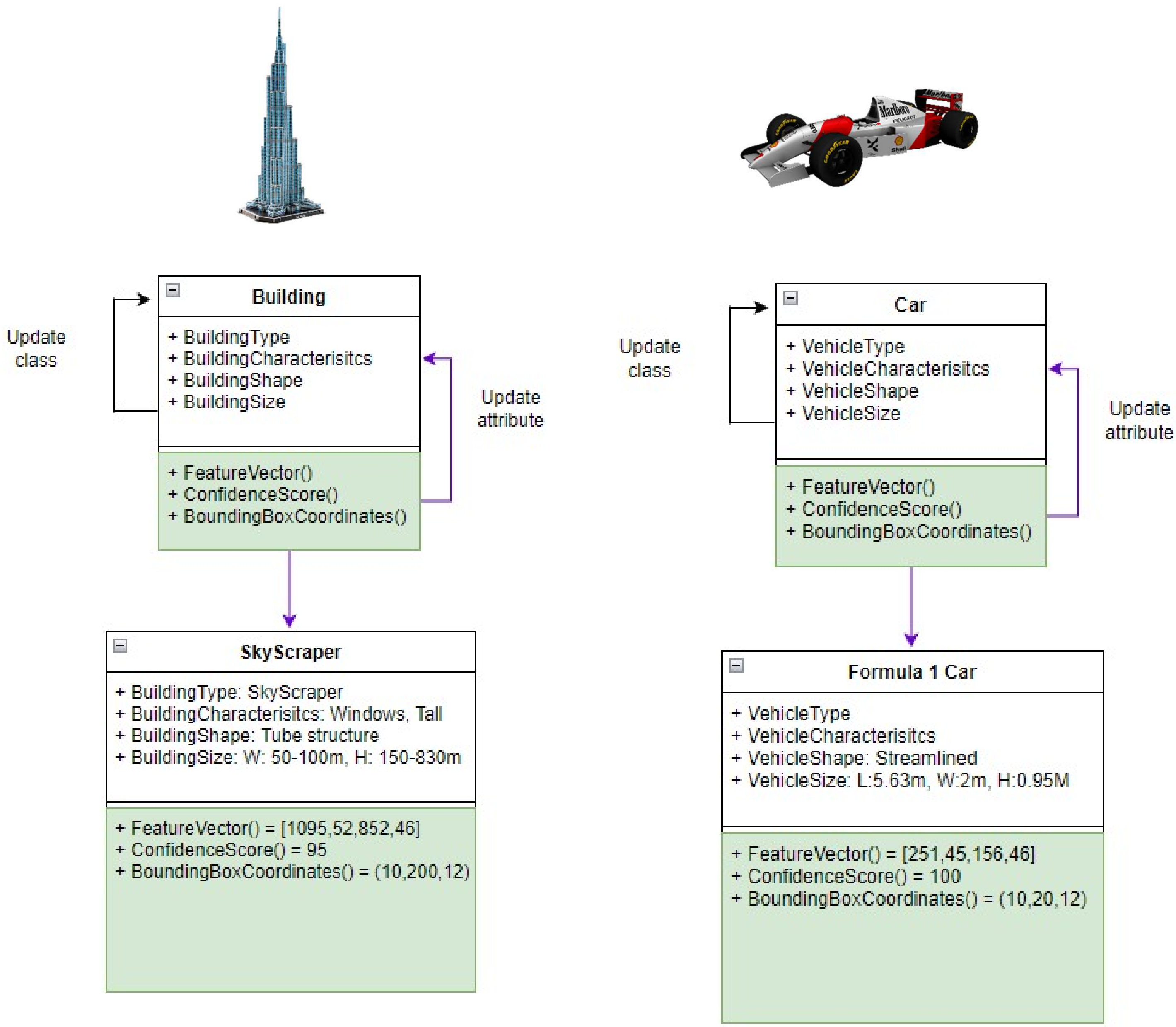
The above Figure 3 (b) brings all the histograms together to form feature vector, which is unique for every object, in the above example the face object vector = [234, 124, 56, …45, 98][2]. This vector is unique to this particular object face. So if the model is to detect another face image it will check if the feature vector value is similar to the new face image.

