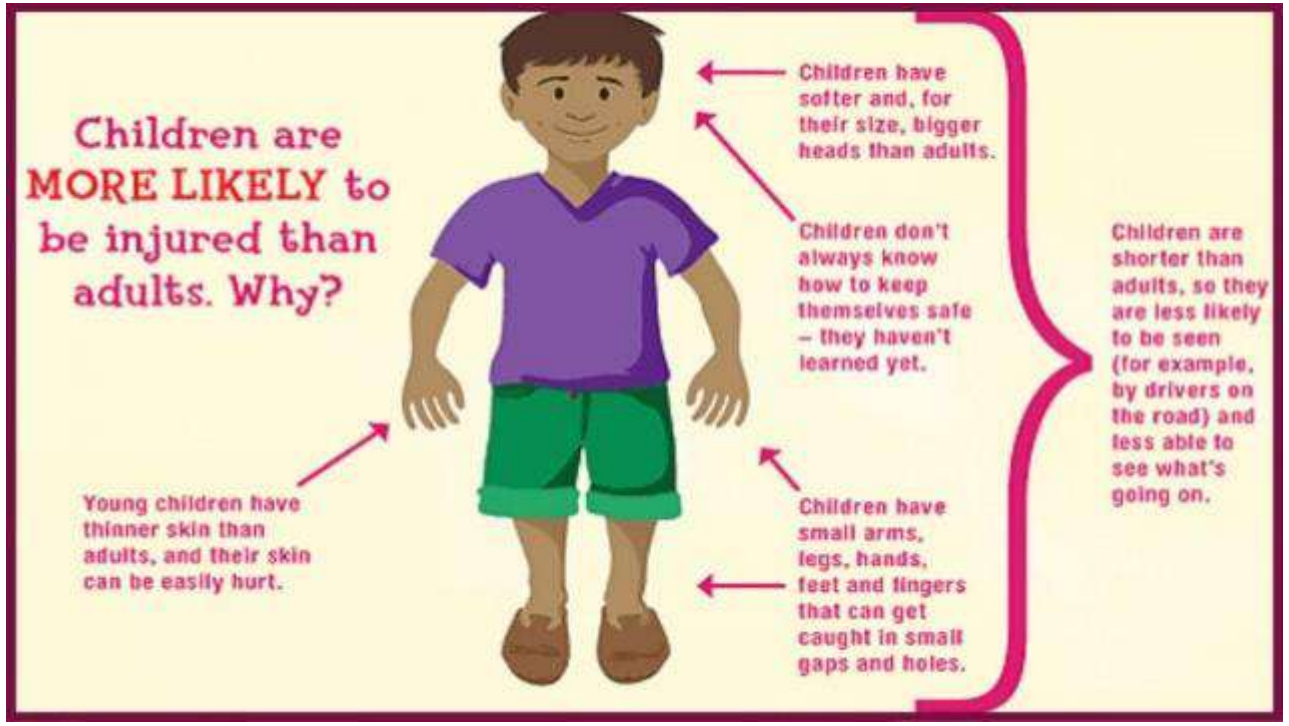
Harmful Child Activity Detection and Assistance

2021-115

Overall Research Problem



- Children are more prone to accidents than adults.
- Lockdown limited the child's playing space to house area increasing the risk of domestic accidents.
- The pressure of engaging in numerous activities within a timeframe of 24 hours made parents occupied with work most part of the day.
- The role of a working mother is sandwiched between two roles: mother and wife and it worsened when corporates changed their rosters to work from home due to the pandemic situation
- Babysitters might look like a quick solution but then again leaves us with the same question, whether it is safe enough?
- Having a babysitter is also not the most popular solution when aligning to different cultures.

Proof of Concept

- To prove the concept that it is realistically possible to create a surveillance-based methodology using computer vision to prevent, falls, leaving safe zone, cuts and burns, kidnap and electric shock related to children in an identified domestic space.

Overall Research Objectives

Main Objective:

- Capturing harmful events (Child hazardous events) and objects (dangerous objects) effectively and accurately and taking prompt responsive actions to avoid the danger.

Objective 1:

- Capture safety zone boundary breach and spot unsafe heights from the current position.

Objective 2:

- Identify injurious sharp objects and hot liquid containers within reach and detect its usage closer to the body.

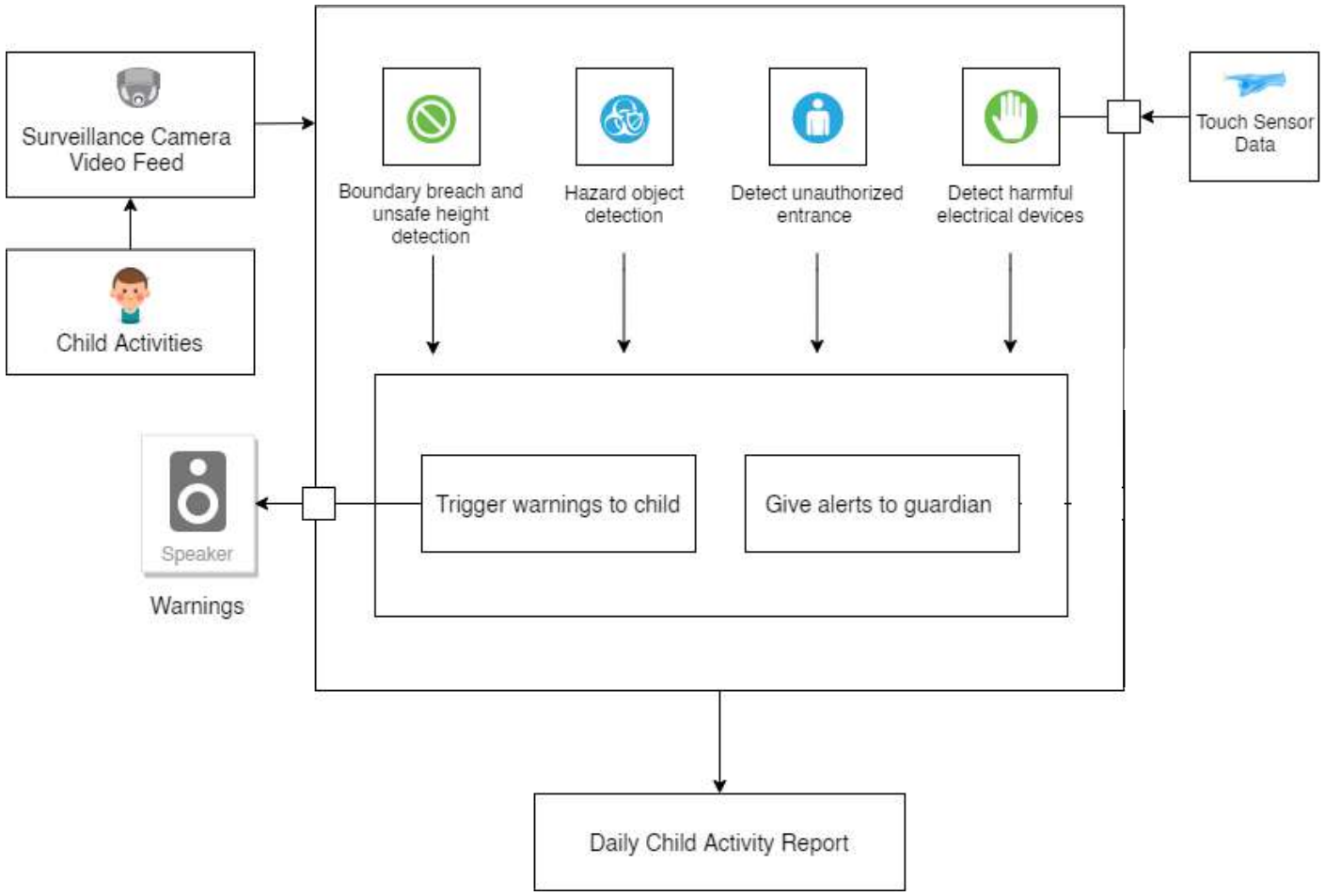
Objective 3:

- Identify unauthorized entrance and take immediate actions to stop possible kidnap.

Objective 4:

- Recognize harmful electric devices in the area and notify when such device is in contact.

Overall System Overview Diagram



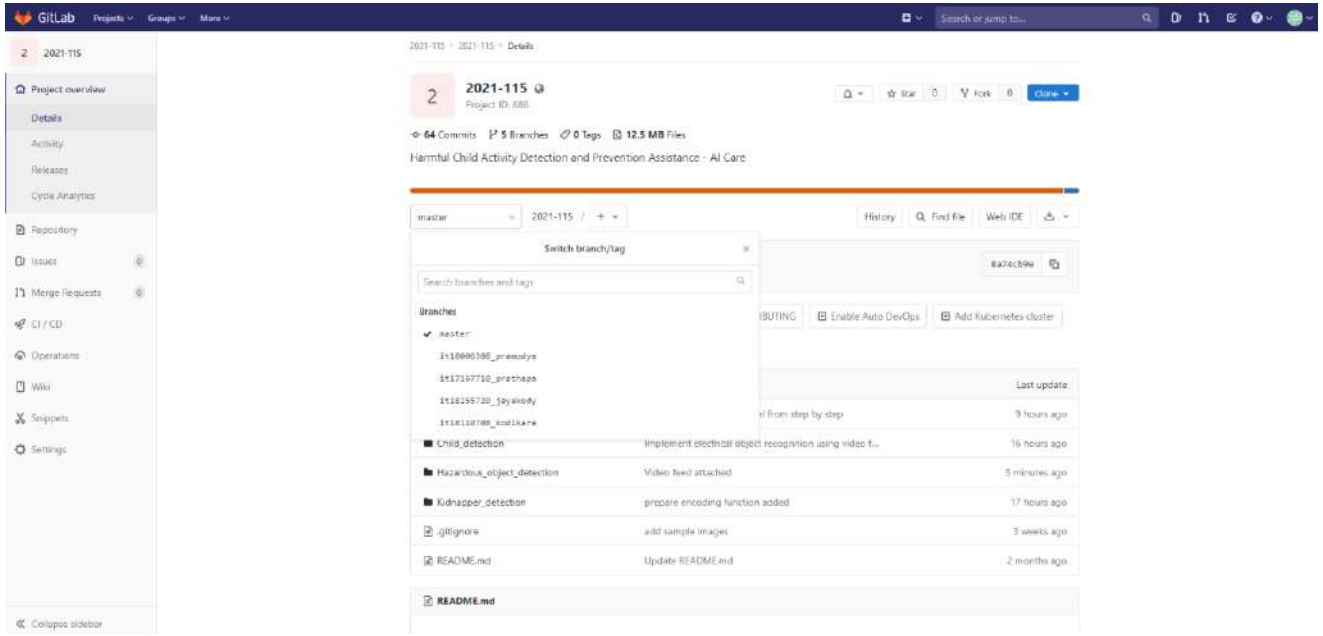
Commercialization

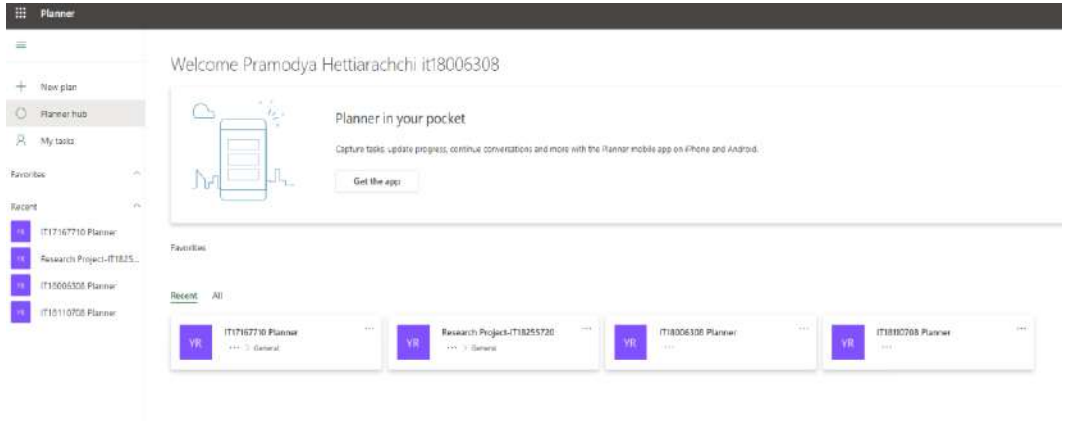
- In a society where families with both parents working has become a common norm, chi ldren has left to grow up by themselves.
- Children between the age 1 year and 5 year is the most crucial period where a child need a lot of parental attention.
- Thereby, AICare has the potential to be the latest trend in childcare in the coming decade.
- Being able to give real time protection assistance to a child when parents are attending to work increase the average working time of an employee. Being able to work from home reduces the number of leaves an employee might take.
- We anticipate that AICare is going to be a top solution companies will invest on providing for their employees because of the high return of investment AICare provides.

Risk Mitigation

- To mitigate technical risks that can occur when integrating GitLab is used.

- To mitigate the scheduling-based risk; that is taking more time to finish the project than planned, MS Planner is used.





Requirements

Functional Requirements

- Integration should be allowed between subsystems.
- There should be a way to identify child separately.

Non-Functional Requirements

- Response time and net processing time.
- Efficiency
- Availability

Personal Requirements

- Parent/Guardian should be available.
- Child should listen to the warnings.
- Parent/Guardian should react to the alerts.

Hardware Requirements

- There should be a way to configure the speaker to the system
- There should be a way to configure the camera to the system

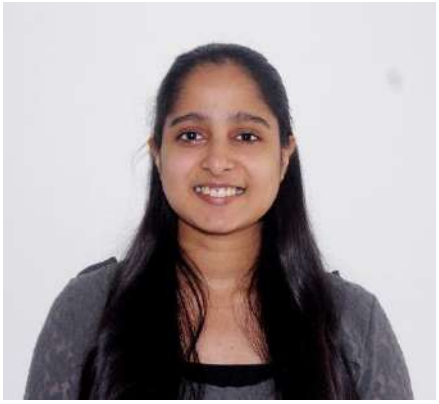
Best Practices

1. Keep code simple and consistent.
   - Reduce unnecessary complexity
   - Easy to read, upgrade and debug

2. Test Continuously
   - Increase code quality and code coverage
   - Ensure all components work together as expected

3. Multiple resources to check code
   - Check each others code
   - Learn from each other

4. Set realistic time estimates
   - Prepare the time breakdowns and task planners at the beginning itself(gantt chart, planners).



IT18110708 | KODIKARA K.A.O.V

Data Science

Research Question

- Kidnap detection is a popular research aspect and has many applications in areas such as
  - Area localizing [1]
  - Bluetooth tracking [2][3]
  - Pose detection[5]
  - Frame based event detection [6]
- All these approaches are based on the person who is kidnapped.
- And these applications does not provide a proper implementation in detecting child abduction/kidnap
- Child kidnap is different and more dangerous than a common adult kidnap.[7]
  - Children are very open and trusting
  - Children tend to believe and listen to adults
  - Children get easily fooled by petty means of kind actions
- A child can be kidnapped without using force and thereby will not be captured by mere action, event and pose detection algorithms.
- Here I seek to prove the concept that it is realistically able to build a methodology of real-time child kidnap detection and prevention assistance based on kidnapper characteristics mentioned follows,
  - Suspicious face covering
  - Quick actions/speed movements
  - Posses harmful objects

Objectives

- **Main Objective**

  - To ensure that children in the early development stages are safe from kidnapping.

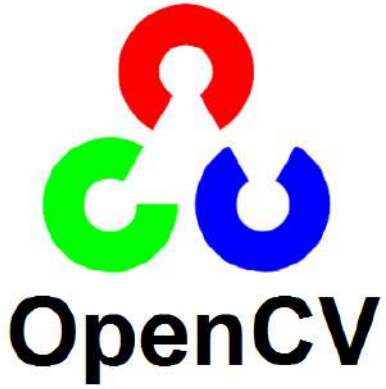
  - This safety is achieved by an intelligent surveillance system that is placed in the area where the child is present.

  -The proposed solution will be implemented with the explained motive by indicating the caretakers or parents of such incidents and to prevent harm to minors as the system makes sure to notify the responsible adults subsequent to detecting kidnapping suspect.

