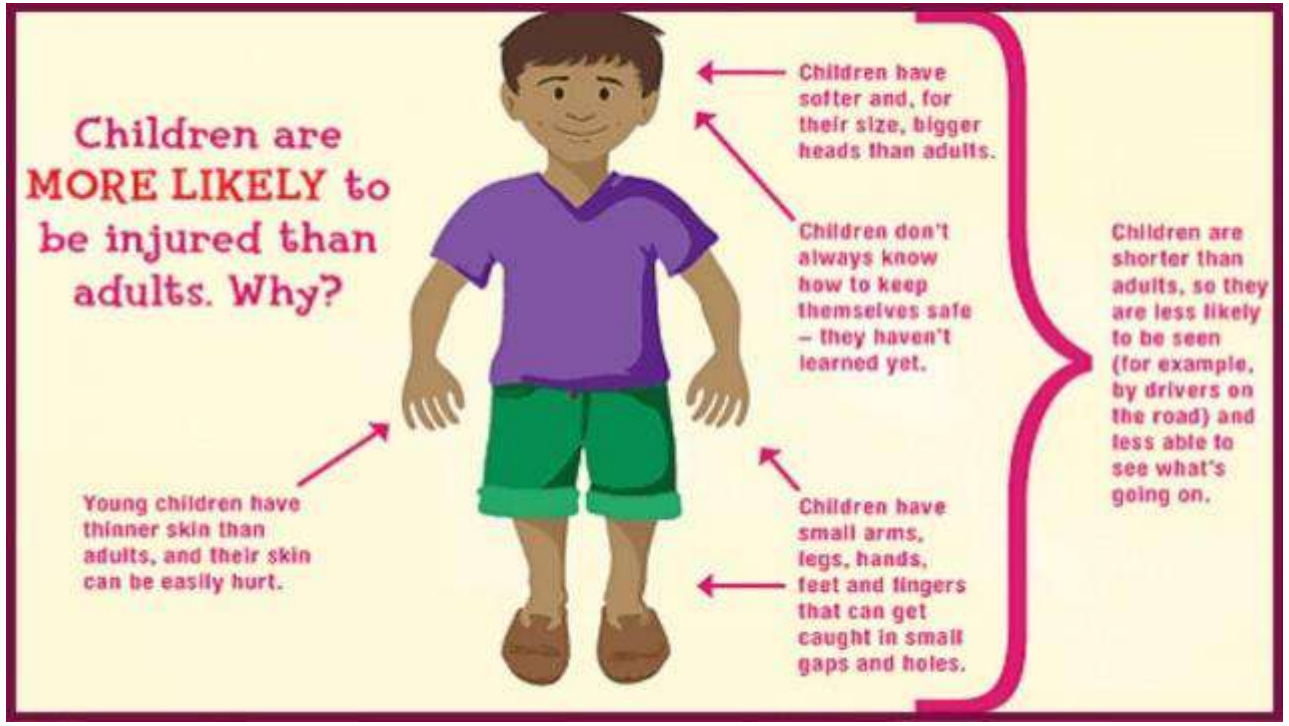
Harmful Child Activity Detection and Prevention Assistance

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.

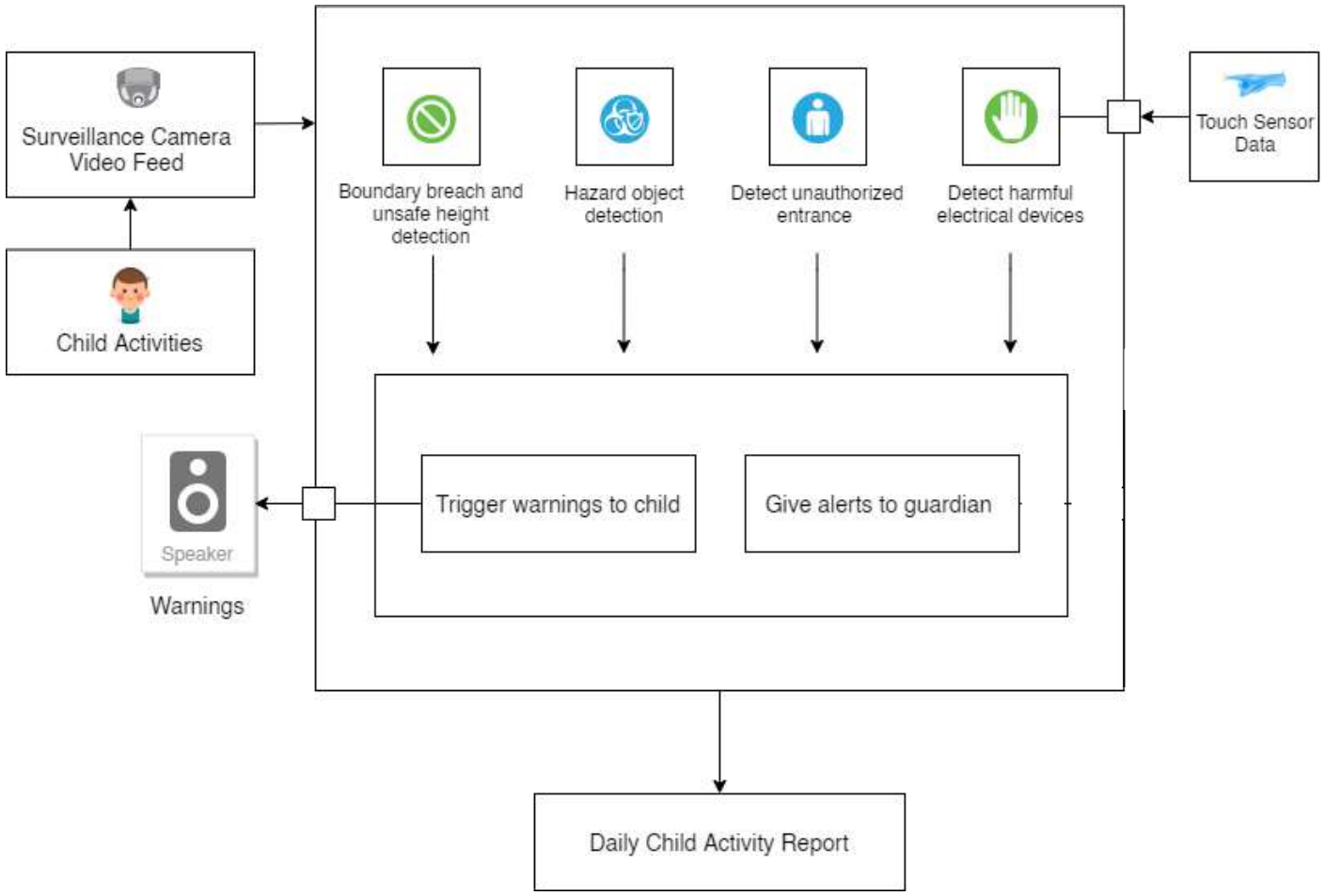
Objective 3:

- 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



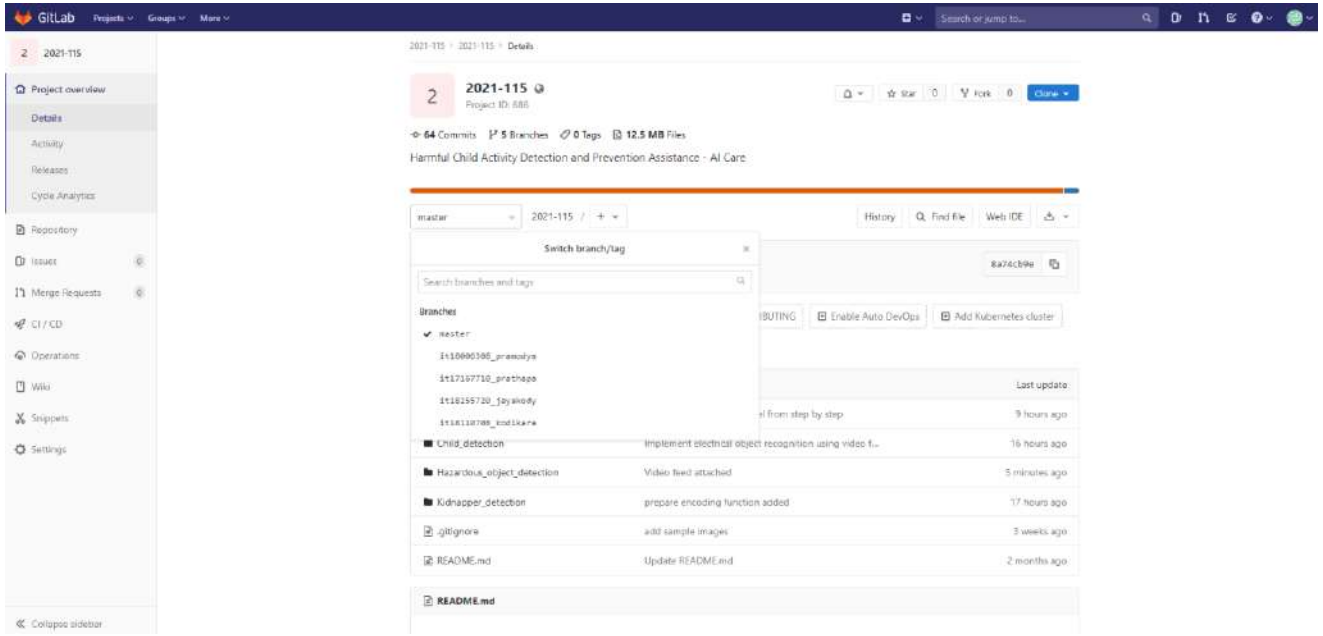
Commercialization

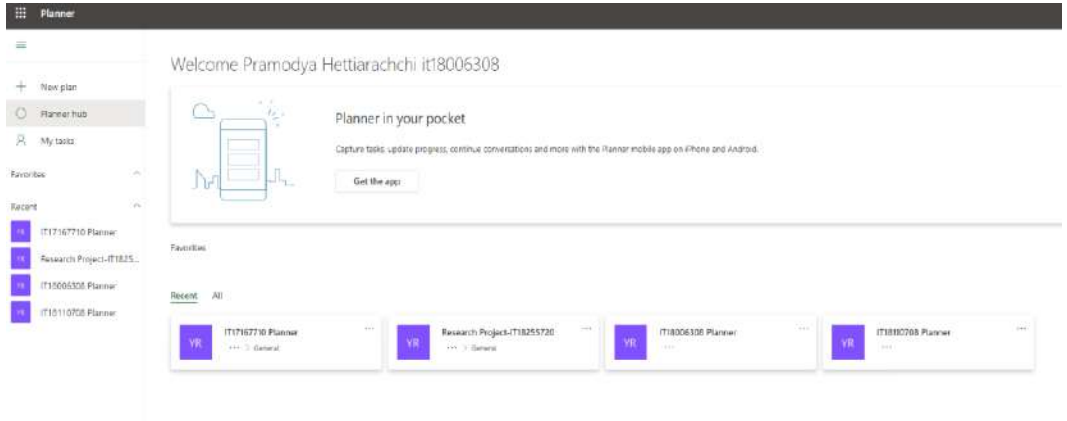
- In a society where families with both parents working has become a common norm, chi ldren has left to grow up by themselves.
- Children between the age 5 year and 12 year is the most crucial period where a child need a lot of parental attention.
- Thereby, AICare has the potential to be the latest 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

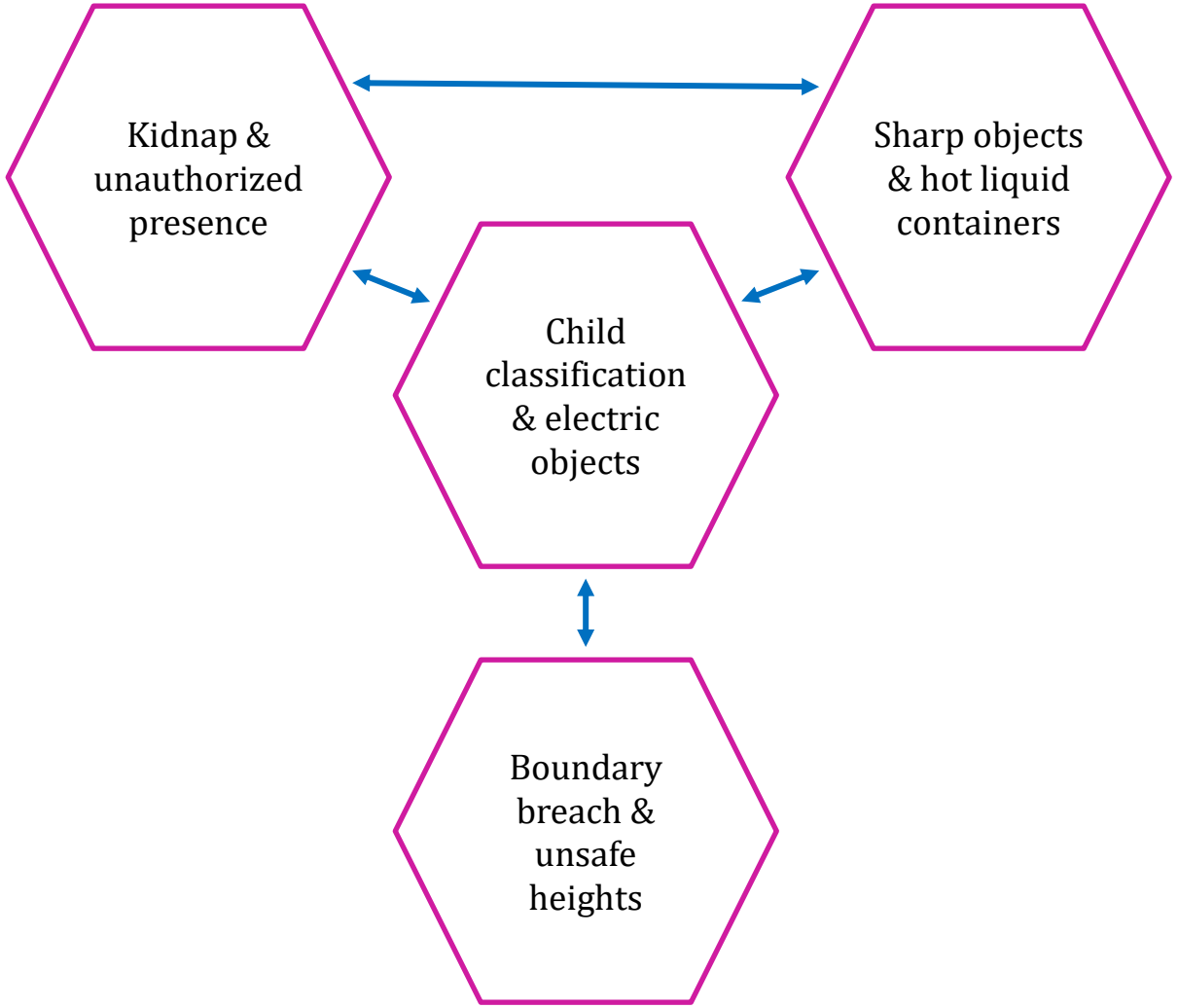
1. Keep code simple and consistent.
   - Reduce unnecessary complexity
   - Easy to read, upgrade and debug

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

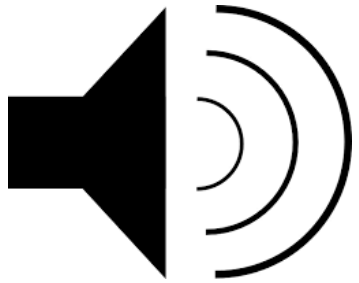
4. Set realistic time estimates
   - Prepare the time breakdowns and task planners at the beginning itself(gantt chart, planners).