Step 3:For each object create Class and initialize with Attribute and operations

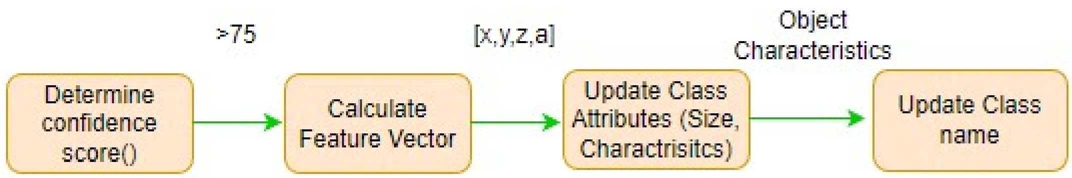


We can create object classes based on our dataset. Object type represents. For example, a Vehicle can be a sedan car or bus, truck. Depending on the size and characteristics on the detected objects we can form new classes such as sports car, sedan.
A building type can be a skyscraper, or apartment type buildings, villa based.
The operation include feature vector as mentioned in the previous step, feature vector values are specific for each object class. We can also get the confidence score which is the probability of the object in the image and based on that we can form the bounding box to represent the detected object.

Step 4: Update the object's attributes and class name as the object is tracked over time.

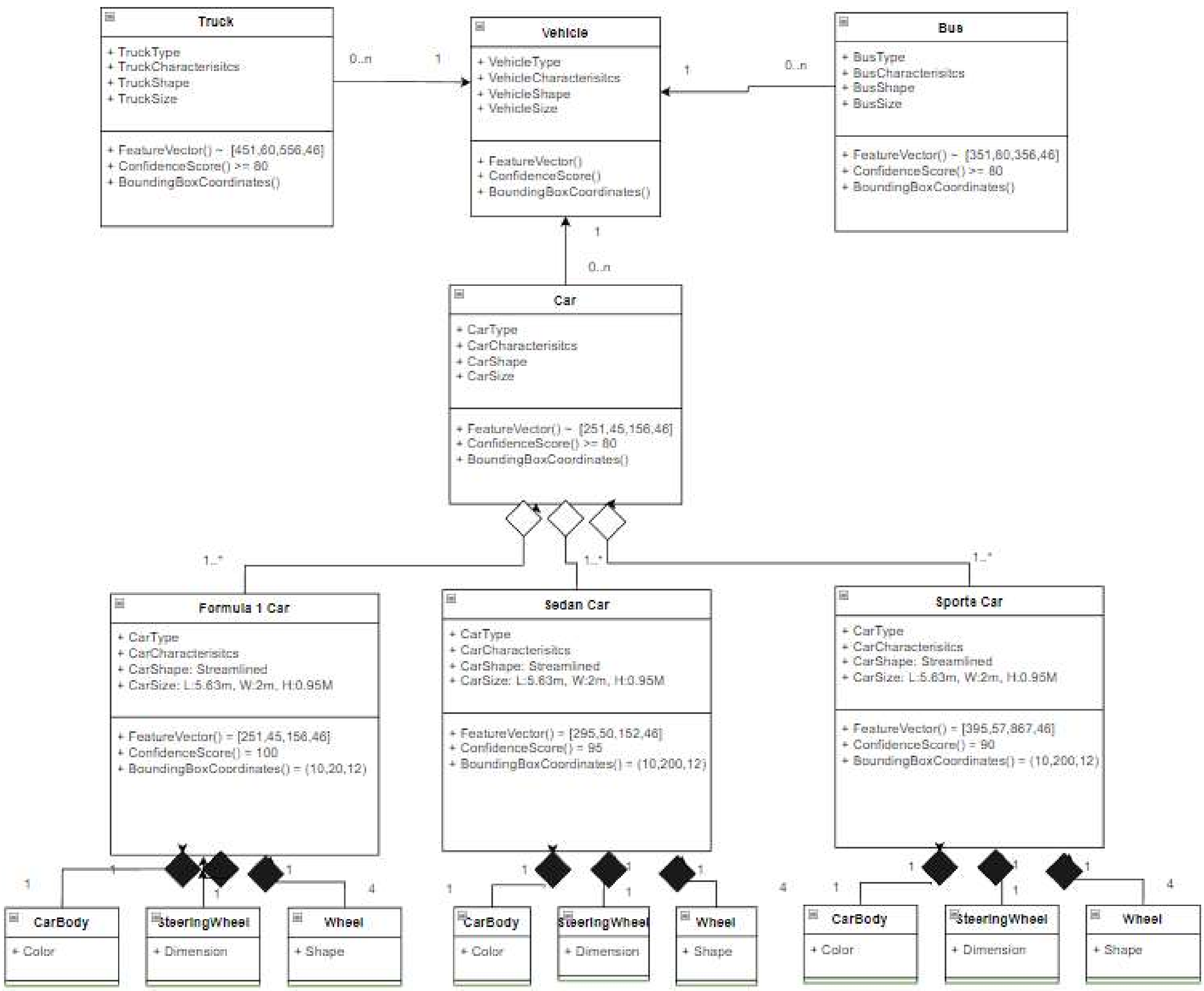


The above representation gives two cases of what happens to the class diagram when the object detection model detects different object instances. Lets take the first example, When the object detection model detects the building, it first checks the confidence score. If it is a satisfactory score, then it calculates the feature vector. If it matches the predefined feature vector of building then it updates the class