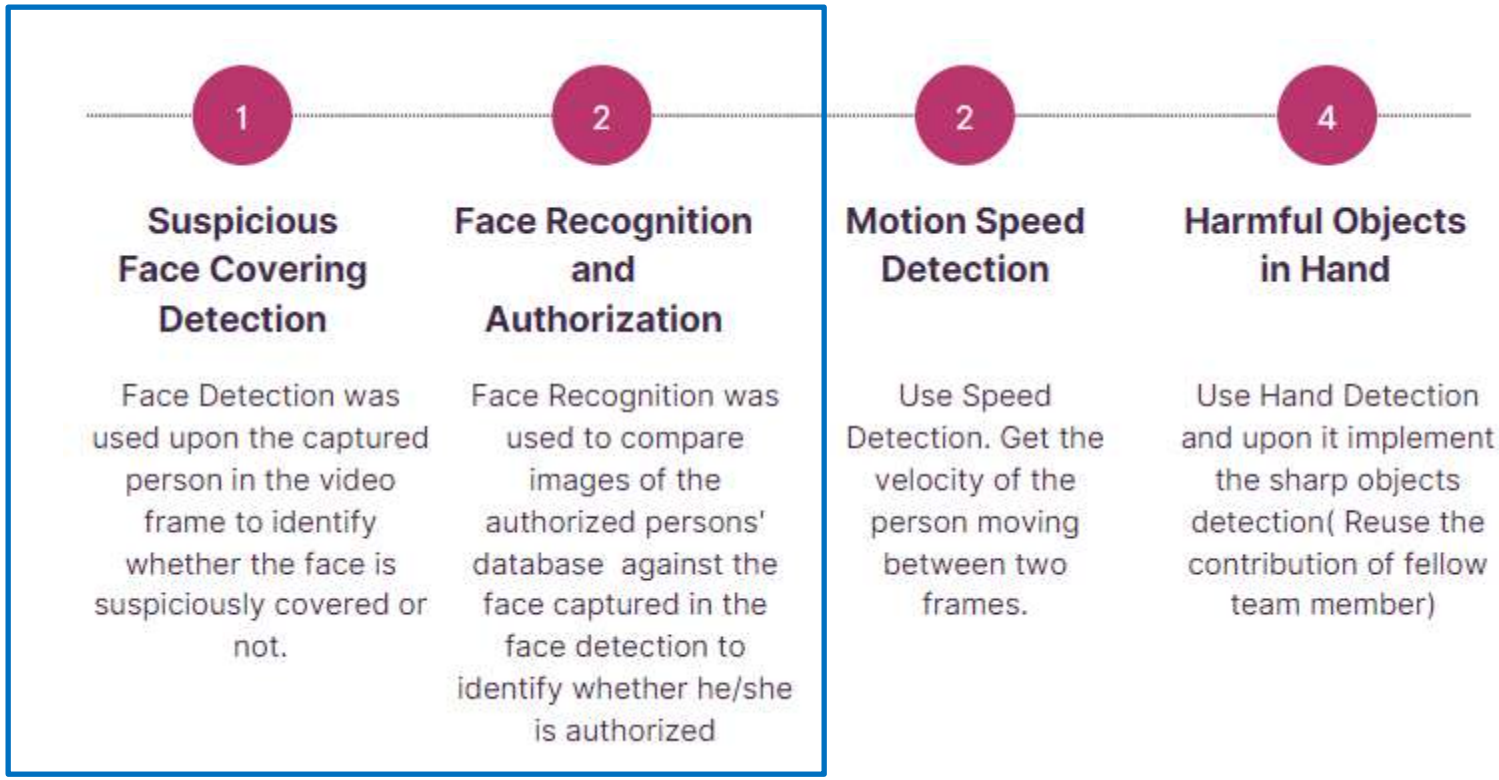
- **Specific Objectives**

1. Identify suspicious face covering
2. Face recognition and authorization
3. Detect motion speed of entering person
4. Identify harmful objects in hand

Technologies



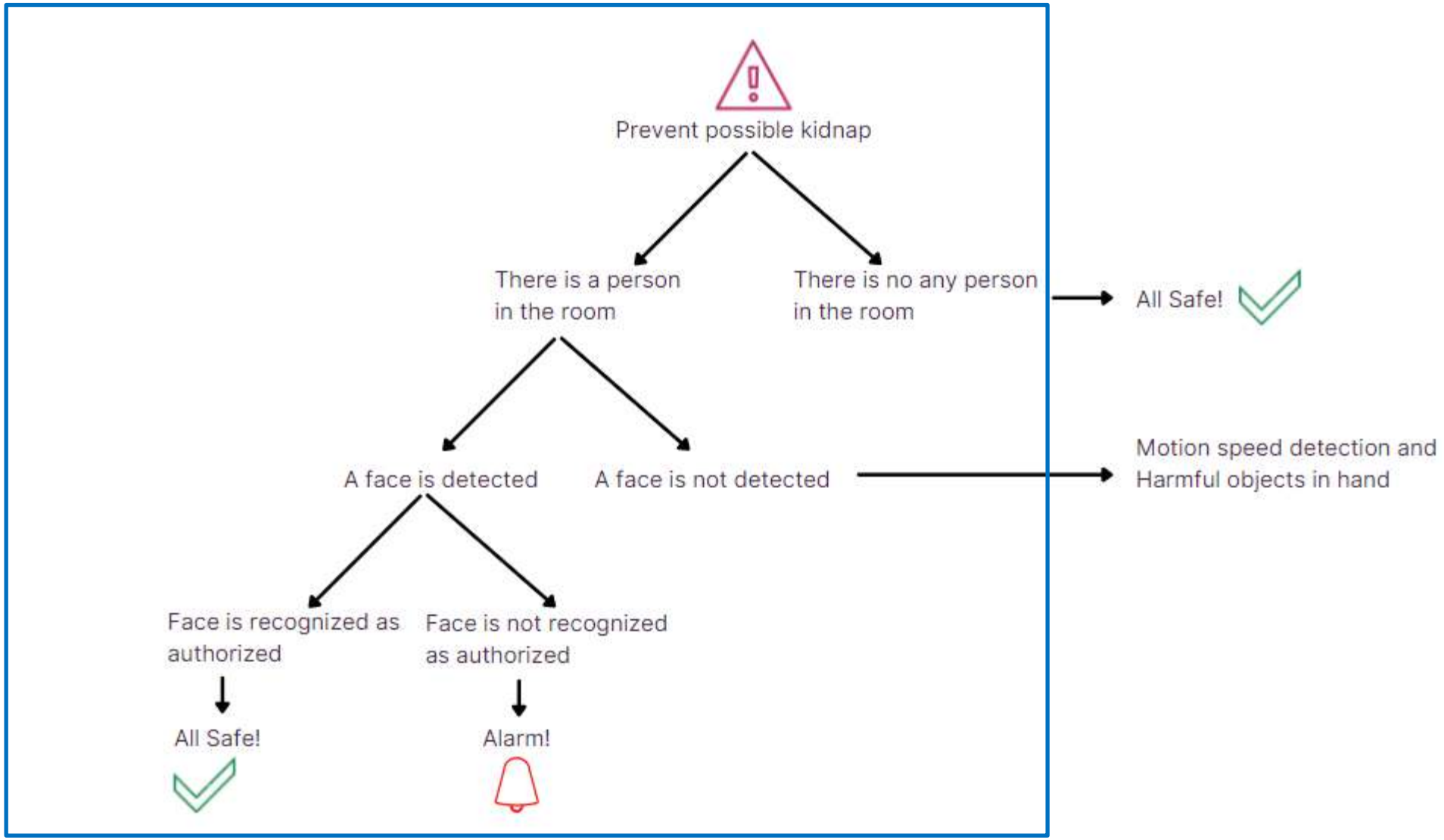
Completion of the project



✓ 60%

Methodology

60%



Methodology

Step 1 : Check if there is a person or not in the room

```
# Instantiate the Object Detector for person identification
detector=VideoObjectDetection()
# Use RetinaNet Model
detector.setModelTypeAsYOLOv3()
# Identify Person Only
custom_objects = detector.CustomObjects(person=True)
# Set the Path to the Model File
detector.setModelPath("models/yolo.h5")
detector.loadModel(detection_speed="fast")
```

Step 2 : Only if a person has been detected, carry out face detection

```
def face_detection(frame, frame_number):
    found=False
    faceCascade = cv2.CascadeClassifier(cascPath)

    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

    faces = faceCascade.detectMultiScale(
        gray,
        scaleFactor=1.1,
        minNeighbors=10,
        minSize=(30, 30),
        flags=cv2.CASCADE_SCALE_IMAGE
    )

    # Draw a rectangle around the faces
    for (x, y, w, h) in faces:
        cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 255, 0), 2)

    # Save the Image Locally if there are faces
    if len(faces) > 0:
        cv2.imwrite(output_path+"capture"+str(frame_number)+".png", frame)
        found=True

    return found
```

Methodology

Step 3 : If a face can be identified, then carry out face recognition. If a face cannot be identified but a person can be identified, print a warning.

Step 4 : If a face can be identified, then carry out face recognition. Then check if it passes facial recognition. If it passes print success, else warning.

```
def facial_recognition(image, frame_number):
    print("Facial Recognition Begins for Frame " + str(frame_number) + "!")

    #find path of xml file containing haarcascade file
    cascPathface = "haarcascade_frontalface_alt2.xml"
    # load the haarcascade in the cascade classifier
    faceCascade = cv2.CascadeClassifier(cascPathface)
    # load the known faces and embeddings saved in last file
    data = pickle.loads(open('face_enc', "rb").read())
    #Find path to the image you want to detect face and pass it here
    # image = cv2.imread("capture"+str(image_num)+".png")
    rgb = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
    #convert image to Grayscale for haarcascade
    gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
    faces = faceCascade.detectMultiScale(gray,
                                         scaleFactor=1.1,
                                         minNeighbors=5,
                                         minSize=(60, 60),
                                         flags=cv2.CASCADE_SCALE_IMAGE)

    # the facial embeddings for face in input
    encodings = face_recognition.face_encodings(rgb)
    names = []
    print(encodings)
# loop over the facial embeddings incase
# we have multiple embeddings for multiple fcaes
for encoding in encodings:
    #Compare encodings with encodings in data["encodings"]
    #Matches contain array with boolean values and True for the embeddings it matches closely
    #and False for rest
    matches = face_recognition.compare_faces(data["encodings"],
    encoding)
    #set name =inknown if no encoding matches
    name = "Unknown"
    # check to see if we have found a match
    if True in matches:
        print("Known Person")
        #Find positions at which we get True and store them
        matchedIdxs = [i for (i, b) in enumerate(matches) if b]
        counts = {}
        # loop over the matched indexes and maintain a count for
        # each recognized face face
        for i in matchedIdxs:
            #Check the names at respective indexes we stored in matchedIdxs
            name = data["names"][i]
            #increase count for the name we got
            counts[name] = counts.get(name, 0) + 1
            #set name which has highest count
            name = max(counts, key=counts.get)
            print("Person is! "+str(name))

        # update the list of names
        names.append(name)
        # loop over the recognized faces
        for ((x, y, w, h), name) in zip(faces, names):
            # rescale the face coordinates
            # draw the predicted face name on the image
            #cv2.rectangle(image, (x, y), (x + w, y + h), (0, 255, 0), 2)
            cv2.putText(image, name, (x, y), cv2.FONT_HERSHEY_SIMPLEX,
            0.75, (0, 255, 0), 2)
            cv2.imwrite(output_path + "not_sus.png", image)
    else:
        print("Face Not Recognized")
        cv2.imwrite(output_path+"sus.png", image)
        print("Unauthorized Person!")
```

Test Results


```
person  :  99.24293160438538
Face Not Detected. Person's Face Suspiciously Covered!
person  :  99.09639954566956
Face Not Detected. Person's Face Suspiciously Covered!
person  :  98.80953431129456
Face Detected in Frame: 200
Facial Recognition Begins for Frame 200!
[]
person  :  98.81949424743652
Face Not Detected. Person's Face Suspiciously Covered!
person  :  98.83725643157959
Face Not Detected. Person's Face Suspiciously Covered!
```

Suspicious


```
Known Person
Person is: Odhara
Person is: Odhara
Person is: Odhara
Person is: Odhara
Person is: Odhara
Person is: Odhara
Person is: Odhara
Person is: Odhara
Person is: Odhara
person  :  99.4857132434845
```

Not Suspicious

References

[1] Y. Tian and S. Ma, “Kidnapping Detection and Recognition in Previous Unknown Environment,” *J. Sensors*, vol. 2017, 2017, doi: 10.1155/2017/6468427.

[2] Y. Mori *et al.*, “A self-configurable new generation children tracking system based on mobile ad hoc networks consisting of android mobile terminals,” *Proc. - 2011 10th Int. Symp. Auton. Decentralized Syst. ISADS 2011*, pp. 339–342, 2011, doi: 10.1109/ISADS.2011.51.

[3] S. C, “Guardian Uses Bluetooth Low Energy Tech to Keep Child Safe.”

[4] A. Miyahara and I. Nagayama, “An intelligent security camera system for kidnapping detection,” *J. Adv. Comput. Intell. Intell. Informatics*, vol. 17, no. 5, pp. 746–752, 2013, doi: 10.20965/jaciii.2013.p0746.

[5] J. H. Park, K. Song, and Y. Kim, “A Kidnapping Detection Using Human Pose Estimation in Intelligent Video Surveillance Systems,” *J. Korea Soc. Comput. Inf.*, vol. 23, no. 8, pp. 9–16, 2018, doi: 10.9708/jksci.2018.23.08.009.

[6] R. H. Gwon, K. Y. Kim, J. T. Park, H. Kim, and Y. S. Kim, “A kidnapping detection scheme using frame-based classification for intelligent video surveillance,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 8170 LNAI, pp. 345–354, 2013, doi: 10.1007/978-3-642-41218-9_37.

[7] A. Sofranova, “12 Signs that can help you Recognize a child kidnapper,” *Bright Side*.

IT18006308 | PRAMODYA HETTIARACHCHI

Data Science

Background

- 37% of deaths in children are between the age of one and four using hazardous objects.
- These accidents are avoidable and most of them happened in a chain of events.
- Kids do not have the sense to distinguish hot from cold or sharp from dull.
- Most accidents are happened due to the curiosity of the children and lack of supervision
- Every third household has a child burn victim.

Research Question

- How to Identify the hazardous object is sharp or hot liquid container object
- How to determine the danger level of hazardous object
- How to check weather the child is in proximity to the hazardous object or not
- How to mitigate impact of accidents by ensuring the AI Assistance is notified as fast as possible
- How to trigger warnings to the child to distract or give alerts to parent to prevent or minimize the danger.

Research Gap

Product	Research A [2] [6] [7]	Research B [3] [4]	Research C [5]	Research D [8] [9]	Proposed solution (AI Care)
Child Surveillance System	✓	✗	✗	✗	✓
Real Time Object Detection	✗	✗	✓	✓	✓
Real Time Sharp Object Detection	✗	✗	✗	✓	✓
Classify the danger level	✗	✗	✗	✗	✓
Child detection	✓	✗	✗	✗	✓
Identify child is in proximity to hazardous object	✗	✗	✗	✗	✓
Prevent accidents before happening	✗	✗	✗	✗	✓

Objectives

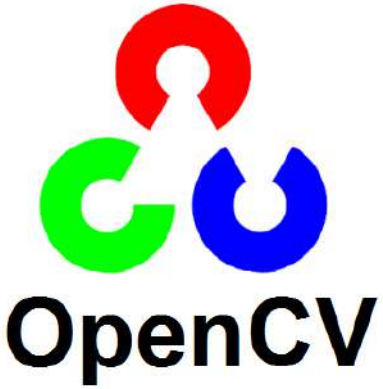
- Main Objective

  - The main objective is to ensure that child is in the early development stage are safe from accidents using hazardous objects by using surveillance cameras integrated with computer vision.