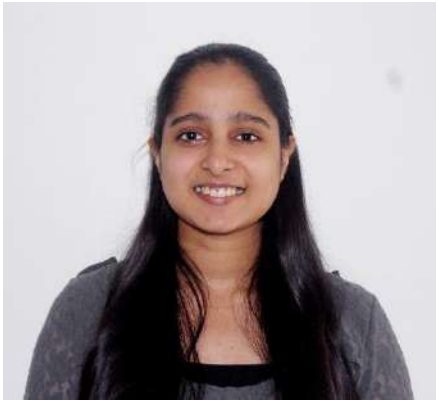
Integration



For 100%

Customized Alerting





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

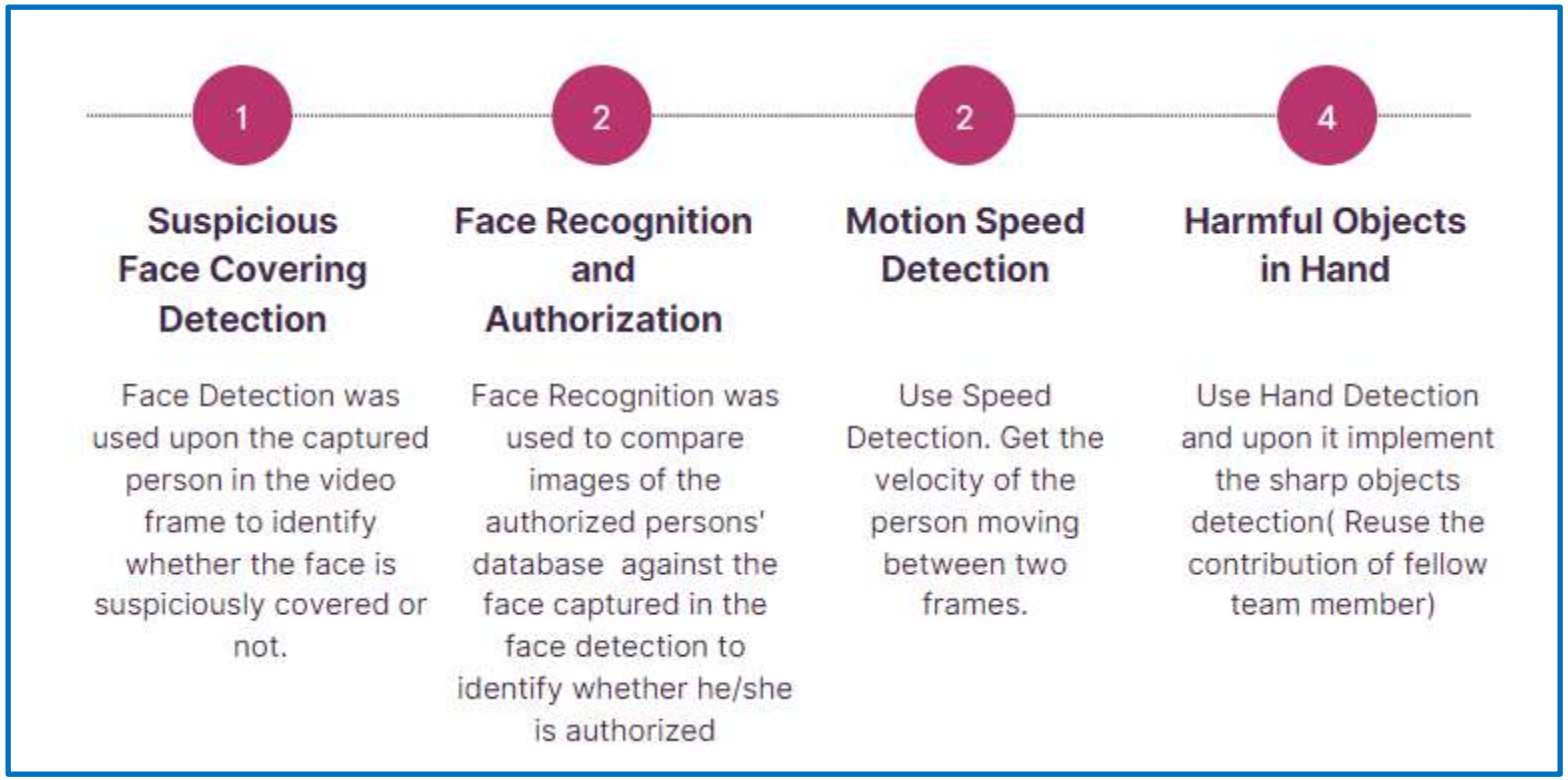
Technologies

python™

NumPy

OpenCV

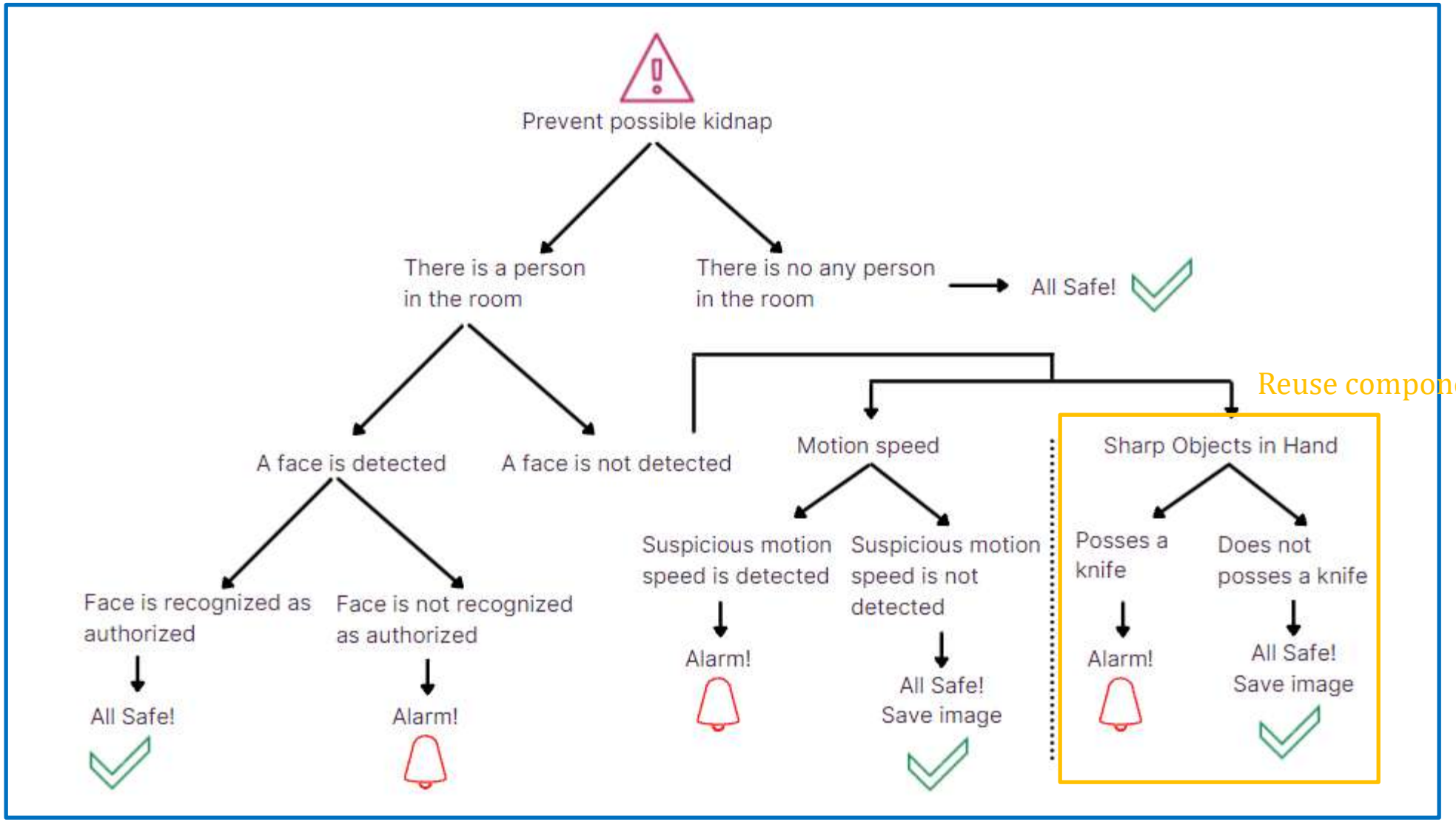
Completion of the project



✓ 90%

Methodology

90%



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

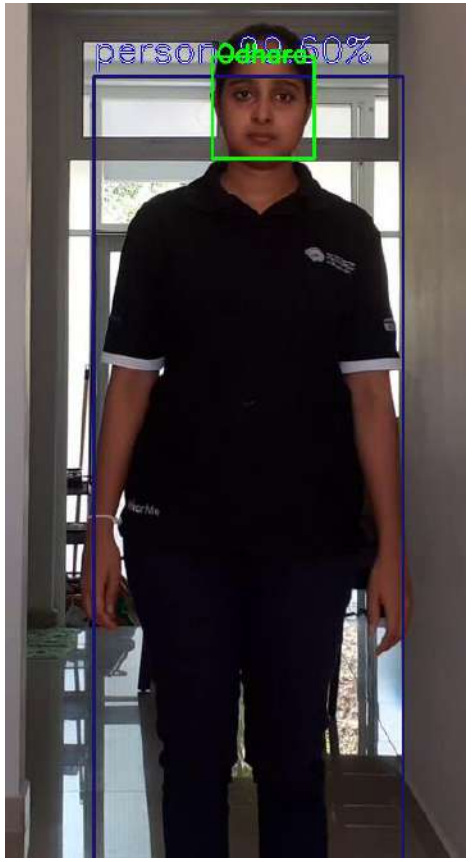
Step 3 : If a face can be identified, then carry out face recognition. If a face cannot be identified but a person can be identified, show warning.