attributes which represent the characteristics of the building. In this case it retrieves the dimensions of the building such as height, width and characteristics of the building : window. From those attributes it can update the class to SkyScraper. The representation can be shown in a flow diagram below:



Step 5: Form entity relationship between classes

For the vehicle the class relationship between entities can be represented below:



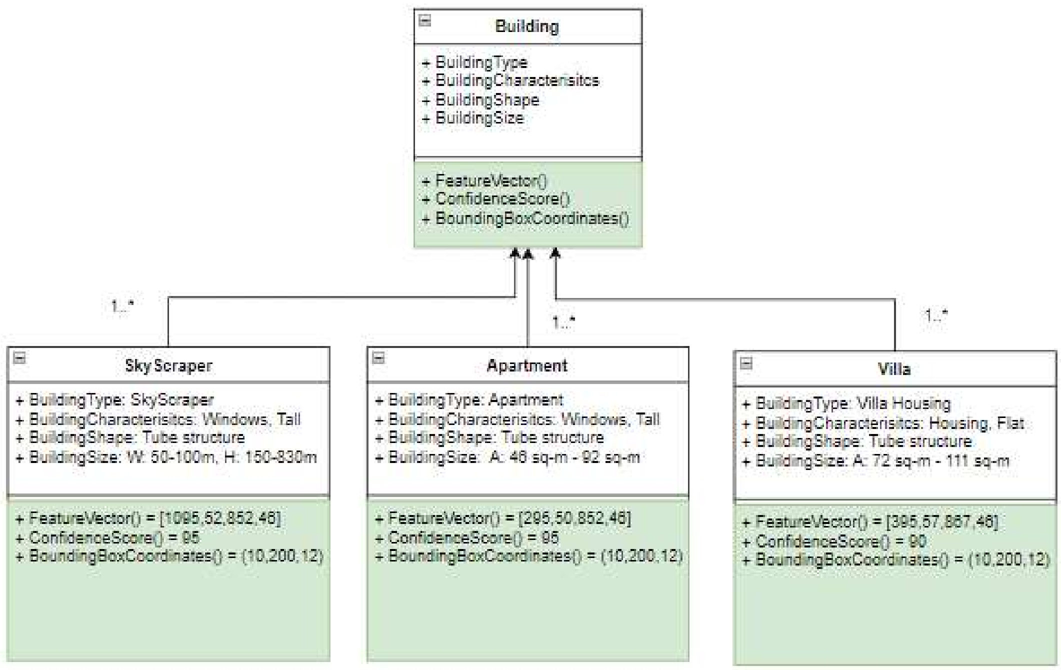
Relationship:

Inheritance: Each car has a CarBody, SteeringWheel, and 4 wheels (4..1)

Multiplicity: A car can be assigned to 1 or more Formula 1 car, Sedan Car or Sports Car.

Composition: The Steering wheel and body of a Formula 1 car cannot be shared with other cars : Sedan car or Sports car.

Class \relationship using the Building entities



calculates the feature vector and the confbuilding attributes. From the feat which can form new classes which will be described in the next step.

which include type of Objects such as Car, Person. If we want to dive in detail we can define subclasses of Object. For example, subclass of Object “Car” can represent Steering Wheel, Tire, Engine etc…

we can prototype a software model that represents a vehicle and its taxonomy relationship with other transportation vehicles.