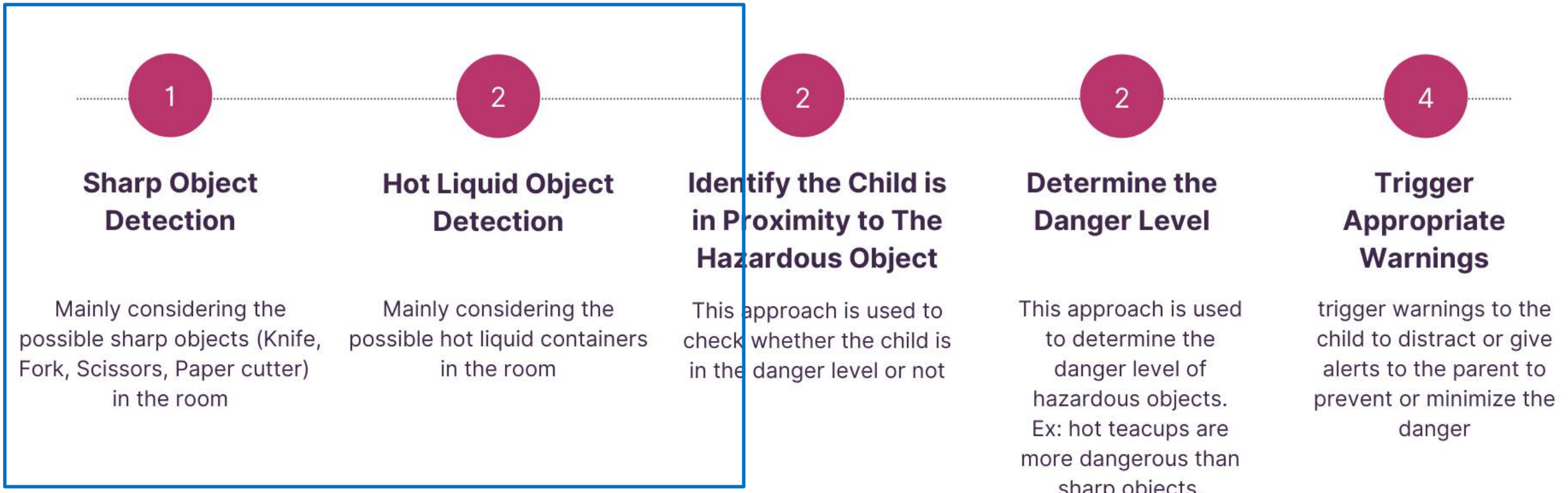
- Specific Objectives

  - Detect possible sharp objects COMPLETED
  - Detect possible hot liquid containers COMPLETED
  - Detect child in proximity to hazardous object (40% Complete)
  - Identify the danger level of object
  - Trigger warnings to child or give alert to parent as suitable to the situation

Technologies

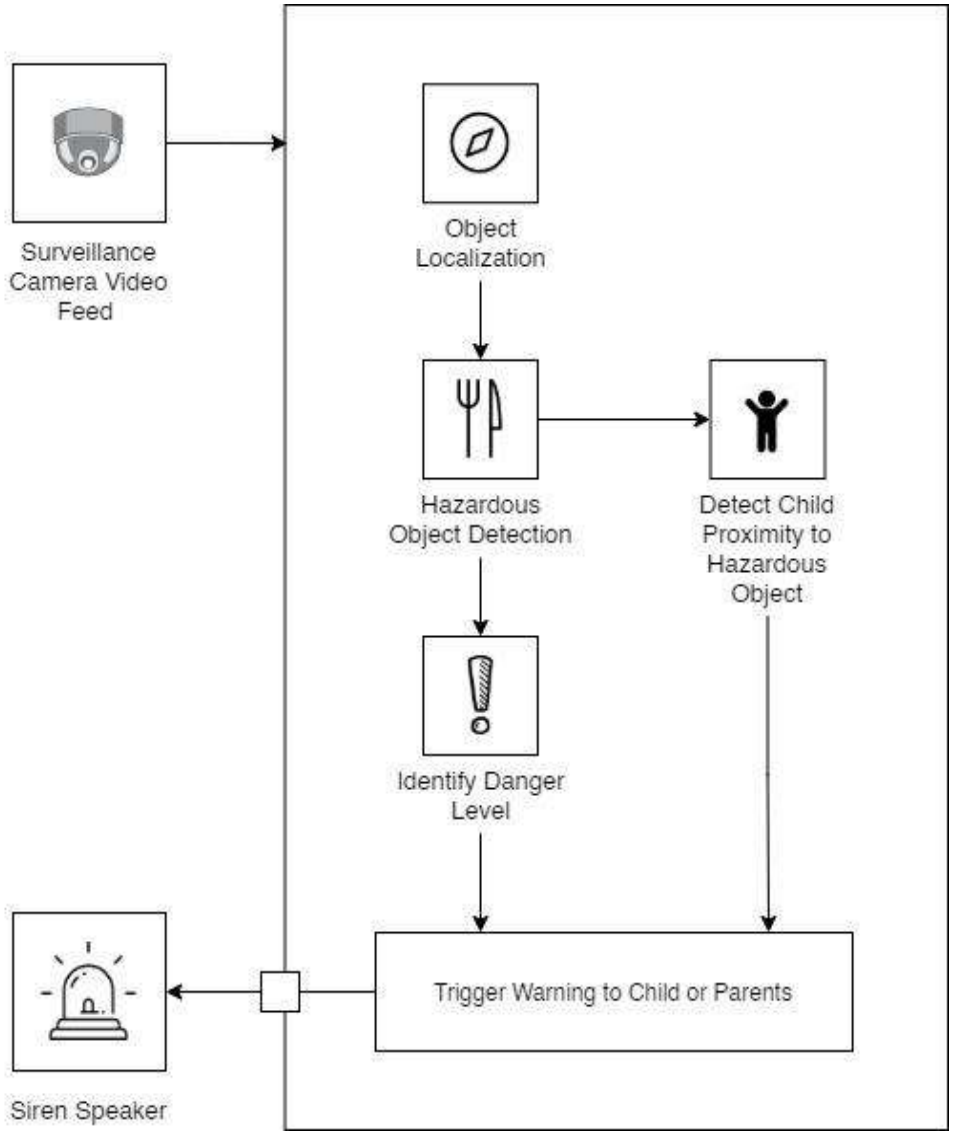


Completion of the project



60%

Methodology



Methodology

Detect Possible Sharp Objects and Hot Liquid Objects

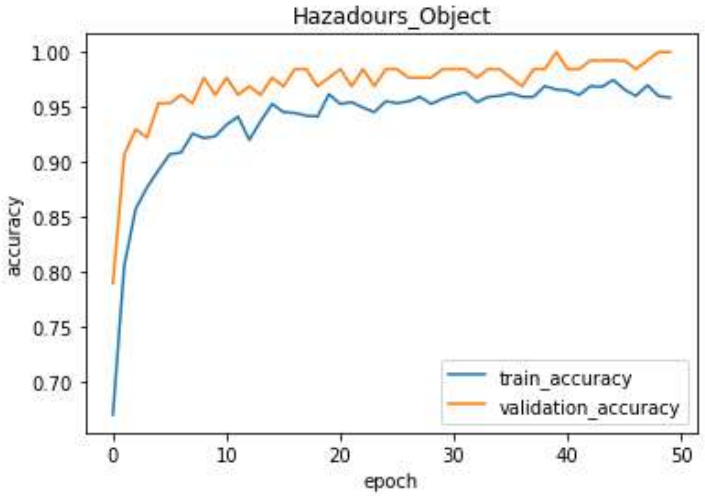
- **Train the models using RetinaNet**

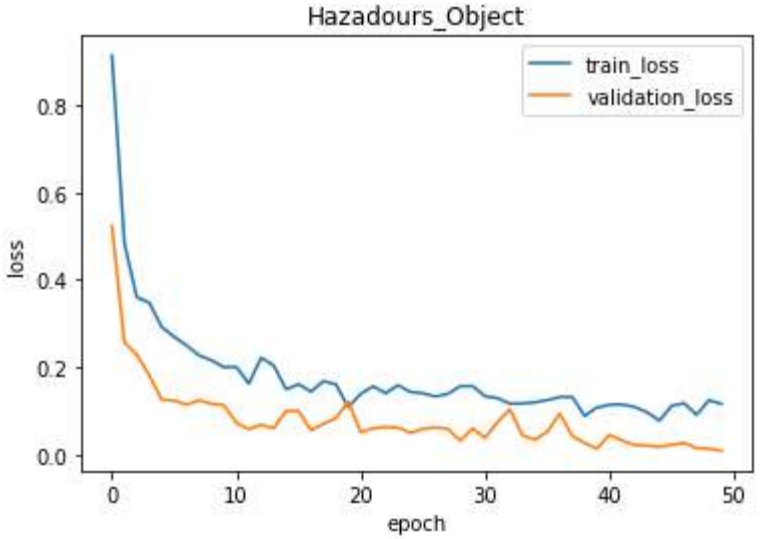
- **Sharp and Hot Liquid Object Detection**

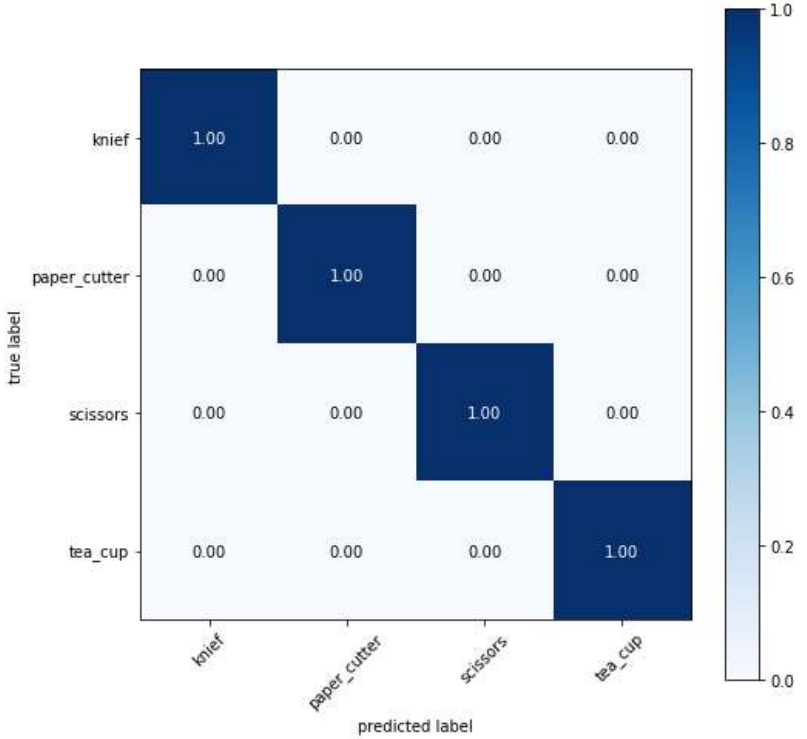
Model Training	Trained the model to detect hazardous objects
No. of Classes 4 classes	Knife, Paper Cutter, Scissor, Teacup
Model & Architecture	RetinaNet
Training set	400
Testing set	40
100 epochs	50
Accuracy	0.95

Completion of Project

✓ Test Results of sharp and hot liquid object detection (screenshots) - ***Completed***

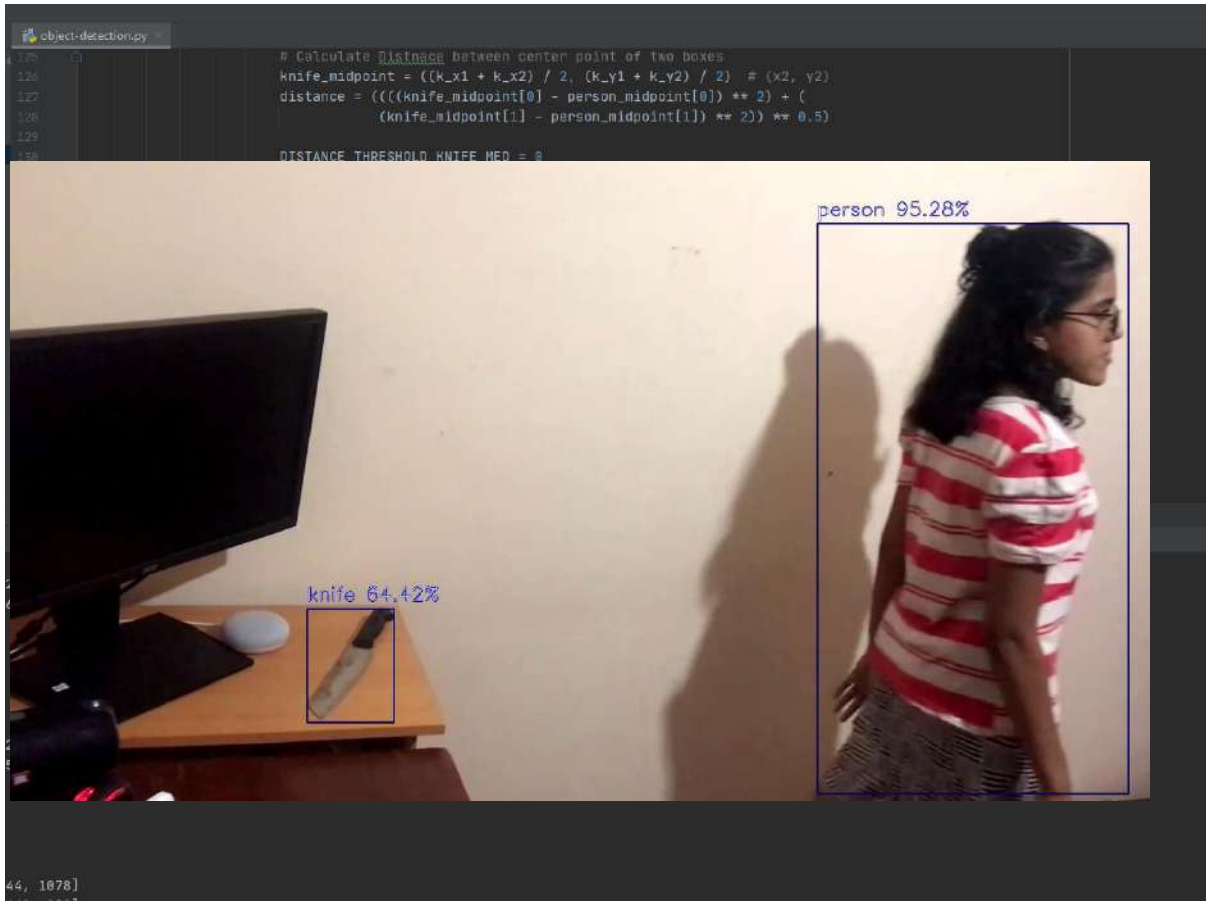


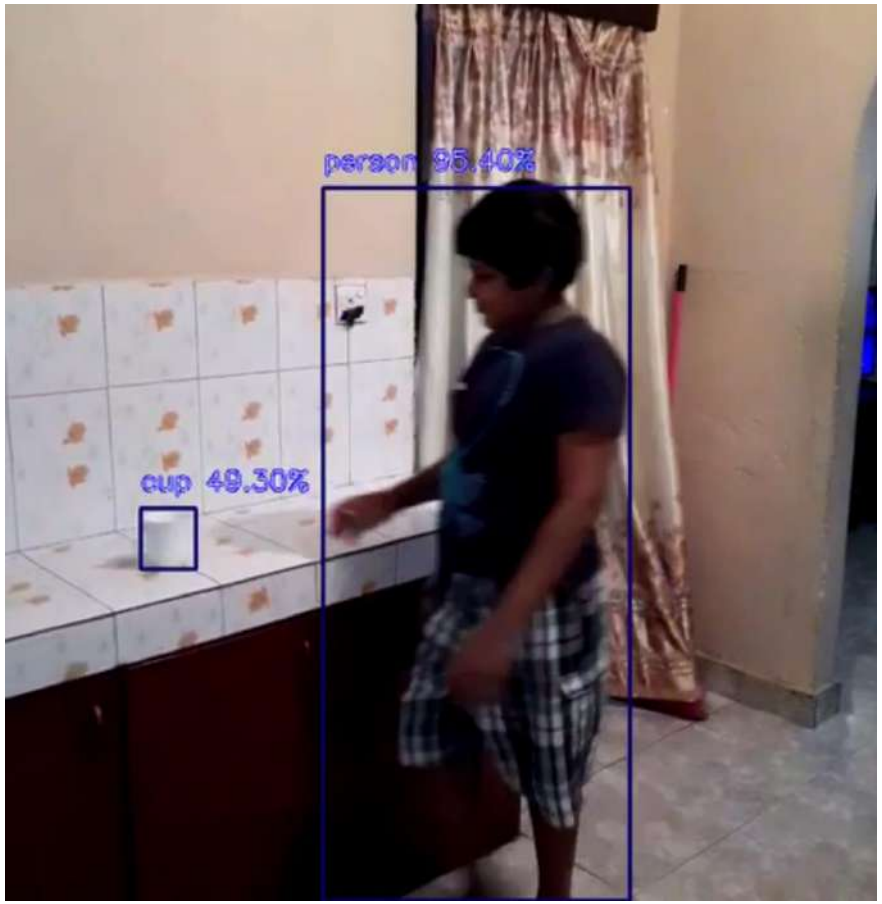




Completion of Project

✓ Test Results of sharp and hot liquid object detection (screenshots) - ***Completed***

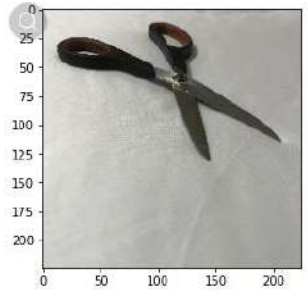




Completion of Project

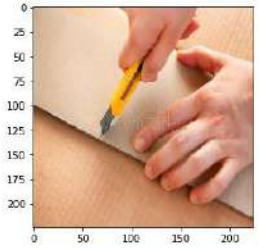
✓ Test Results of sharp and hot liquid object detection (screenshots) - ***Completed***

```
In [106]: img.show
# plt.subplots(figsize = (15,8))
plt.imshow(img)
for i in range(0,1):
    print('Predicted Class # : ',np.argmax(prediction[i]))
    print('Predicted Class Name :' ,category[np.argmax(prediction[i])])

Predicted Class # :  2
Predicted Class Name : scissors
```



```
In [118]: img.show
# plt.subplots(figsize = (15,8))
plt.imshow(img)
for i in range(0,1):
    print('Predicted Class # : ',np.argmax(prediction[i]))
    print('Predicted Class Name :' ,category[np.argmax(prediction[i])])

Predicted Class # :  0
Predicted Class Name : knief
```

```
In [98]: img.show
# plt.subplots(figsize = (15,8))
plt.imshow(img)
for i in range(0,1):
    print('Predicted Class # : ',np.argmax(prediction[i]))
    print('Predicted Class Name :' ,category[np.argmax(prediction[i])])

Predicted Class # :  1
Predicted Class Name : paper_cutter
```



```
In [112]: img.show
# plt.subplots(figsize = (15,8))
plt.imshow(img)
for i in range(0,1):
    print('Predicted Class # : ',np.argmax(prediction[i]))
    print('Predicted Class Name :' ,category[np.argmax(prediction[i])])

Predicted Class # :  3
Predicted Class Name : tea_cup
```