Step 4 : If a face can be identified, then carry out face recognition. Then check if it passes facial recognition. If it passes show person name, 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

Suspicious person detected



Not Suspicious & Face is Recognized

Methodology

Step 5 : Check for suspicious motion speed

```
# Check Speed after face recog - not recog
if (elapsed_time > TIME_THRESHOLD):
  # Calculate Distance
  distance_x = abs(bbox1[0] - initial_pos[0])
  print("Distance Horizontal: "+str(distance_x))
  distance_y = abs(bbox1[1]) - abs(initial_pos[1])
  print("Distance Vertical: "+str(distance_y))
  speed_y=distance_y/elapsed_time
  speed_x=distance_x/elapsed_time
  print("Speed Horizontal: "+str(speed_x))
  print("Speed Vertical: "+str(speed_y))

  # Reset Initial Position and Time
  initial_pos=None
  initial_time=time.time()

  if ((speed_y > SPEED_THRESHOLD_Y) or (speed_x > SPEED_THRESHOLD_X)):
    warning_message+="Suspicious Movement Speed\n"
```

Test Results

Suspicious motion speed detected

Methodology

Step 6 : Check for knives

```
# Check if the person is holding a knife
# Get Predictions from the Hazardous Objects Detection Model
detections2, width_ratio2, height_ratio2 = darknet_helper2(frame, width2, height2)

# loop through detections2 and draw them on transparent overlay image
for label, confidence, bbox in detections2:
  if label == "Knife" and float(confidence) >= CONFIDENCE_KNIFE:
    left, top, right, bottom = bbox2points(bbox)
    left, top, right, bottom = int(left * width_ratio2), int(top * height_ratio2),
    int(right * width_ratio2), int(bottom * height_ratio2)
    cv2.rectangle(frame, (left, top), (right, bottom), class_colors2[label], 2)
    cv2.putText(frame, "{} [{:.2f}]".format(label, float(confidence)),
                (left, top - 5), cv2.FONT_HERSHEY_SIMPLEX, 0.5,
                class_colors2[label], 2)
    # Person - A (Left1, Bottom1) (x1,y1) # B (Right1, Top1) (x2,y2)
    # Extension - C (Left, Bottom) (x3,y3) # D (Right, Top) (x4,y4)
    left2=0 if (left-PROXIMITY_KETTLE)<=0 else (left-PROXIMITY_KETTLE)
    top2=0 if (top-PROXIMITY_KETTLE)<=0 else (top-PROXIMITY_KETTLE)
    right2=right+PROXIMITY_KETTKE
    bottom2=bottom+PROXIMITY_KETTLE

    if (intersection([left2, top2, right2, bottom2], [left1, top1, right1, bottom1])):
      warning_message+="Person holding a Knife\n"
    else:
      no_intersection=True
```

Step 7 : Check for intersection of knife and person bounding boxes

```
# Calculate the intersection between two bounding boxes
def intersection(a,b):
  x = max(a[0], b[0])
  y = max(a[1], b[1])
  w = min(a[0]+a[2], b[0]+b[2]) - x
  h = min(a[1]+a[3], b[1]+b[3]) - y
  if w<0 or h<0: return False # or (0,0,0,0) ?
  return True
```

Test Results

Suspicious motion speed

Suspicious face covering

Sharp object in hand

Kidnapper detected

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 and Research Problem

- The increasing number of accidents in the home causes more than 37% of death in children between the age of one and four due to harmful child activities from hazardous objects
- Most accidents are happened due to the curiosity of the children and lack of supervision
- There are many 'child-proofing' methods and devices used in homes in the present.
- The existing child safety practices do not provide extensive information on hazardous indicates
- How can parents ensure their child’s early development stage safe ?

Research Question

- How to Identify the hazardous object is sharp or hot liquid container 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

- Some approaches that have been proposed over the last few years

1. Security Camera with Baby Monitoring System
2. IoT based smart baby monitoring system

- Automated surveillances had significant growth since past decade.
- Surveillance was used for abnormal behavior detection, feature detection, object detection, action recognition etc.
- None of these surveillance methodologies had potential ability in child safeguard.
- At present, object detection is a viral topic when consider the computer vision. Only the object detection is not sufficient to address this issue.

Objectives

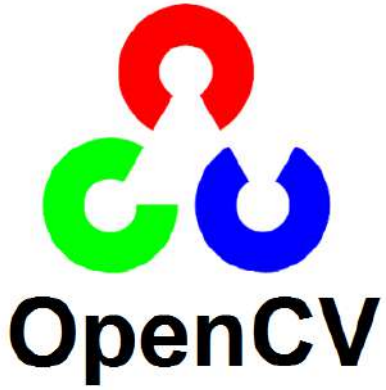
- Main Objective

  - 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.

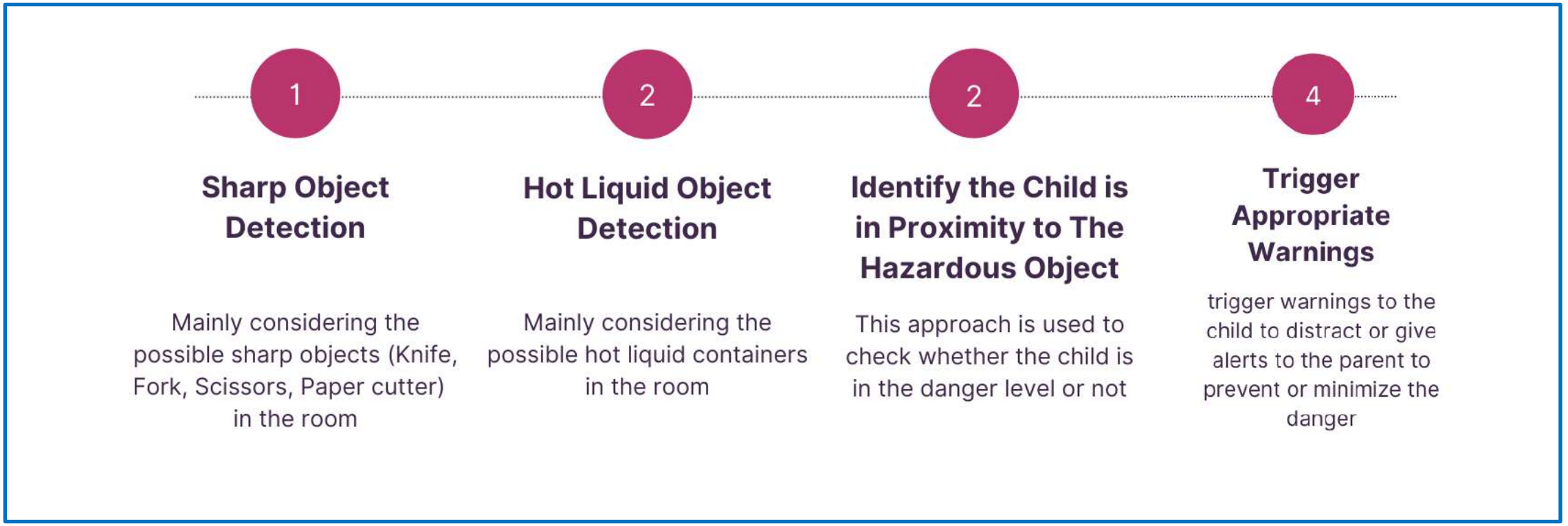
- Specific Objectives

  - Detect possible sharp objects COMPLETED
  - Detect possible hot liquid containers COMPLETED
  - Detect child in proximity to hazardous object COMPLETED
  - Increase the accuracy and efficiency of the detection model COMPLETED
  - Trigger warnings to child or give alert to parent as suitable to the situation COMPLETED