6. The trend of the field

6.1 Criticize the existing approaches

Computational complexity: The current approach of using Object detection algorithms can be expensive as it requires cost for more hardware resources, especially when using large images or video frames. This can make real-time object detection difficult on resource-constrained devices.

Limited generalization: Object detection algorithms are typically trained on large datasets, and they may not perform well on images that are significantly different from the training data. This can lead to poor performance on uncommon objects.

Ethical concerns: There may be ethical concerns around the use of object detection in other people’s property or facial detection.

6.2 Identify the difficulties or constraints of the field

Occlusion and scale variability: Objects in real-world images may be partially occluded by other objects or have changes in lighting, background which can make them difficult to detect accurately.

Object detection is a challenging task, as it requires the model to accurately identify and locate objects of interest in images or videos that may be cluttered or contain multiple objects at different scales and orientations.

heavy weight and requires more hardware resources which can cause lagging in the detection if use in real time.

6.3 Predict the future development trends of the field

Currently the field of computer vision is relatively new. And there is an increasing amount of research being done in using OO principles for Object detection.The future of the field is likely to continue to evolve and improve in the future. Some potential directions for future progress in object detection include:

Improved accuracy: Researchers are working on developing new algorithms and techniques that can improve the accuracy and robustness of object detection, particularly in challenging situations such as occlusion and scale variability.

Real-time performance: There is a growing demand for object detection systems that can operate in real-time, particularly in applications such as self driving vehicles and robotics. In these forms of applications, it is critical for the object detection model to be at its highest accuracy. To ensure maximum safety for the user.

Researchers are exploring ways to optimize object detection algorithms including using specialized hardware and integrating design techniques such as Object Oriented principles to perform interaction between different classes of object which reduce the complexity and allows for reuse of code. This allows for faster detection and improves throughput of the program. There are many objects to be detected. If a big and complex project is divided Into a set of relational classes, then it is possible to achieve the low production time and cost

7. Conclusion

7.1 Summarization and suggested further work

In this report we started of with section 1 by giving a brief introduction of object detection, then we showcased how we can integrate OO principles to reduce the complexity and provide reuse in the object detection model.

In Section 2, we represented a timeline to showcase The history of object detection which showed the improvement of Object detection field over the years. And we presented the research papers we used for this report in a survey table format

In section 3, we defined the current problem the object detection is facing. Where there is lack of relationships between object classes and does not perform analysis deeper from the parent class that define specific subclass.
also we shoecased how we can update the attribute based on the detected object in the image. This allows for more reuse and reduces the need for developing new classes.

In section 4, we represented two approaches of Object detection these approaches were retreived from the research papers. We showcases the UML diagram and implementation.

In section 5, we proposed on our approach which was using OO principles for the object detection. The approach involved five steps whcih include: 1. Define Class, 2. Use HOG alg to calc feature vector, 3. create instance of class for each object in trained data set.
4. Update the object's attributes and class name as the object is tracked over time. 5. We Form UML class diagram for the Vehicle and Buildings objects. from the entity relationships
In section 6 we gave few reasons of the current approaches, identified the difficulties, gave our prediction for the future research development of the field of Object detection.

8. References

List and number references that you referred in your report

[1] https://www.linkedin.com/pulse/object-oriented-programming-neural-networks-``````part-1-lance-dooley/

[2] S. Jack, "Face detection using dlib HOG", *Medium*, 2022. [Online]. Available:https://medium.com/mlcrunch/face-detection-using-dlib-hog-198414837945# [

[3] Redmon, S. Divvala and R. Girshick, "You Only Look Once: Unified, Real-Time Object Detection", Allen Institute of AI, 2022. Available: https://arxiv.org/pdf/1506.02640v5.pdf.

[4]F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ra- manan. Object detection with discriminatively trained part based models. IEEE Transactions on Pattern Analysis and Machine Intelligence, 32(9):1627–1645, 2010.

[5] Girshick, J. Donahue, T. Darrell, and J. Malik. Rich fea- ture hierarchies for accurate object detection and semantic segmentation. In Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on, pages 580–587. IEEE, 2014.