Methodology

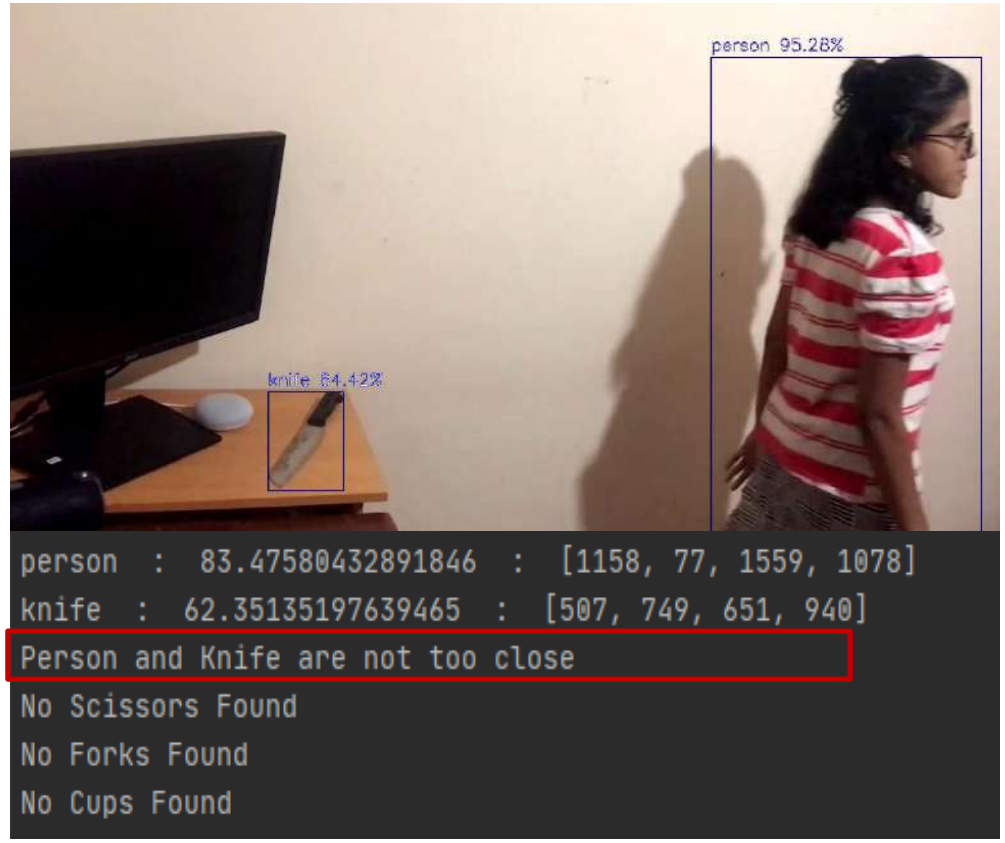
(40% Complete)

Detect child in proximity to hazardous object

- Check if Child is in 100 Pixels to Left and 100 Pixels to Right of the Identified Object(s)
- Check if Knife Overlaps with the Person's Safety Boundary
- If the Bounding Boxes Overlap at some point, calculate distance between center point of two boxes using Open CV.

Completion of Project

✓ Test Results of Detect child in proximity to hazardous object (screenshots) - ***Completed***



IT17167710 | PRATHAPA D.M.J

DATA SCIENCE

Research Questions

- ✓ How to prevent the child from stepping out of the safe zone of the room
- ✓ How to determine the possible furniture that child can climb on
- ✓ How to prevent the child from climbing on furniture
- ✓ How to detect the action of falling
- ✓ How to mitigate or diminish impact of m injuries by ensuring the caregiver is notified as fast as possible

Objectives

Specific Objective

✓Detect Child leaving the safe zone boundary of the room and Child climbing to a dangerous position and fall detection and taking prompt responsive actions to avoid danger.

Sub Objectives

✓calculate the mid-point of the captured child (This will be used for other sub objectives as well) COMPLETED

✓Capture safety zone boundary breach COMPLETED

✓Capture spot unsafe heights from the current position (Detecting the action of Climbing) - 70% - completed

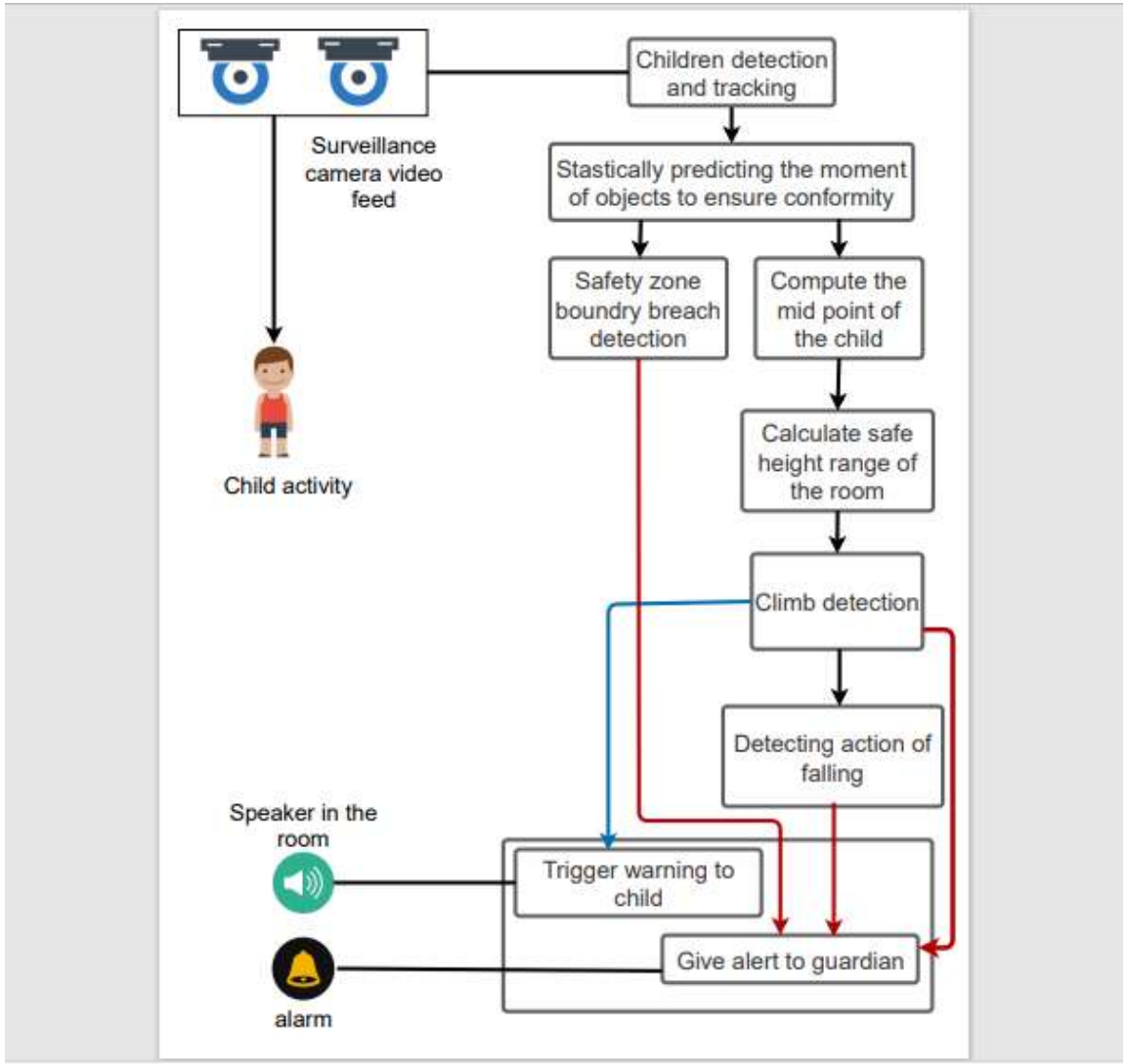
✓Fall Detection (Detecting the action of falling) - 20% - completed

✓Alerting (communicate the incident efficiently to the parent or caregiver.) – 10% completed

Comparison between existing Research and Products

Product	[1] [2]	[3] [4] [5]	[6] [7]	[8] [9] [10]	**Proposed solution (AI Care)**
Fall detection algorithm	✓	✓	✓	✓	✓
Climb detection algorithm	✓	✗	✗	✗	✓
Taking prompt responsive actions to avoid danger (alerts)	✓	✓	✓	✗	✓
Child detection	✓	✗	✗	✗	✓
Capture safety zone boundary breach	✗	✗	✗	✗	✓
Discriminate falls from a child sitting down abruptly	✗	✗	✗	✗	✓
Differentiate between a real fall and an incident in which a person is lying	✗	✗	✓	✓	✓
Calculate child midpoint & Calculate the average safe height of the room.	✗	✗	✗	✗	✓
Height and hazard segmentation	✗	✗	✗	✗	✓

High-level System Diagram



✓100% - completed

Sub Objectives 1 & 2

calculate the mid-point

Capture safety zone boundary breach

Methodology

Capture safety zone boundary breach

- **Calculate the mid-point:**

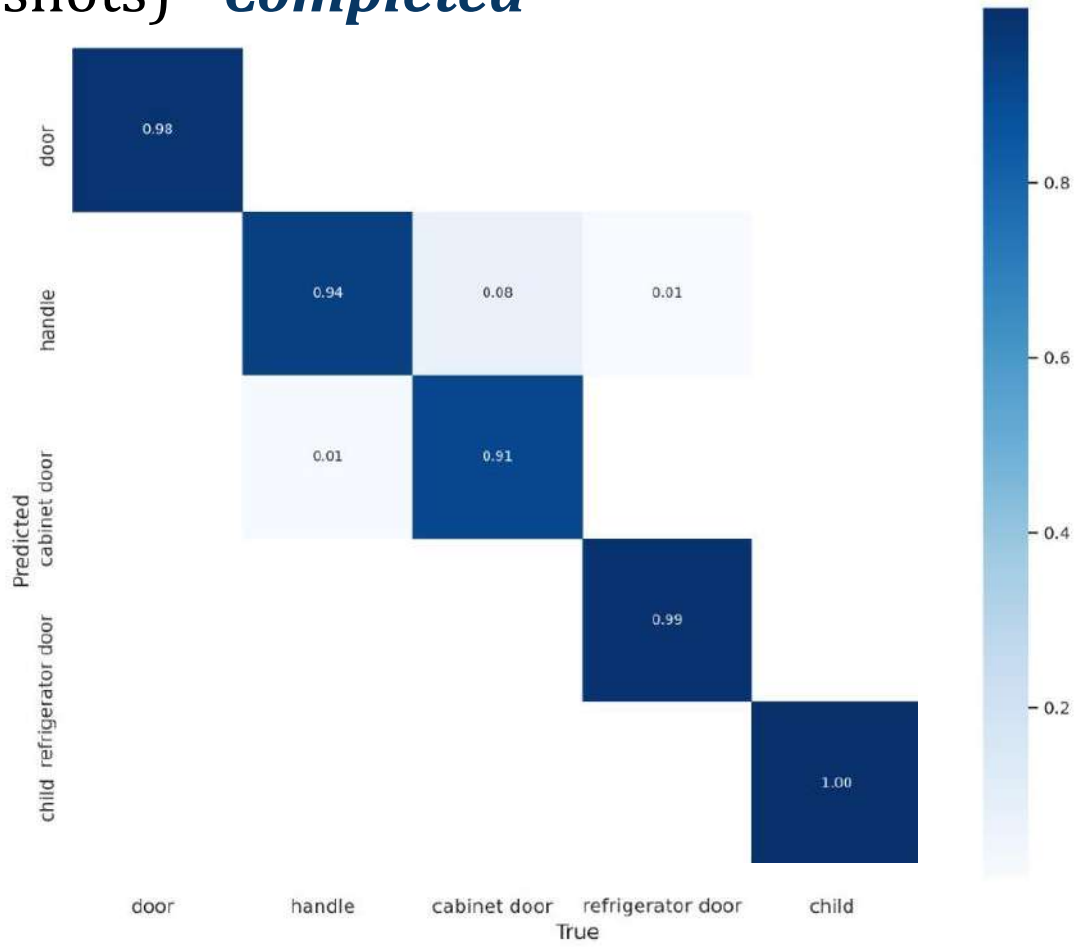
  computed the midpoint of child and Door of the room using Open CV.

- **Door classification**

Model Training	Trained it to classify Door images under 4 door types
No. of Classes 5 classes	Door , cabinet door , handle door , refrigerator door including Child
Model & Architecture	Used Yolo5 and Resnet 101
Training set	2800
Testing set	600
epochs	100
Accuracy	0.93

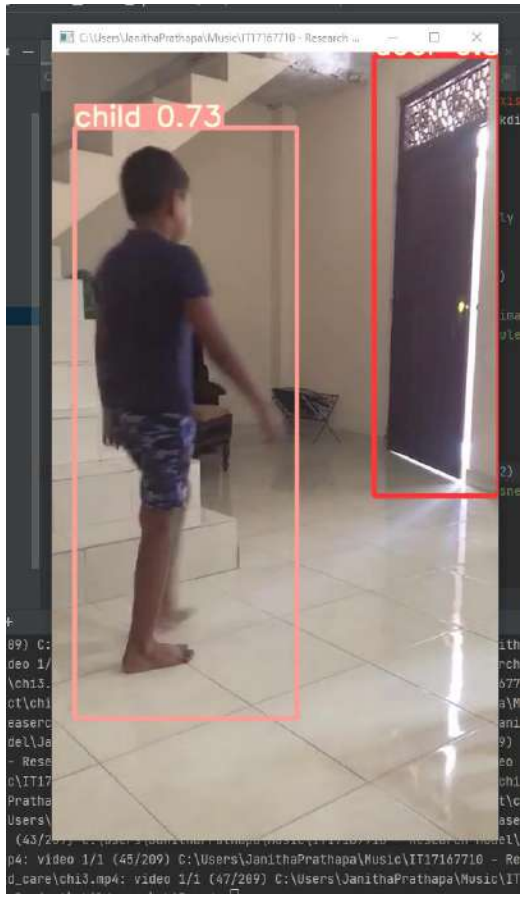
Completion of the project

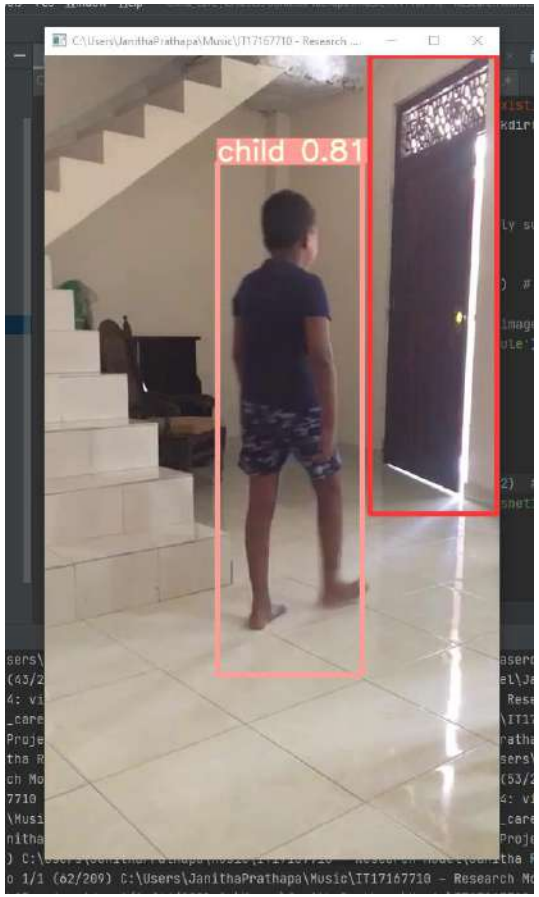
✓ Test Results (screenshots) - ***Completed***