Technologies

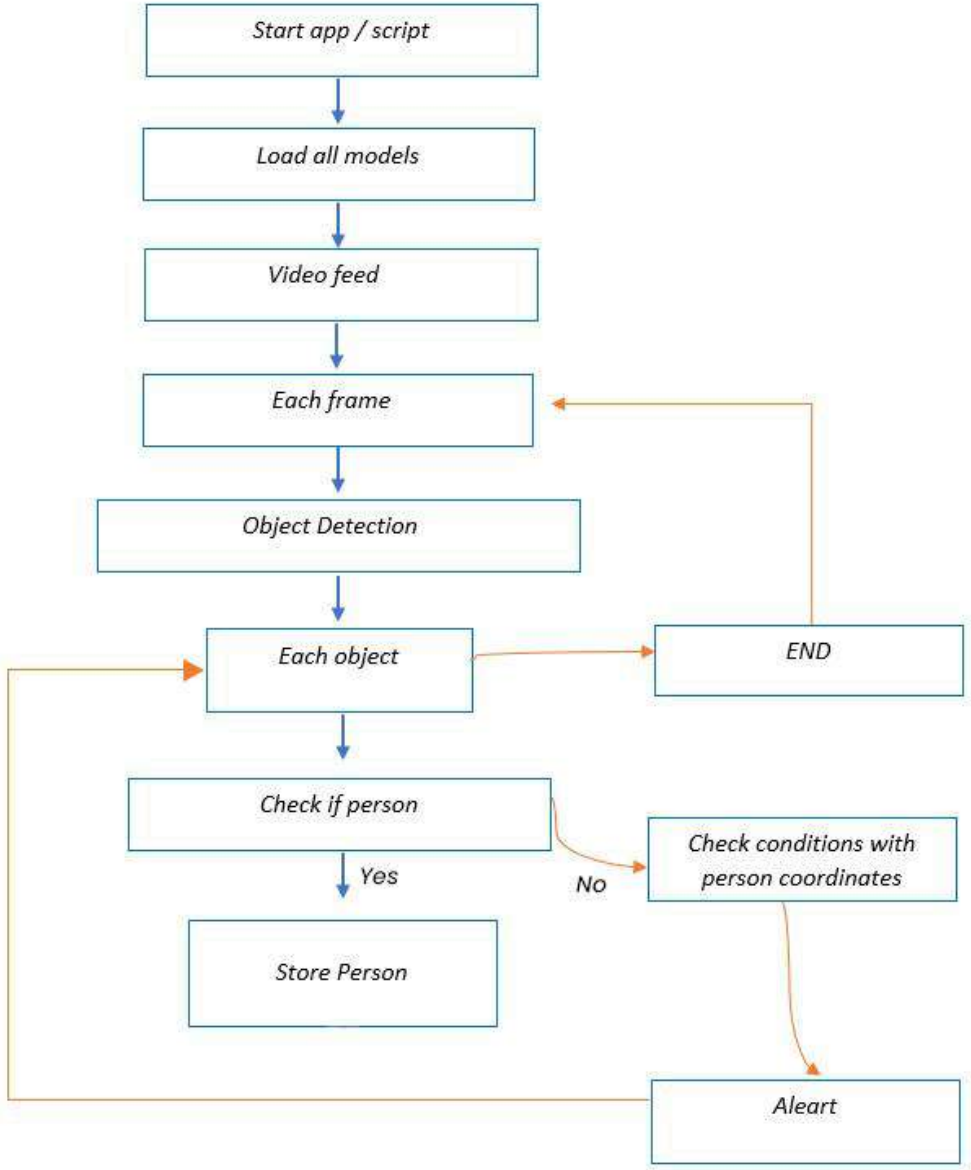


Completion of the project

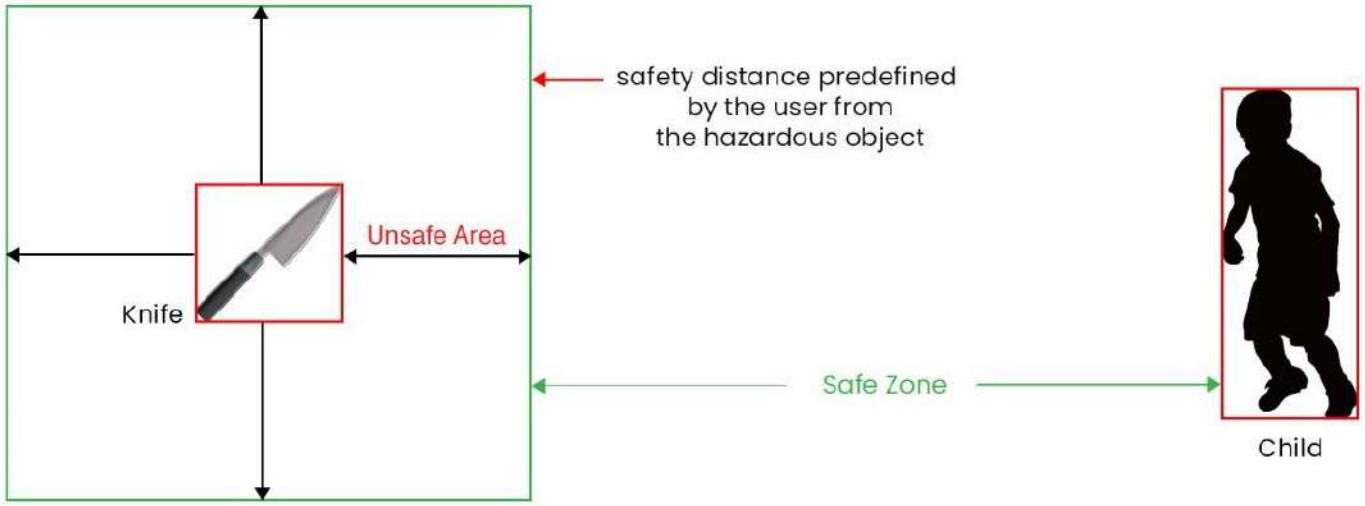


90%

Methodology



Methodology

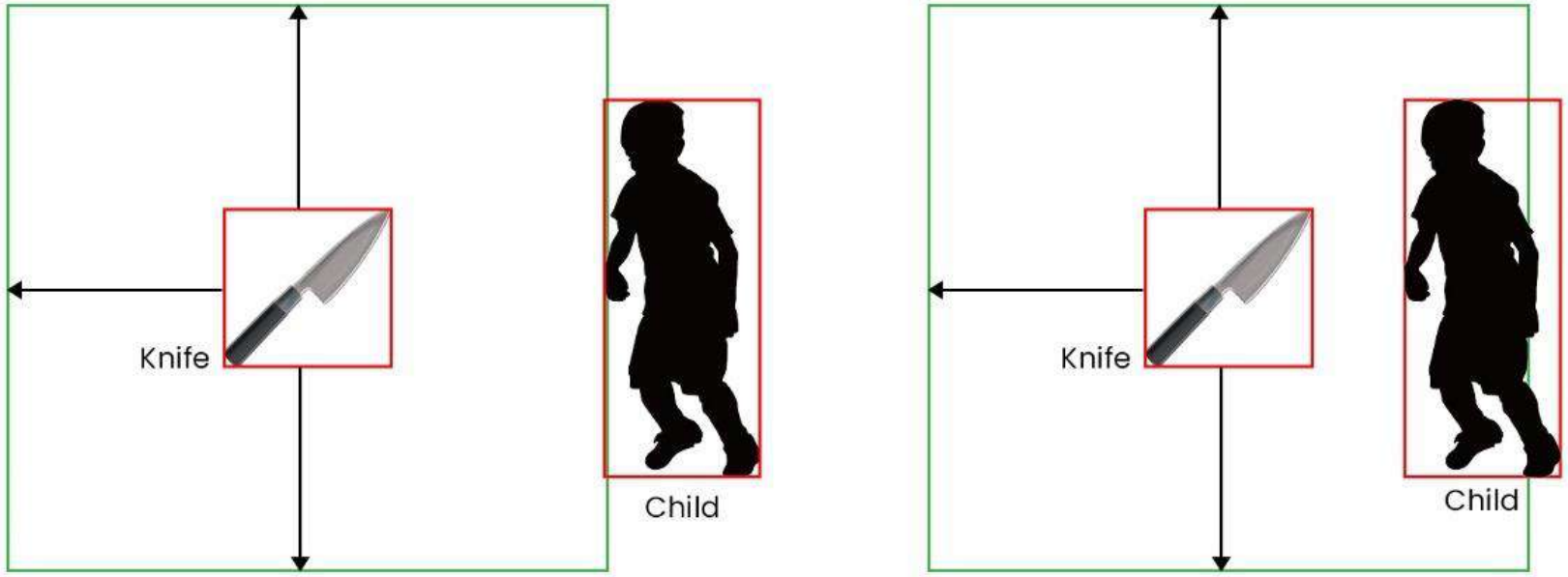


Initially, system detects where the hazardous objects are located and marks the bounding boxes and does the same for the child.

The danger distance which is carried out as the major detecting fact is a predefined variable by the user.