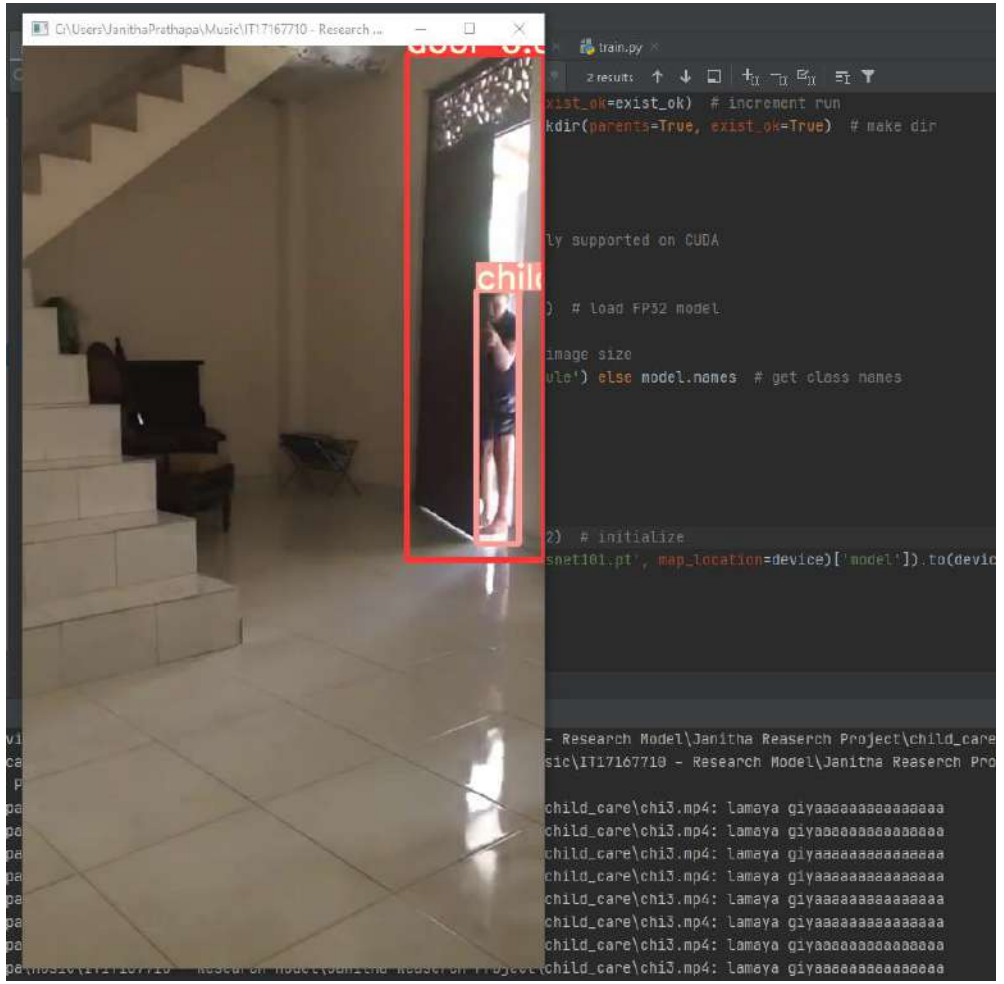


Completion of the project

✓ Detect safety zone boundary breach (screenshots) - ***Completed***

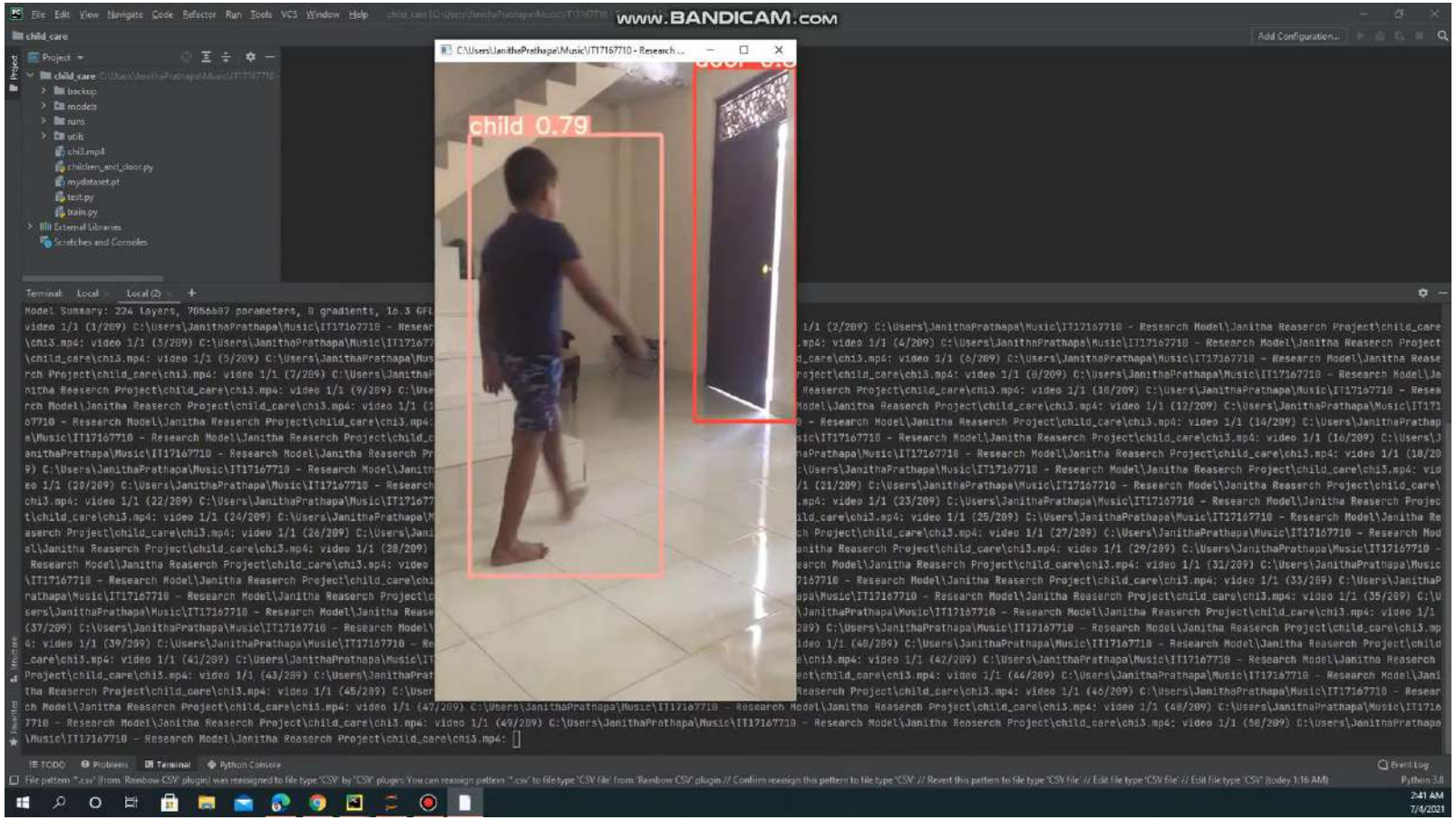






Completion of the project

✓ Detect safety zone boundary breach (Demo) - ***Completed***



Technologies

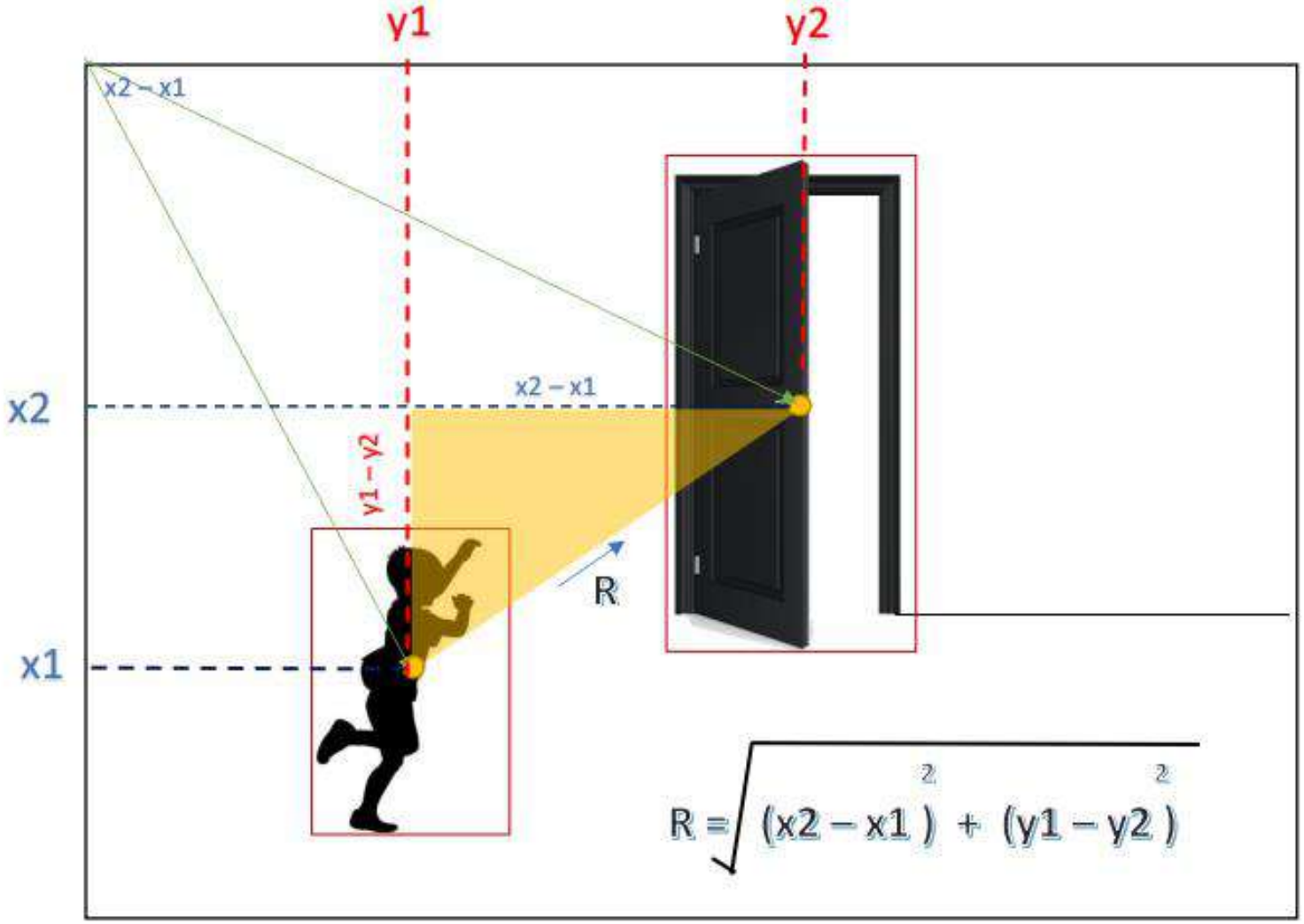
OpenCV
TensorFlow
Yolo5
Python winsound

Techniques

Pythagoras theorem

Completion of the project

- ✓ Pythagoras theorem



✓70% - completed

Sub Objectives 3

Climb Detection

Methodology

- Capture spot unsafe heights from the current position
- Calculate the mid-point:
  - computed the midpoint of child using computer vision.

- Furniture classification

Model Training	Trained it to classify furniture images under 5 door types
No. of Classes 5 classes	Bed , Chair, Sofa, Table, Swivel-chair
Model & Architecture	**VGG16**
Training set	4100 images
Testing set	430 images
epochs	100
Accuracy	0.95

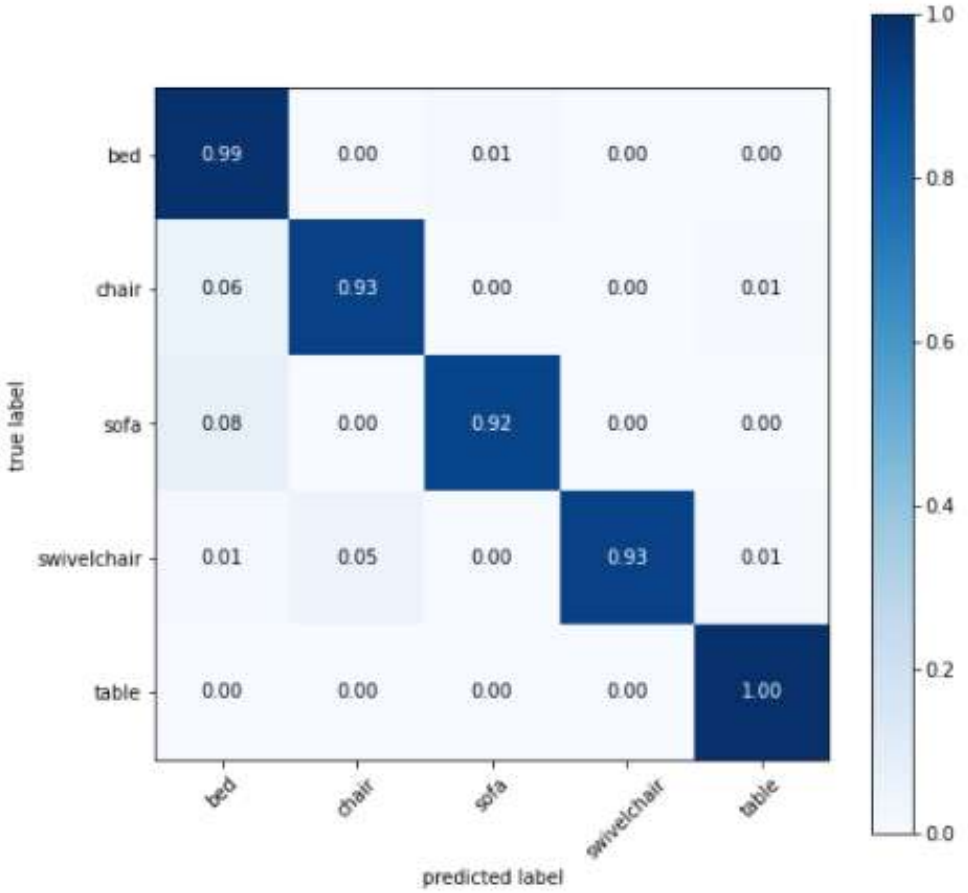
Completion of the project

✓ Test Results of furniture classification (screenshots) - ***Completed***

```
Epoch 93/100
256/256 [==============================] - 1104s 4s/step - loss: 0.2591 - acc: 0.9256 - val_loss: 0.3560 -
Epoch 94/100
256/256 [==============================] - 1101s 4s/step - loss: 0.2376 - acc: 0.9363 - val_loss: 0.2941 -
Epoch 95/100
256/256 [==============================] - 1105s 4s/step - loss: 0.2355 - acc: 0.9346 - val_loss: 0.2262 -
Epoch 96/100
256/256 [==============================] - 1103s 4s/step - loss: 0.2582 - acc: 0.9393 - val_loss: 0.2324 -
Epoch 97/100
256/256 [==============================] - 1105s 4s/step - loss: 0.2504 - acc: 0.9322 - val_loss: 0.2815 -
Epoch 98/100
256/256 [==============================] - 1101s 4s/step - loss: 0.2362 - acc: 0.9412 - val_loss: 0.2390 -
Epoch 99/100
256/256 [==============================] - 1105s 4s/step - loss: 0.2437 - acc: 0.9334 - val_loss: 0.2820 -
Epoch 100/100
256/256 [==============================] - 1109s 4s/step - loss: 0.2564 - acc: 0.9346 - val_loss: 0.2454 -

---------------------------
Total time:  151821.5455789566 seconds
Wall time: 1d 18h 10min 21s
```

```
In [24]: print('Classification Report')
         print(classification_report(validation_generator.classes, y_pred, target_names=category))
```

```
Classification Report
              precision    recall  f1-score   support

         bed       0.87      0.99      0.93       100
       chair       0.95      0.93      0.94       100
        sofa       0.99      0.92      0.95       100
  swivelchair       1.00      0.93      0.96       100
       table       0.94      1.00      0.97        30

    accuracy                           0.95       430
   macro avg       0.95      0.95      0.95       430
weighted avg       0.95      0.95      0.95       430
```



Completion of the project

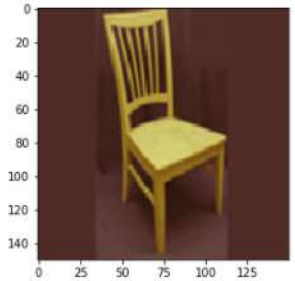
- ✓ Test Results of furniture classification (screenshots) - ***Completed***

```
img = image.load_img('E:/Data Reasech all/furniture-images/img/val/sofa/00000211
img_array = image.img_to_array(img)
img_batch = np.expand_dims(img_array, axis=0)
img_preprocessed = preprocess_input(img_batch)
prediction = new_model.predict(img_preprocessed)
img.show
plt.imshow(img)
for i in range(0,1):
    print('Predicted Class # : ',np.argmax(prediction[i]))
    print('Predicted Class Name :' ,category[np.argmax(prediction[i])])
```

```
Predicted Class # :  2
Predicted Class Name : couch
```

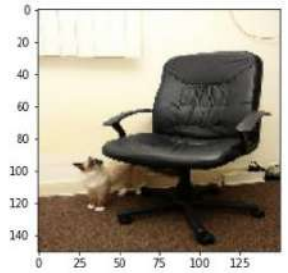
```
img = image.load_img('E:/Data Reasech all/furniture-images/img/val/chair/0000
img_array = image.img_to_array(img)
img_batch = np.expand_dims(img_array, axis=0)
img_preprocessed = preprocess_input(img_batch)
prediction = new_model.predict(img_preprocessed)
img.show
plt.imshow(img)
for i in range(0,1):
    print('Predicted Class # : ',np.argmax(prediction[i]))
    print('Predicted Class Name :' ,category[np.argmax(prediction[i])])
```

```
Predicted Class # :  1
Predicted Class Name : chair
```



```
img = image.load_img('E:/Data Reasech all/furniture-images/img/val/swivelchair/0000
img_array = image.img_to_array(img)
img_batch = np.expand_dims(img_array, axis=0)
img_preprocessed = preprocess_input(img_batch)
prediction = new_model.predict(img_preprocessed)
img.show
plt.imshow(img)
for i in range(0,1):
    print('Predicted Class # : ',np.argmax(prediction[i]))
    print('Predicted Class Name :' ,category[np.argmax(prediction[i])])
```

```
Predicted Class # :  3
Predicted Class Name : swivelchair
```



```
plt.imshow(img)
for i in range(0,1):
    print('Predicted Class # : ',np.argmax(prediction[i]))
    print('Predicted Class Name :' ,category[np.argmax(prediction[i])])
```

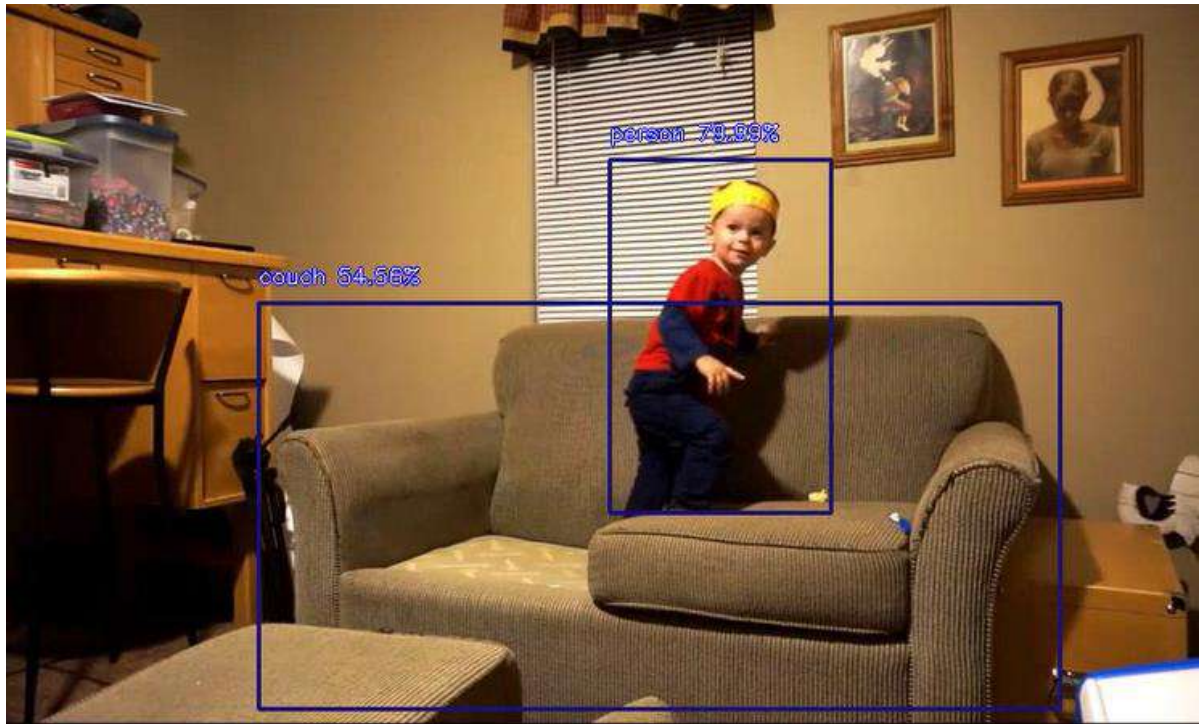
```
Predicted Class # :  4
Predicted Class Name : table
```

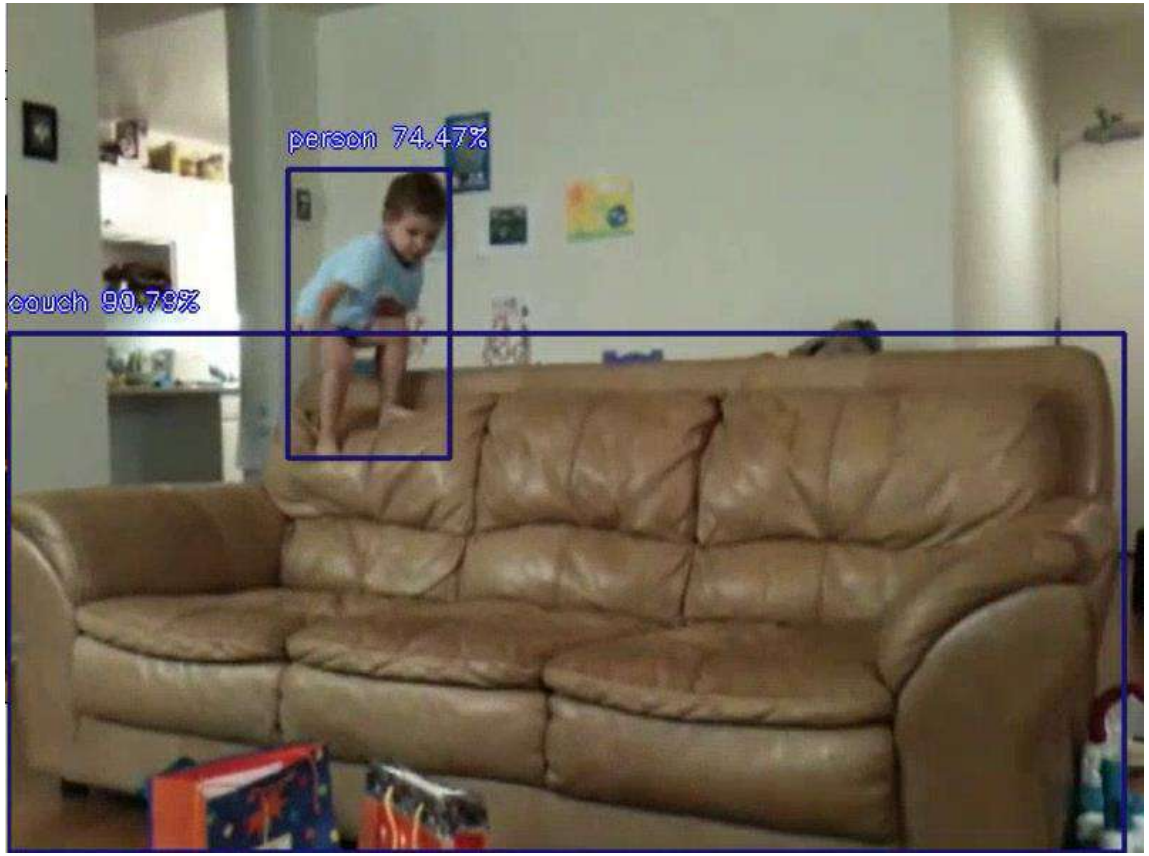
```
for i in range(0,1):
    print('Predicted Class # : ',np.argmax(prediction[i]))
    print('Predicted Class Name :' ,category[np.argmax(prediction[i])])
```

```
Predicted Class # :  0
Predicted Class Name : bed
```

Completion of the project

✓ Climb Detection (screenshots)





✓20% - completed

Sub Objectives 4

Fall Detection

Fall Detection

IN PROGRESS

image

Import from file | Save as... | Copy to clipboard

grayscaled image

Chain with... | Save as... | Copy to clipboard

image

Import from file | Save as... | Copy to clipboard

grayscaled image

Chain with... | Save as... | Copy to clipboard

Methodology

- Capture spot unsafe heights from the current position

Pose classification

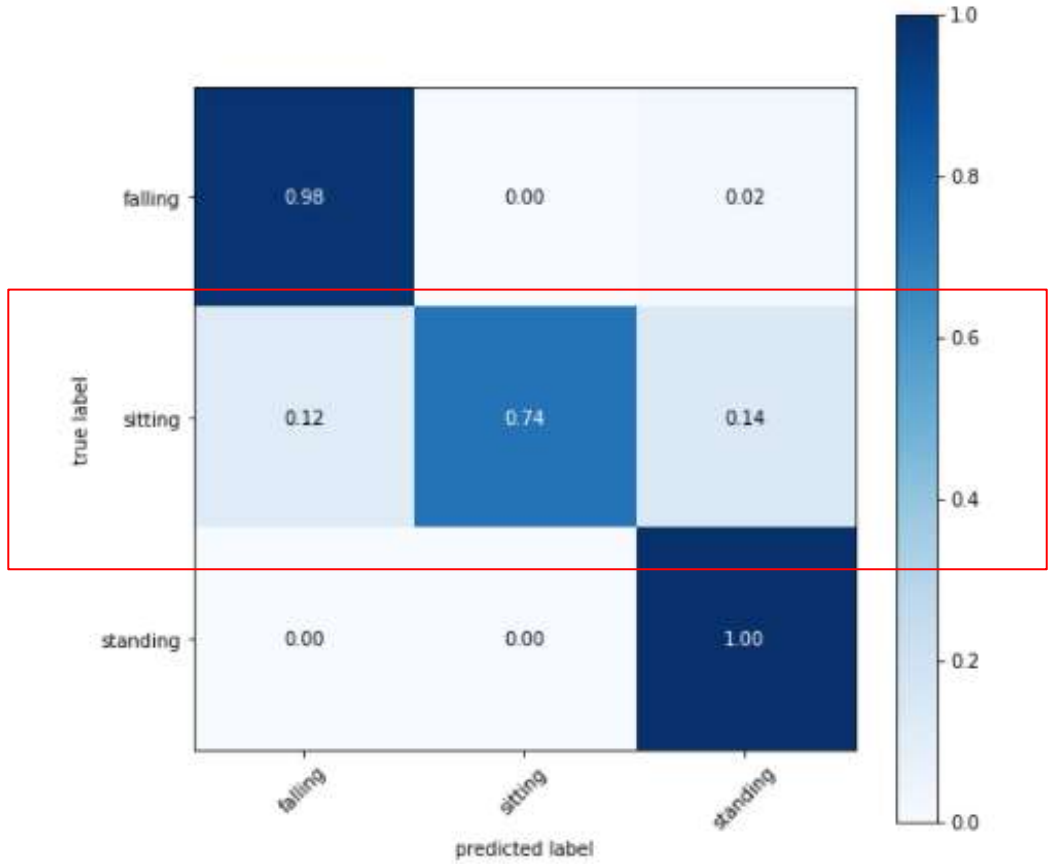
Model Training	**Fall identification using pose detection**
No. of Classes 3 classes	Falling , Sitting , Standing
Model & Architecture	Sequential CNN Classification.
Training set	1100 images
Testing set	150 images
epochs	50
Accuracy	0.91

Completion of the project

IN PROGRESS

- ✓ Test Results of Pose classification (screenshots)

```
In [174]: print('Confusion Matrix')
          print(confusion_matrix(validation_generator.classes, y_pred))

          Confusion Matrix
          [[49  0  1]
           [ 6 37  7]
           [ 0  0 50]]

In [175]: print('Classification Report')
          print(classification_report(validation_generator.classes, y_pred, target_names=category))

          Classification Report
                        precision    recall  f1-score   support

               falling       0.89      0.98      0.93        50
               sitting       1.00      0.74      0.85        50
              standing       0.86      1.00      0.93        50

              accuracy                           0.91       150
             macro avg       0.92      0.91      0.90       150
          weighted avg       0.92      0.91      0.90       150

In [176]: from sklearn.metrics import classification_report
          from mlxtend.plotting import plot_confusion_matrix
          from sklearn import metrics
          from sklearn.metrics import confusion_matrix

In [177]: cm = metrics.confusion_matrix(validation_generator.classes, y_pred)
          figure, ax = plot_confusion_matrix(conf_mat = cm,
                                             show_absolute = False,
                                             show_normed = True,
                                             colorbar = True,
                                             class_names=category,
                                             figsize=(8, 8))
```



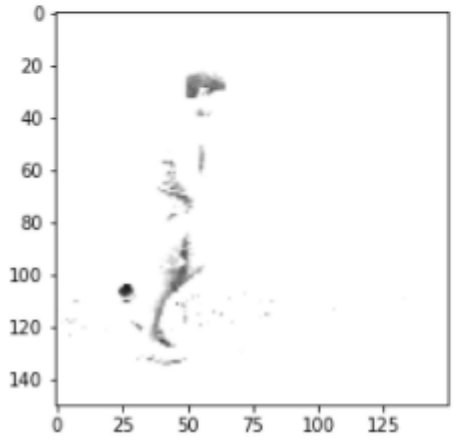
Completion of the project

IN PROGRESS

- ✓ Test Results of Pose classification (screenshots)

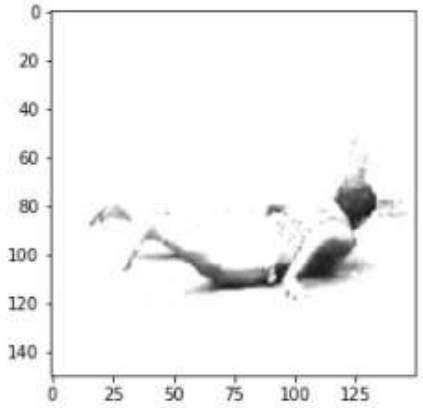
```
img = image.load_img('C:/Users/JanithaPrathapa/Music/IT17167710 - Research M
img_array = image.img_to_array(img)
img_batch = np.expand_dims(img_array, axis=0)
img_preprocessed = preprocess_input(img_batch)
prediction = new_model.predict(img_preprocessed)
img.show
plt.imshow(img)
for i in range(0,1):
    print('Predicted Class # : ',np.argmax(prediction[i]))
    print('Predicted Class Name :' ,category[np.argmax(prediction[i])])
```

```
Predicted Class # :  2
Predicted Class Name : standing
```




```
img = image.load_img('C:/Users/JanithaPrathapa/Music/IT17167710 - Research Mode
img_array = image.img_to_array(img)
img_batch = np.expand_dims(img_array, axis=0)
img_preprocessed = preprocess_input(img_batch)
prediction = new_model.predict(img_preprocessed)
img.show
plt.imshow(img)
for i in range(0,1):
    print('Predicted Class # : ',np.argmax(prediction[i]))
    print('Predicted Class Name :' ,category[np.argmax(prediction[i])])
```

```
Predicted Class # :  0
Predicted Class Name : falling
```



To do....

- ✓ Have to complete the rest of the climb detection part
- ✓ For the fall detection I'm planning to come up with combination approach of Pose classification and pose estimation approach.
- ✓ Alert Management

IT18255720 | JAYAKODY J.M.A.M.S

Data Science

Research Question

- The most common cause of home injuries is a general disregard for personal protection and a lack of oversight.
- Every year, an average of children getting injured because of unsafe electrical appliances sustained at home.
- How to prevent the child from electrical items.?
- How to detect child over adult.?
- How to implement Electrical injuries preventing assistance system?

Objectives

- **What is the specific objective ?**
  - Child detection.
  - Harmful electrical devices detection.
  - Alerting

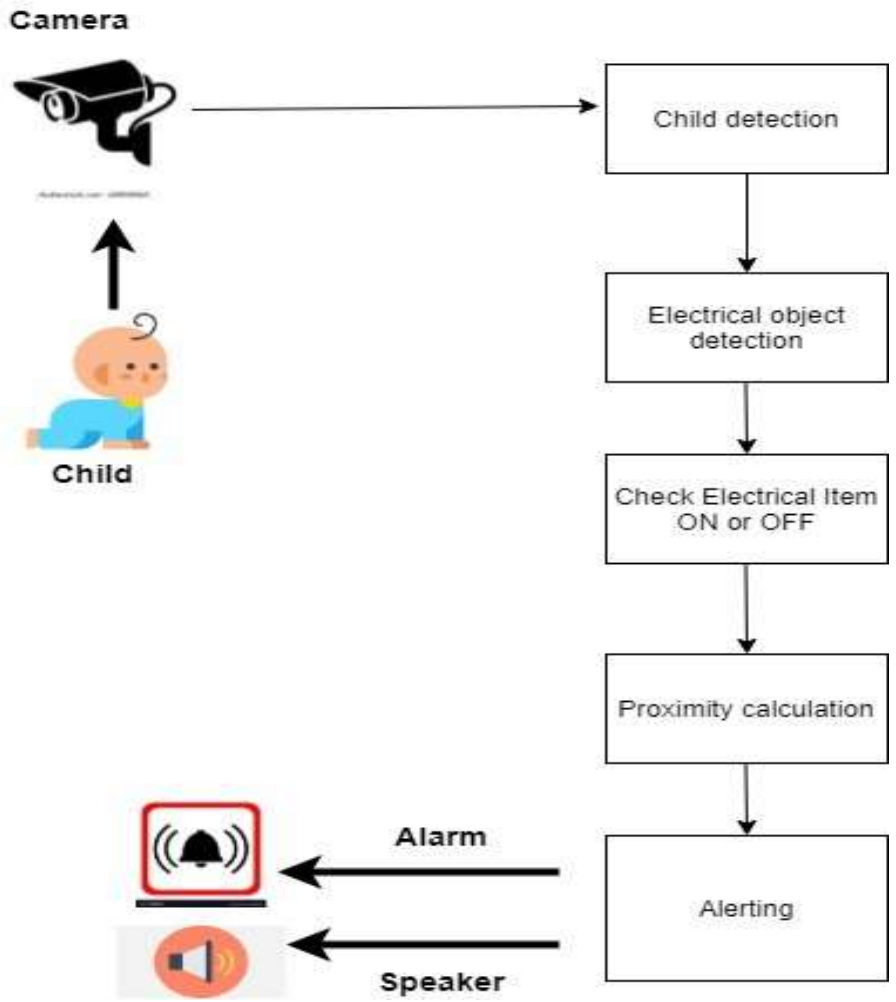
Objectives

- **What are the sub objectives ?**
  - ✓ Object Detection(Electrical appliances)
  - ✓ Person detection
  - ✓ Model creation and optimization
  - ✓ Proximity Calculation
  - ✓ Testing

Research Gap

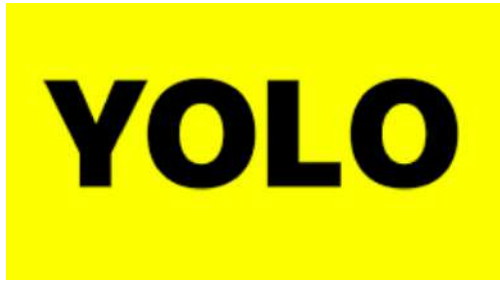
Research Projects	[7]	[5]	[4]	[6]	[8]	Proposed Solution
Real time child detection (Geo Metric feature based)	Available	Not available	Available	Not available	Available	Available
Real Time child detection (Head and body ratio)	Not available	Available	Not available	Not available	Not available	Available
Real time electrical object detection	Not available	Not available	Not available	Not available	Not available	Available
Computer vision-based child specific real time alerting system	Not available	Not available	Not available	Not available	Not available	Available
Child surveillance system	Available	Available	Available	Available	Available	Available
Real Time object detection	Not available	Not available	Not available	Not available	Available	Available