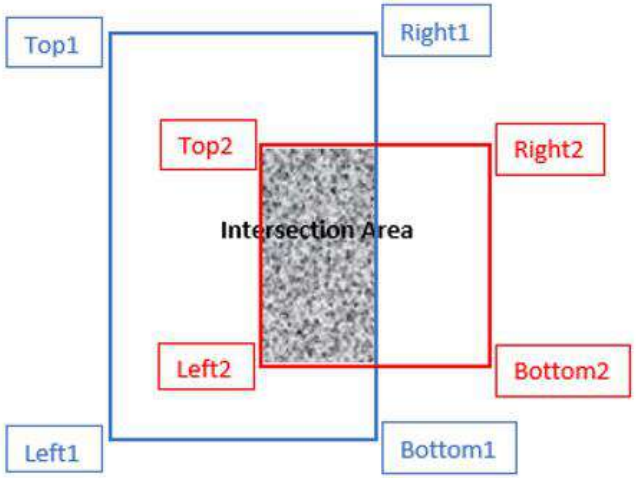
The detector would process the video feed only when there is a child detected within the area

Methodology



The proximity level is tracked by the detector with the real-time process of detection by repeatedly comparing with the predefined danger level, in order to execute the trigger warnings of the detector.

Methodology




```
x = max(Left1, Left2)
y = max(Top1, Top2)
width = min(Left1 + Right1, Left2 + Right2) - x
height = min(Top1 + Bottom1, Top2 + Bottom2) - y
```

This method is used to identify whether the child is in proximity to the hazardous object or not

This logic is implemented by using a calculation to measure the width and the height of the intersection area of the bounding boxes.

Test Results

Test Results

IT17167710 | PRATHAPA D.M.J

DATA SCIENCE

Background

- Falls are the most common cause of accidental injury to children.

  eg : steps , slip ,trips , stumps and etc.

- While most falls aren’t serious but some falls can lead to death or long-term disability.
- Fall accidents may occur when climbing to a high place in a daily living space.
- Child fall accidents are thought to occur according to the following process: a baby climbs to a height through a product (e.g., a chair, a table, or a sofa) and then falls from a high location by losing balance.

Research Questions

- ✓ How to prevent the child from stepping out of the safe zone of the room
- ✓ How to detect the action of falling
- ✓ How to determine the possible furniture that child can climb on
- ✓ How to prevent the child from climbing on furniture
- ✓ 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 , fall accidents and Child climbing to a dangerous position 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 breacl COMPLETED

✓ Fall Detection (Detecting the action of falling) COMPLETED

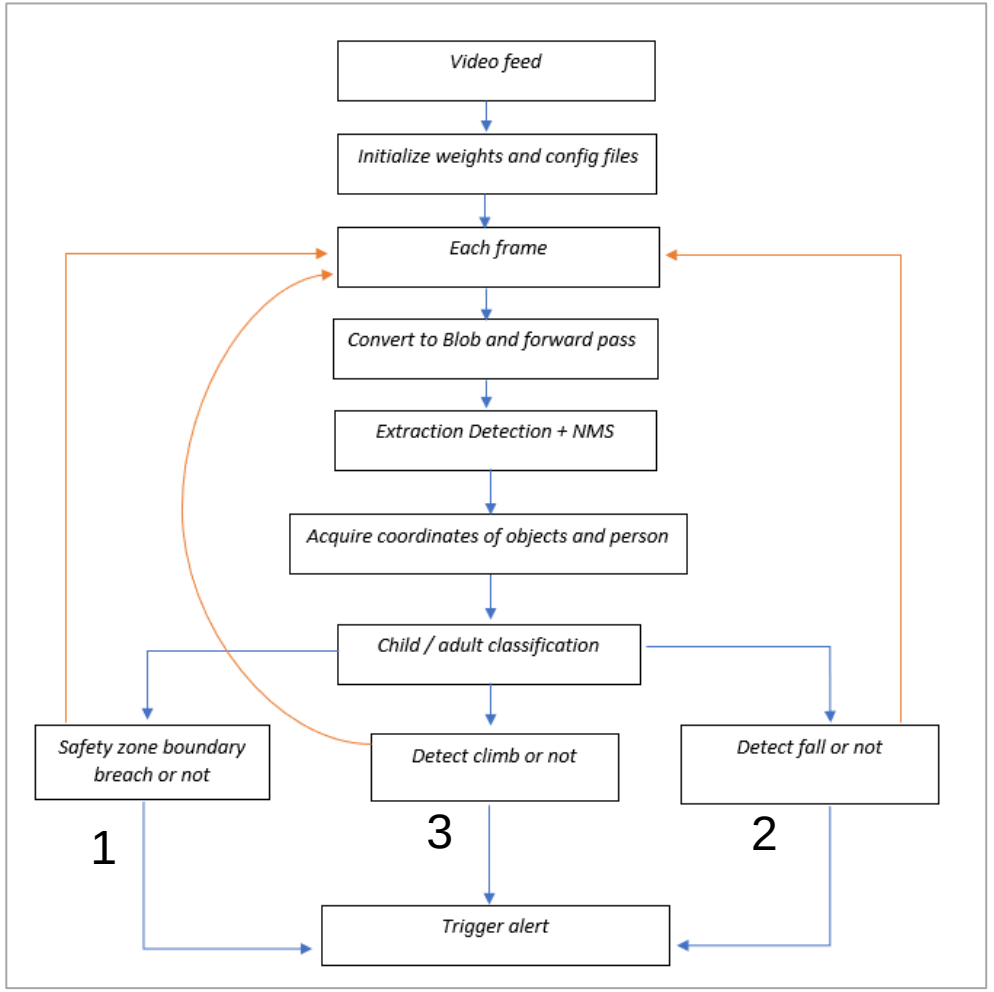
✓Capture spot unsafe heights from the current position (Detecting the action of Climbing) COMPLETED

✓Alerting (communicate the incident efficiently to the parent or caregiver.) - WIP

Comparison between existing Research and Products

	[1][2][3] [4][5]	[36][37] [37][38]	[15][16]	[19][20] [22][23] [24]	[29][30] [34]	[32][33]	[35]	Our solution (AI care)
Door detection	✓	✗	✗	✗	✗	✗	✗	✓
Furniture object detection	✗	✓	✗	✗	✗	✗	✗	✓
Build for child	✗	✗	✗	✗	✗	✗	✓	✓
Furniture object-based climb detection	✗	✗	✗	✗	✗	✗	✗	✓
Child safety zone exiting feature	✗	✗	✗	✗	✗	✗	✗	✓
Non sensor based	✗	✗	✗	✗	✓	✓	✓	✓
Orientation or threshold-based fall detection	✗	✗	✗	✗	✓	✓	✓	✓
Combine approach of orientation, speed and angle for fall detection	✗	✗	✗	✗	✗	✗	✗	✓
Taking prompt, responsive action /alert	✗	✗	✓	✓	✗	✓	✗	✓

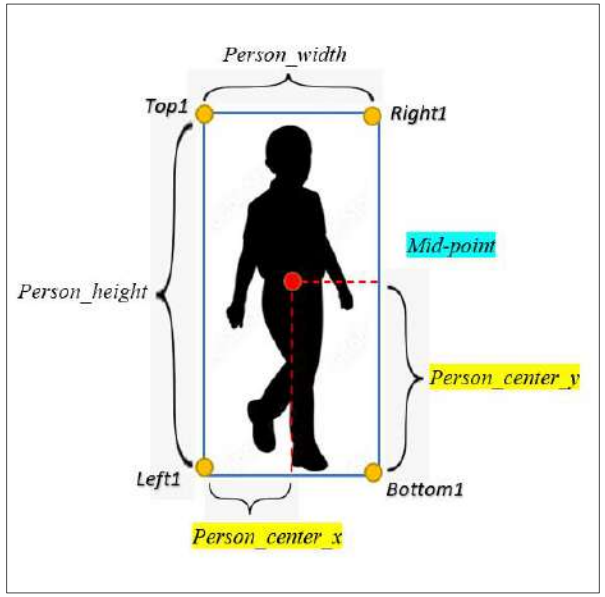
The overall framework of the proposed Approach



For the 50% completion

- Child mid point calculation
- Door detection model
- Using Pythagoras theorem calculate distance between door and the child.
- Complete the safety zone exiting detection.
- Falls detection using pose classification.

The Center Point Calculation



Mid-Point Calculation

Person_height = Top1 - Bottom1

Person_width = Right1 - Left1

Person_center_y = Bottom1 + (Person_height / 2)

Person_center_x = left1 + (Person_width / 2)

Capture safety zone boundary breach

- **Calculate the mid-points:**

  Calculate the midpoint of Child

  Calculate the midpoint of Door

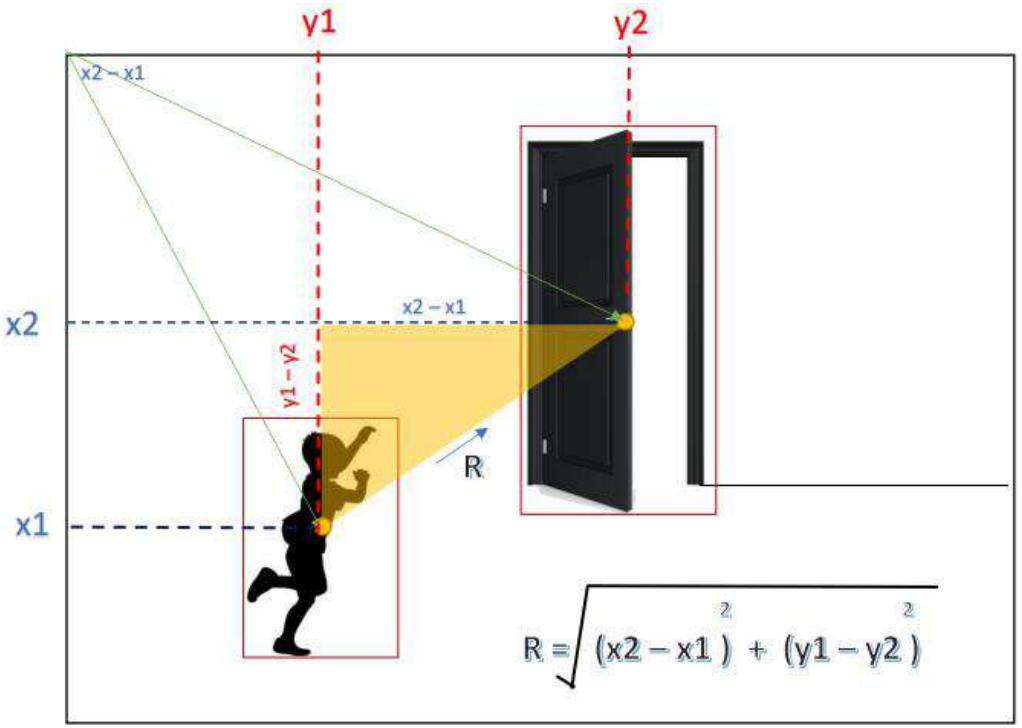
- **Door Detection**

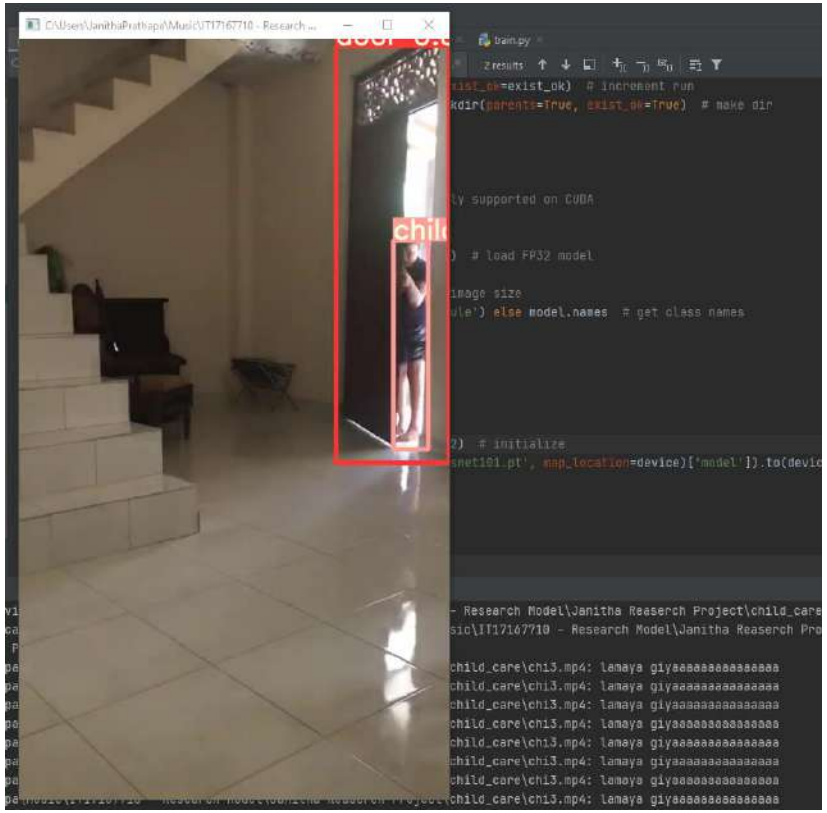
a) normal door b) cabinet door c) refrigerator door

Matrices	YOLOv4	YOLOv5
Mean Average Precision (MAP)	93.07%	97.23%
Precision	0.78	0.92
Recall	0.97	0.96
F1 Score	0.86	0.93

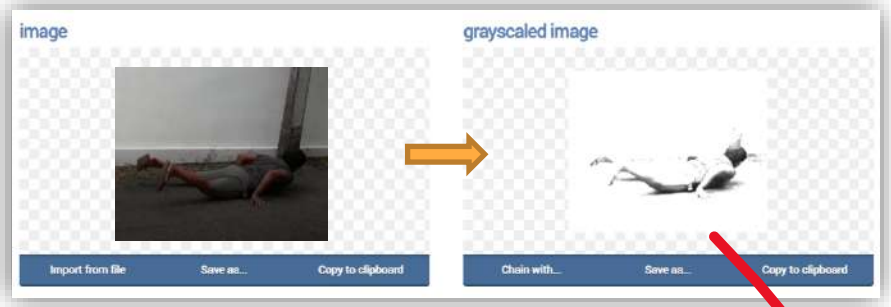
Completion of the project

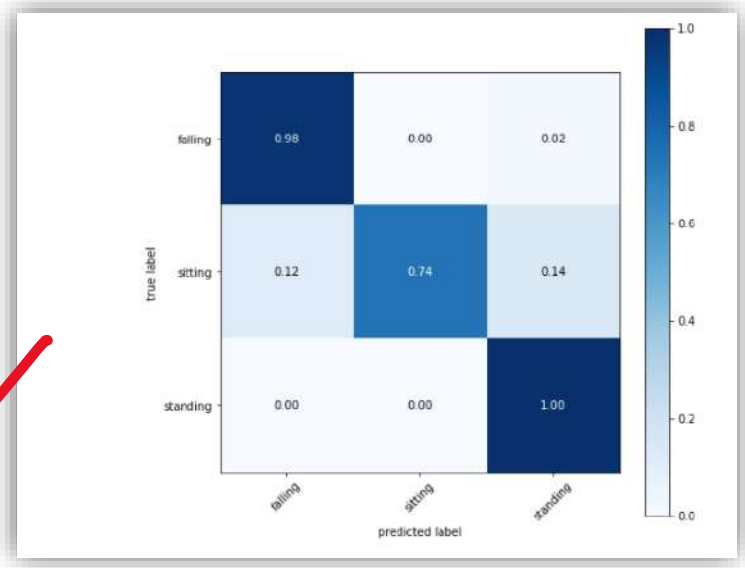
- ✓ Detect safety zone boundary breach (screenshots) - ***Completed***