High-level System Diagram

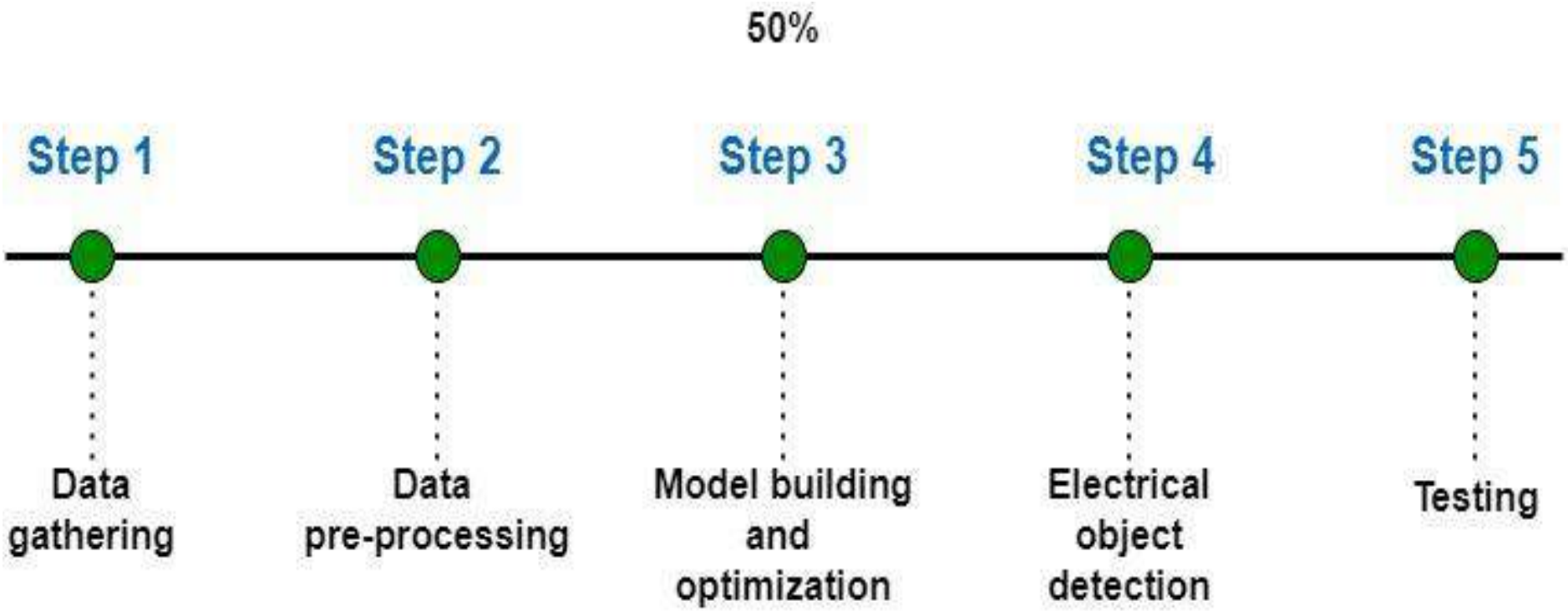


Tools and Technologies

- OpenCV, Keras, NumPy, TensorFlow, Python, ImageAI and LabelImg

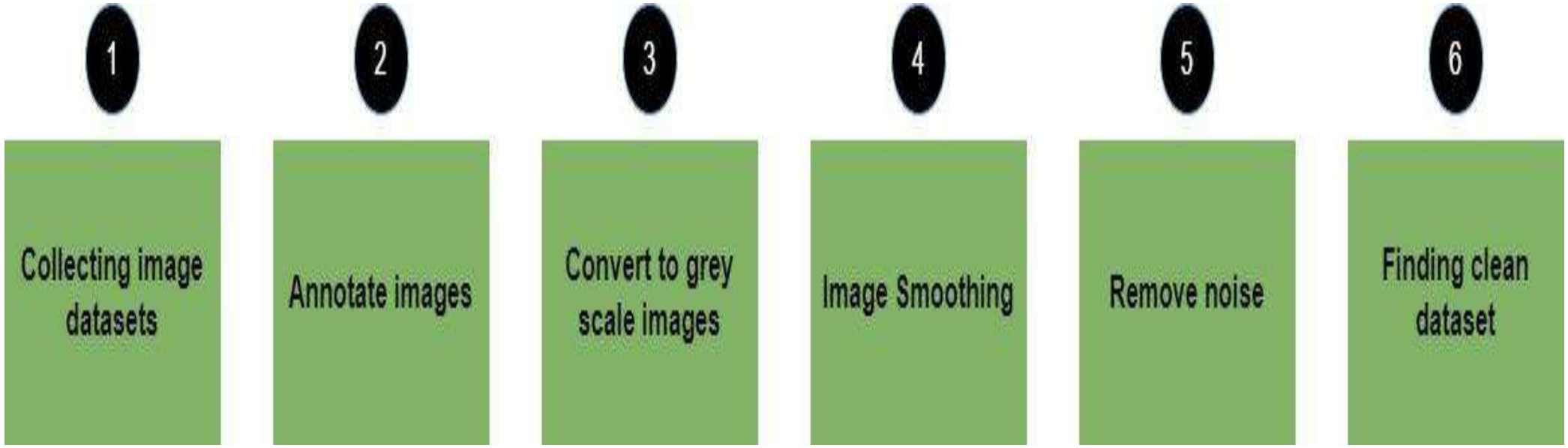


Completion of the project



Methodology

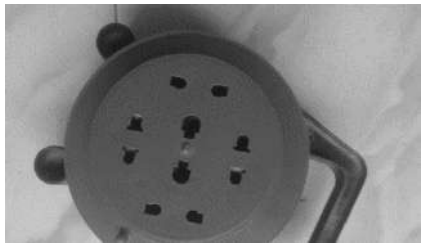
1. Data gathering And Pre-processing

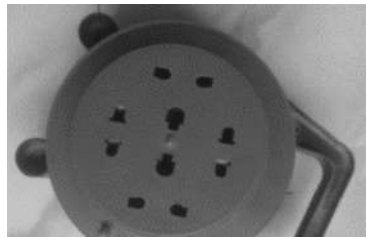


Methodology

- Simulating the dataset

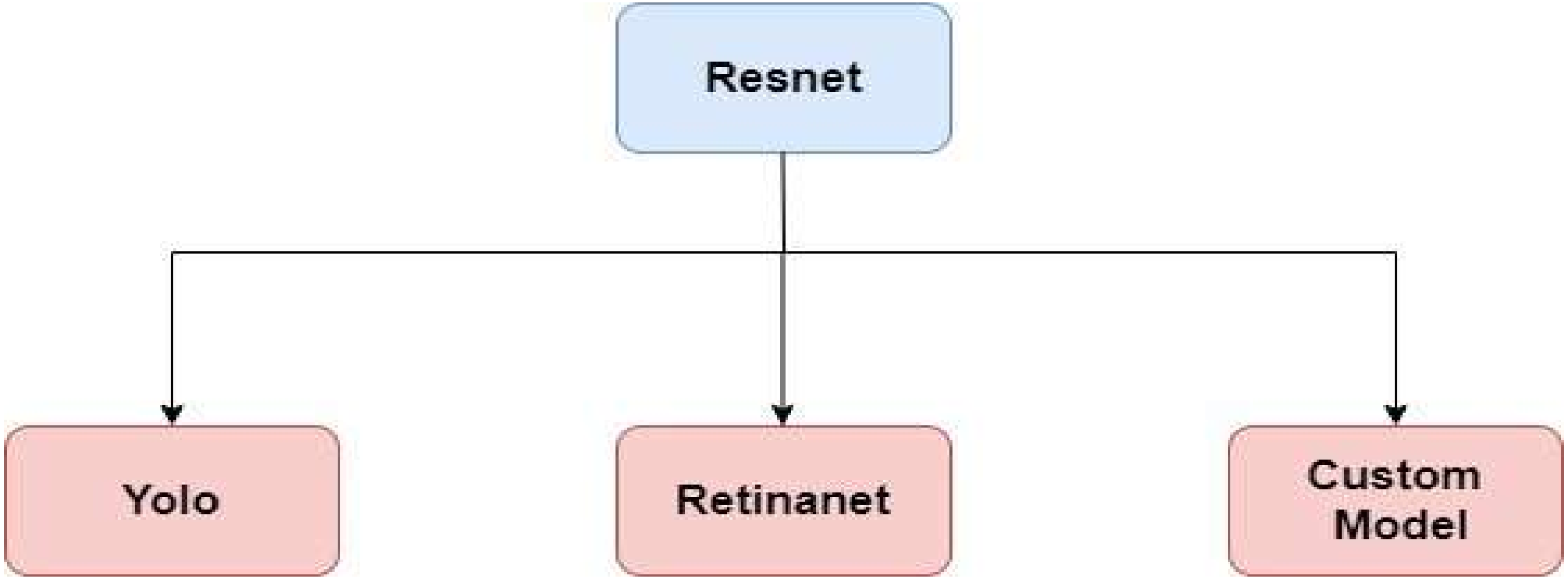
```
if (overlap == True):
    powercord_midpoint = ((p_x1 + p_x2) / 2, (p_y1 + p_y2) / 2)
    distance = ((((powercord_midpoint[0] - person_midpoint[0]) ** 2) +
                ((powercord_midpoint[1] - person_midpoint[1]) ** 2)) ** 0.5)
    powercord_img = returned_frame[powercord[3]:powercord[2], powercord[0]:powercord[1]]
    powercord_gray = cv2.cvtColor(powercord_img, cv2.COLOR_BGR2GRAY)
    powercord_blurred = cv2.GaussianBlur(powercord_gray, (11, 11), 0)
    powercord_thresh = cv2.threshold(powercord_blurred, 200, 255, cv2.THRESH_BINARY)[1]


    powercord_thresh = cv2.erode(powercord_thresh, None, iterations=2)
    powercord_thresh = cv2.dilate(powercord_thresh, None, iterations=4)
```





Methodology

- 2. Models Defining

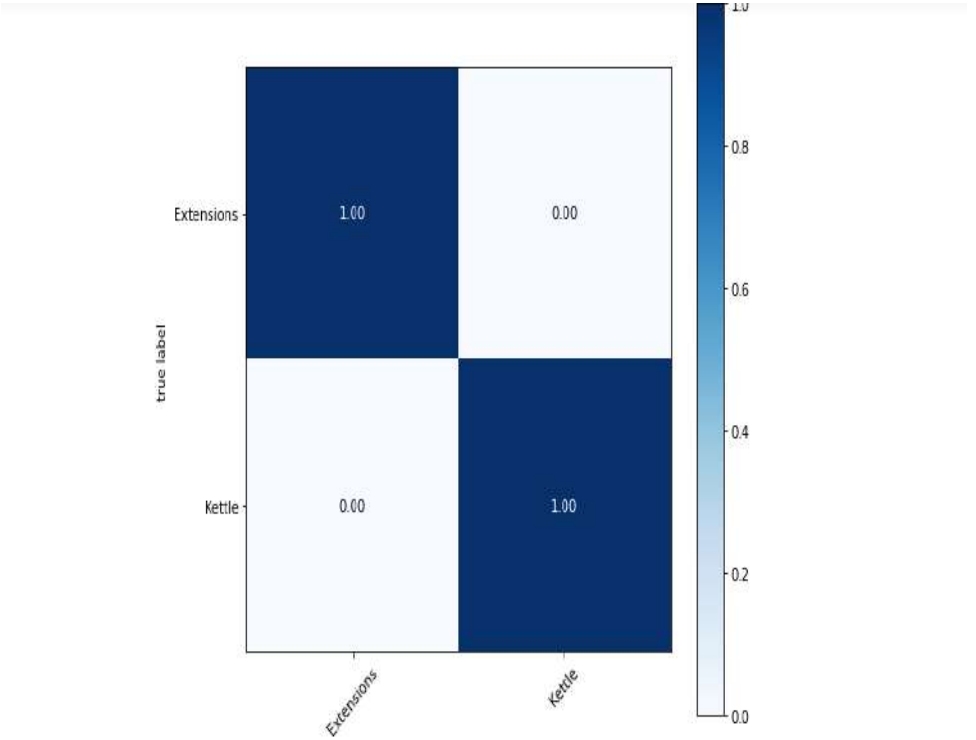


Methodology

Electrical object classification

```
In [33]: print('Classification Report')

         print(classification_report(validation_generator.classes, y_pred, target_names=category))
```

```
Classification Report
              precision    recall  f1-score   support

  Extensions       1.00      1.00      1.00        30
      Kettle       1.00      1.00      1.00        30

    accuracy                           1.00        60
   macro avg       1.00      1.00      1.00        60
weighted avg       1.00      1.00      1.00        60
```

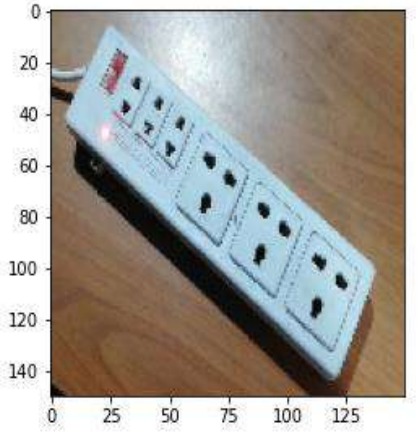


Completion of the project

- Test Results

```
In [57]: img.show
         # plt.subplots(figsize = (15,8))
         plt.imshow(img)
         for i in range(0,1):
             print('Predicted Class # : ',np.argmax(prediction[i]))
             print('Predicted Class Name :' ,category[np.argmax(prediction[i])])

Predicted Class # :  1
Predicted Class Name : Kettle
```


```
In [64]: img.show
         # plt.subplots(figsize = (15,8))
         plt.imshow(img)
         for i in range(0,1):
             print('Predicted Class # : ',np.argmax(prediction[i]))
             print('Predicted Class Name :' ,category[np.argmax(prediction[i])])

Predicted Class # :  0
Predicted Class Name : Extensions
```



Development

- **Import packages**

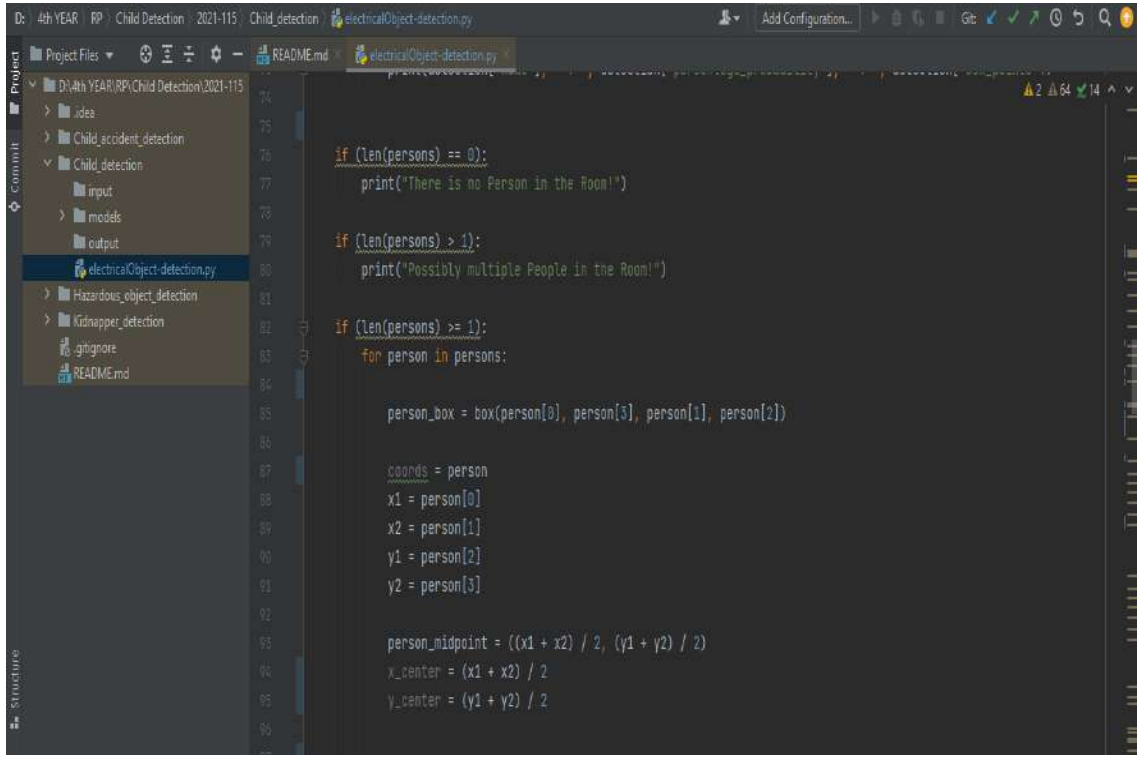
```
from imageai.Detection import ObjectDetection, VideoObjectDetection
import sys
import os
import cv2
import time
from shapely.geometry import box
```

- **Electrical Object Defining**

```
for detection in output_array:
    print(detection['name'] + " - " + str(detection['percentage_probability']) + " - " + str(detection['box_points']))
    if (detection['name'] == 'person' and detection['percentage_probability'] >= THRESHOLD_PERSON):
        count = count + 1
        persons.append(detection['box_points'])
        print(detection["name"], " : ", detection["percentage_probability"], " : ", detection["box_points"])

    print(detection['name'] + " - " + str(detection['percentage_probability']) + " - " + str(detection['box_points']))
    if ((detection['name'] == 'microwave' or detection['name'] == 'oven') and detection[
        'percentage_probability'] >= THRESHOLD_ITEM):
        microwave_count = microwave_count + 1
        microwaves.append(detection['box_points'])
        print(detection["name"], " : ", detection["percentage_probability"], " : ", detection["box_points"])

    if (detection['name'] == 'toaster' and detection['percentage_probability'] >= THRESHOLD_ITEM):
        toaster_count = toaster_count + 1
        toasters.append(detection['box_points'])
        print(detection["name"], " : ", detection["percentage_probability"], " : ", detection["box_points"])

    if (detection['name'] == 'hair dryer' and detection['percentage_probability'] >= THRESHOLD_ITEM):
        hairdryer_count = hairdryer_count + 1
        hairdryers.append(detection['box_points'])
        print(detection["name"], " : ", detection["percentage_probability"], " : ", detection["box_points"])
```

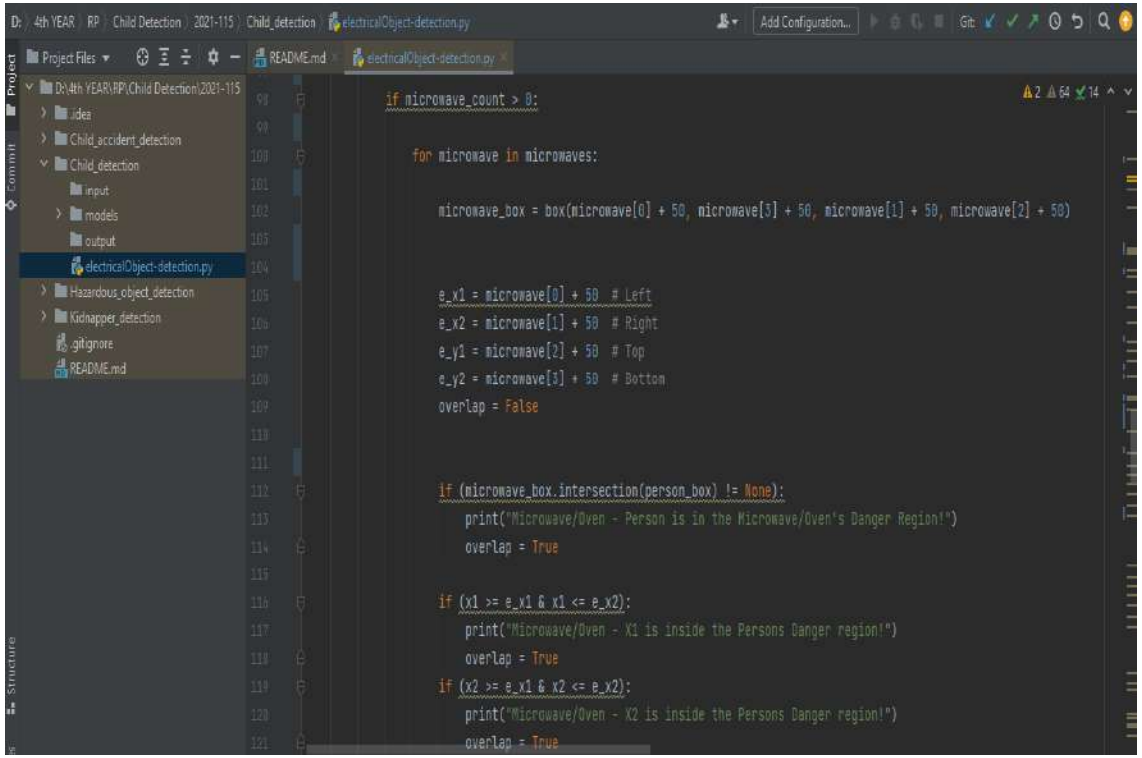
Development

Person Detection



```
if (len(persons) == 0):
    print("There is no Person in the Room!")

if (len(persons) > 1):
    print("Possibly multiple People in the Room!")

if (len(persons) >= 1):
    for person in persons:

        person_box = box(person[0], person[3], person[1], person[2])

        coords = person
        x1 = person[0]
        x2 = person[1]
        y1 = person[2]
        y2 = person[3]

        person_midpoint = ((x1 + x2) / 2, (y1 + y2) / 2)
        x_center = (x1 + x2) / 2
        y_center = (y1 + y2) / 2
```

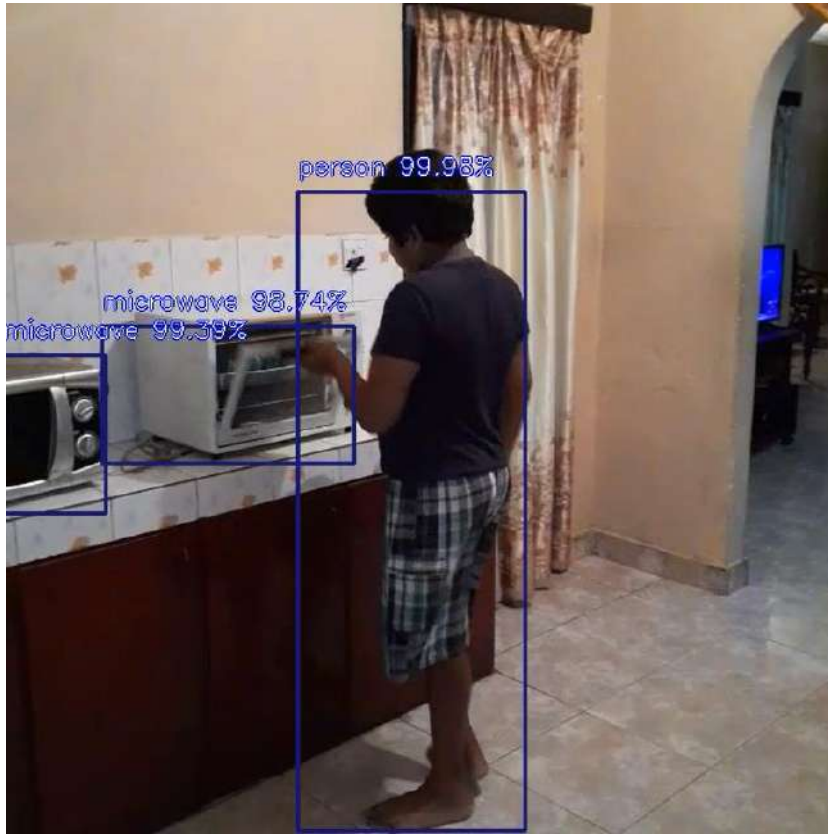
Object detection



```
if microwave_count > 0:

    for microwave in microwaves:

        microwave_box = box(microwave[0] + 50, microwave[3] + 50, microwave[1] + 50, microwave[2] + 50)


        e_x1 = microwave[0] + 50  # Left
        e_x2 = microwave[1] + 50  # Right
        e_y1 = microwave[2] + 50  # Top
        e_y2 = microwave[3] + 50  # Bottom
        overlap = False


        if (microwave_box.intersection(person_box) != None):
            print("Microwave/Oven - Person is in the Microwave/Oven's Danger Region!")
            overlap = True

        if (x1 >= e_x1 & x1 <= e_x2):
            print("Microwave/Oven - X1 is inside the Persons Danger region!")
            overlap = True
        if (x2 >= e_x1 & x2 <= e_x2):
            print("Microwave/Oven - X2 is inside the Persons Danger region!")
            overlap = True
```

Completion of the project

With Yolo model

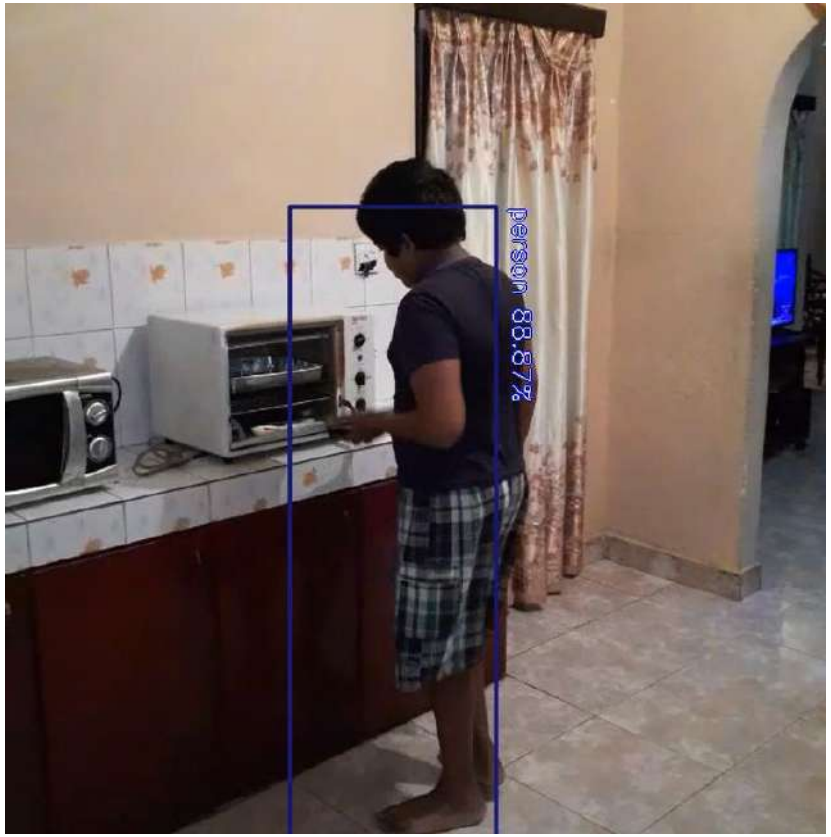


Alert

```
Terminal:  Local  +
Microwave/Oven - Person is in the Microwave/Oven's Danger Region!
Microwave/Oven - X1 is inside the Persons Danger region!
Microwave/Oven - X2 is inside the Persons Danger region!
Microwave/Oven - Y1 is inside the Persons Danger region!
Microwave/Oven - Y2 is inside the Persons Danger region!
No Toasters Found
No Hair Dryers Found
Microwave/Oven - Person is in the Microwave/Oven's Danger Region!
Microwave/Oven - X1 is inside the Persons Danger region!
Microwave/Oven - X2 is inside the Persons Danger region!
Microwave/Oven - Y1 is inside the Persons Danger region!
Microwave/Oven - Y2 is inside the Persons Danger region!
No Toasters Found
No Hair Dryers Found
```

Completion of the project

With Retinanet model



Alert

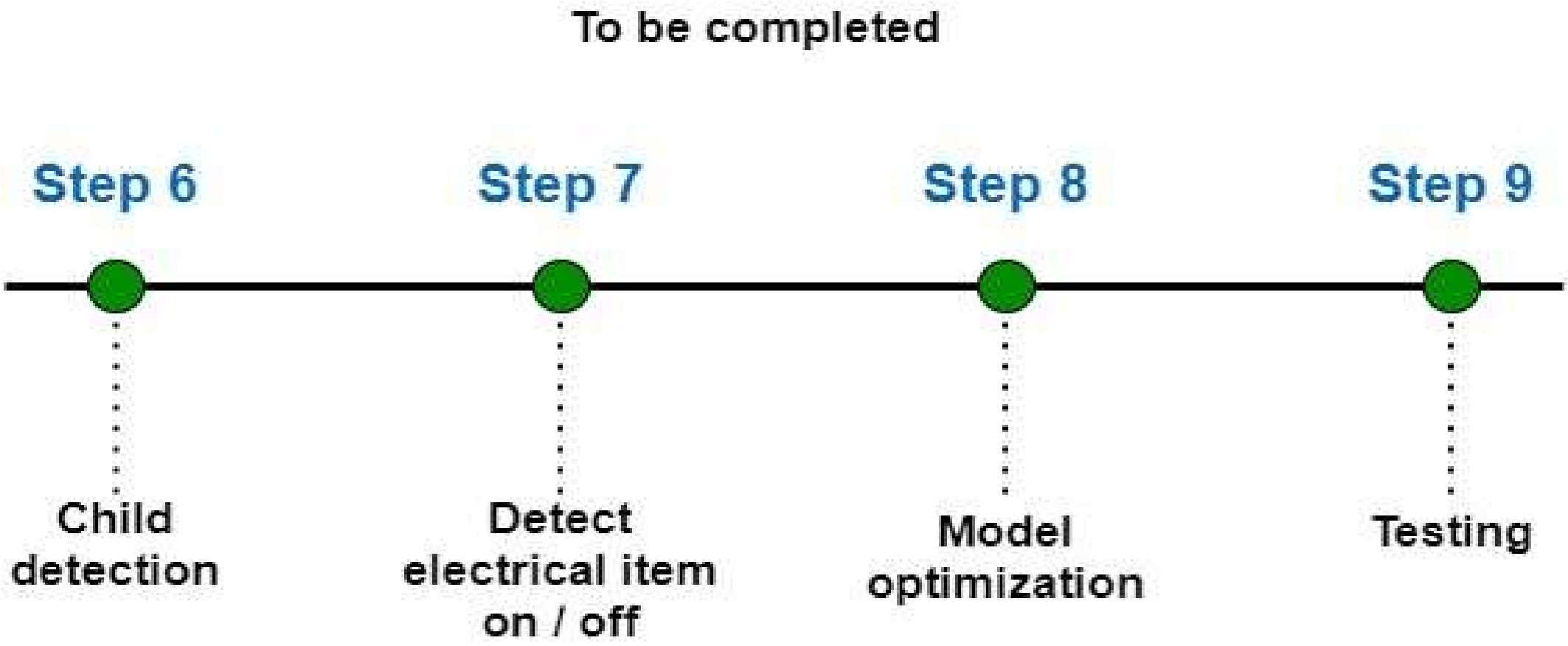
```
person - 93.91705989837646 - [53, 96, 537, 257]
person  :  93.91705989837646  :  [53, 96, 537, 257]
person - 93.91705989837646 - [53, 96, 537, 257]
No Microwaves or Ovens Found
No Toasters Found
No Hair Dryers Found
person - 92.12513566017151 - [115, 112, 560, 255]
person  :  92.12513566017151  :  [115, 112, 560, 255]
person - 92.12513566017151 - [115, 112, 560, 255]
No Microwaves or Ovens Found
No Toasters Found
No Hair Dryers Found
```

Completion of the project

Test results


```
oven - 98.48571419715881 - [337, 143, 468, 351]
oven  :  98.48571419715881  :  [337, 143, 468, 351]
Microwave/Oven - Person is in the Microwave/Oven's Danger Region!
Microwave/Oven - X1 is inside the Persons Danger region!
Microwave/Oven - X2 is inside the Persons Danger region!
Microwave/Oven - Y1 is inside the Persons Danger region!
Microwave/Oven - Y2 is inside the Persons Danger region!
No Toasters Found
```

To be completed



References

- [1] J. Chatrath, P. Gupta, P. Ahuja, A. Goel, and S. M. Arora, "Real time human face detection and tracking," *2014 Int. Conf. Signal Process. Integr. Networks, SPIN 2014*, pp. 705–710, 2014, doi: 10.1109/spin.2014.6777046.
- [2] M. Wu and Y. Yuan, "Gender classification based on geometry features of palm image," *Sci. World J.*, vol. 2014, 2014, doi: 10.1155/2014/734564.
- [3] X. Han, H. Ugail, and I. Palmer, "Gender classification based on 3D face geometry features using SVM," *2009 Int. Conf. CyberWorlds, CW '09*, pp. 114–118, 2009, doi: 10.1109/CW.2009.41.
- [4] A. Samadi and H. Pourghassem, "Children Detection Algorithm Based on Statistical Models and LDA in Human Face Images," pp. 206–209, 2013, doi: 10.1109/CSNT.2013.52.
- [5] O. F. Ince, J. S. Park, J. Song, and B. W. Yoon, "Child and Adult Classification Using Ratio of Head and Body Heights in Images," vol. 3, no. 2, pp. 2–4, 2014, doi: 10.7763/IJCCE.2014.V3.304.
- [6] A. K. Choobeh, "An Image-Based Method of Distinguishing Children from Adults," *Int. J. Inf. Electron. Eng.*, vol. 3, no. 5, pp. 533–535, 2013, doi: 10.7763/ijiee.2013.v3.372.
- [7] L. L. Shen and Z. Ji, "Modelling geiometric features for face based age classification," *Proc. 7th Int. Conf. Mach. Learn. Cybern. ICMLC*, vol. 5, no. July, pp. 2927–2931, 2008, doi: 10.1109/ICMLC.2008.4620909.
- [8] Y. K. Dubey and S. Damke, "Baby monitoring system using image processing and IoT," *Int. J. Eng. Adv. Technol.*, vol. 8, no. 6, pp. 4961–4964, 2019, doi: 10.35940/ijeat.F9254.088619.

Thank you