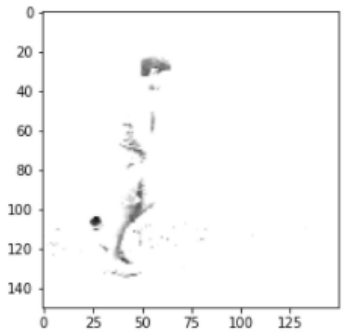
Child Fall Detection





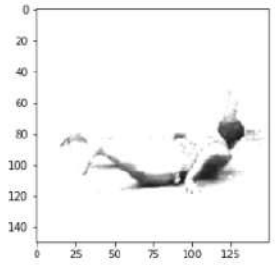

```
img = image.load_img('C:/Users/JanithaPrathapa/Music/IT17167710 - Research
img_array = image.img_to_array(img)
img_batch = np.expand_dims(img_array, axis=0)
img_preprocessed = preprocess_input(img_batch)
prediction = new_model.predict(img_preprocessed)
img.show
plt.imshow(img)
for i in range(0,1):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

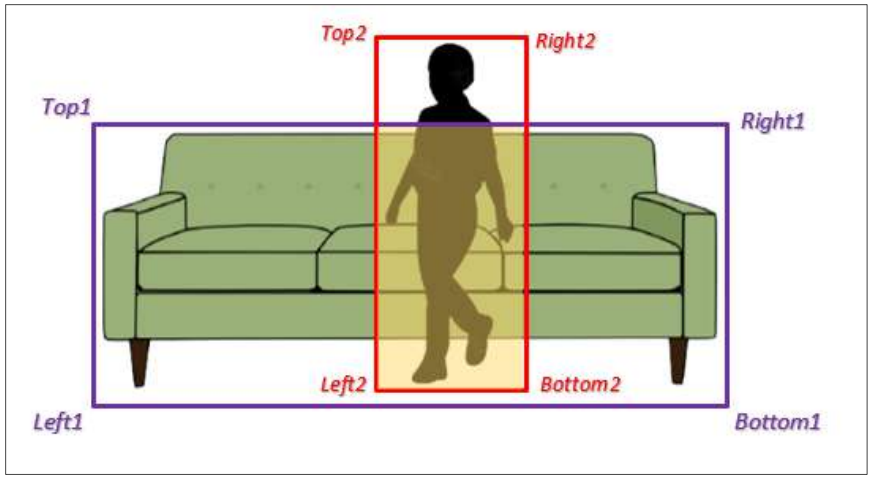
```
Predicted Class # :  0
Predicted Class Name : falling
```



For PP2 – 90%

The Overlapping Calculation- (intersection)

COMPLETED



Overlapping Calculation

$x = max (Left1 , Left2)$

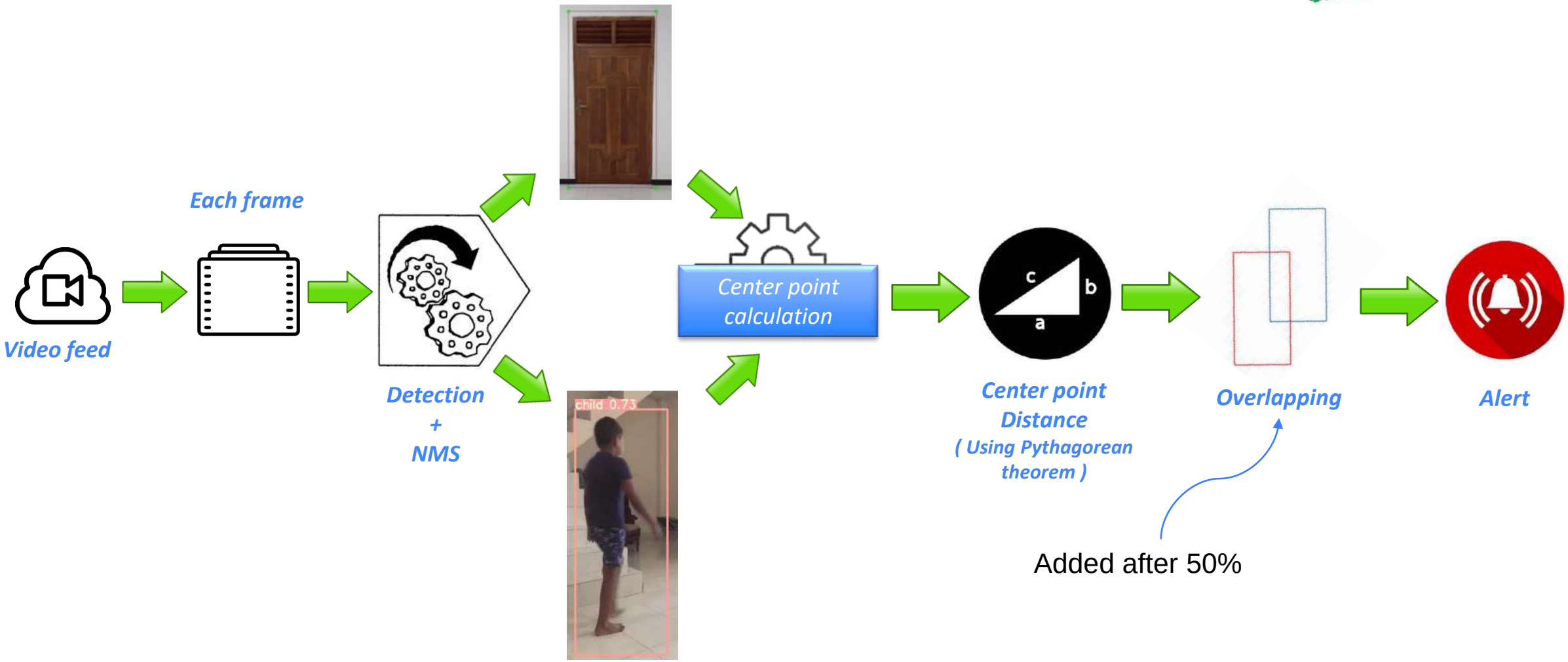
$y = max (Top1 , Top2)$

$width = min (Left1 + right1 , Left2 + Right2) – x$

$height = min (Top1 + Bottom1 , Top2 + Bottom2) - y$

Capture safety zone boundary breach - Modified

COMPLETED



Completion of the project

- ✓ Detect safety zone Exiting (screenshots) - ***Completed***

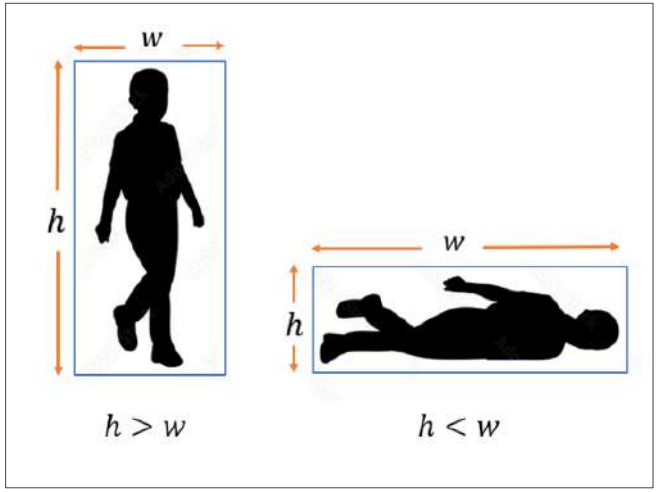
100% - completed

Sub Objectives 3

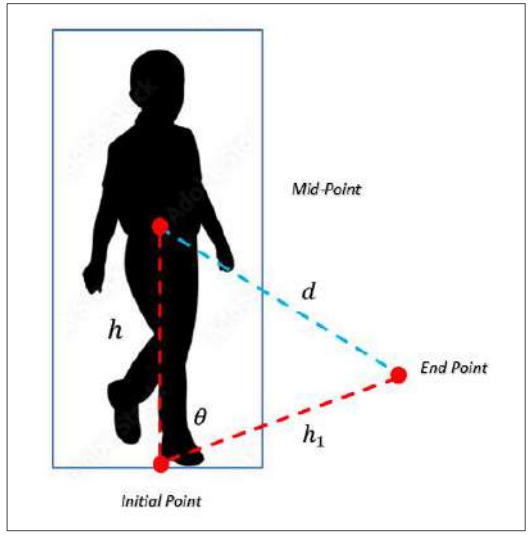
Fall Detection

Child Fall Detection

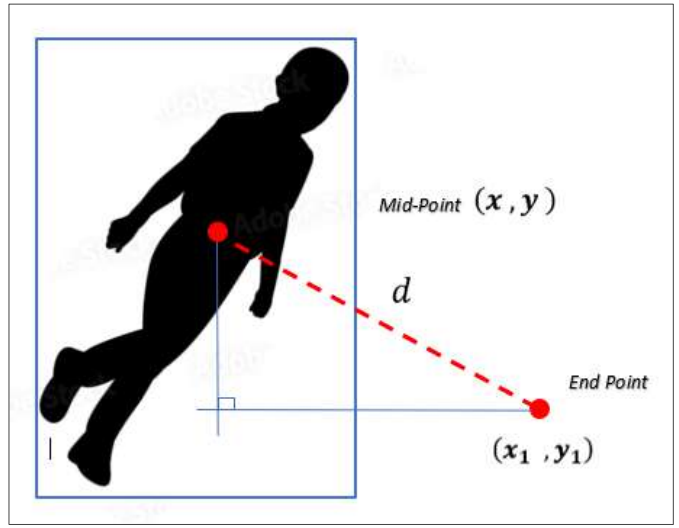
COMPLETED



Bound box orientation



Calculating the falling angle



Calculating the falling speed

h = *Mid-point – Initial Point*

h_1 = *End Point – Initial Point*

t = *threshold*

$$d^2 = h^2 + {h_1}^2 - 2hh_1 \cos\theta$$

$$2hh_1 \cos\theta = h^2 + h_1^2 - d^2$$

$$\cos\theta = \frac{h^2 + h_1^2 - d^2}{2\,h\,h_1}$$

$$\theta = \cos^{-1}\left(\frac{h^2 + h_1^2 - d}{2\,h\,h_1}\right)$$

$$if\ \theta < t$$

Mid Point $= (x\ ,\ y)$

End Point $= (x_1\ ,\ y_1)$

t = *adjustable time threshold*

$$distance = \sqrt{(x_1 - x)^2 + (y_1 - y)^2}$$

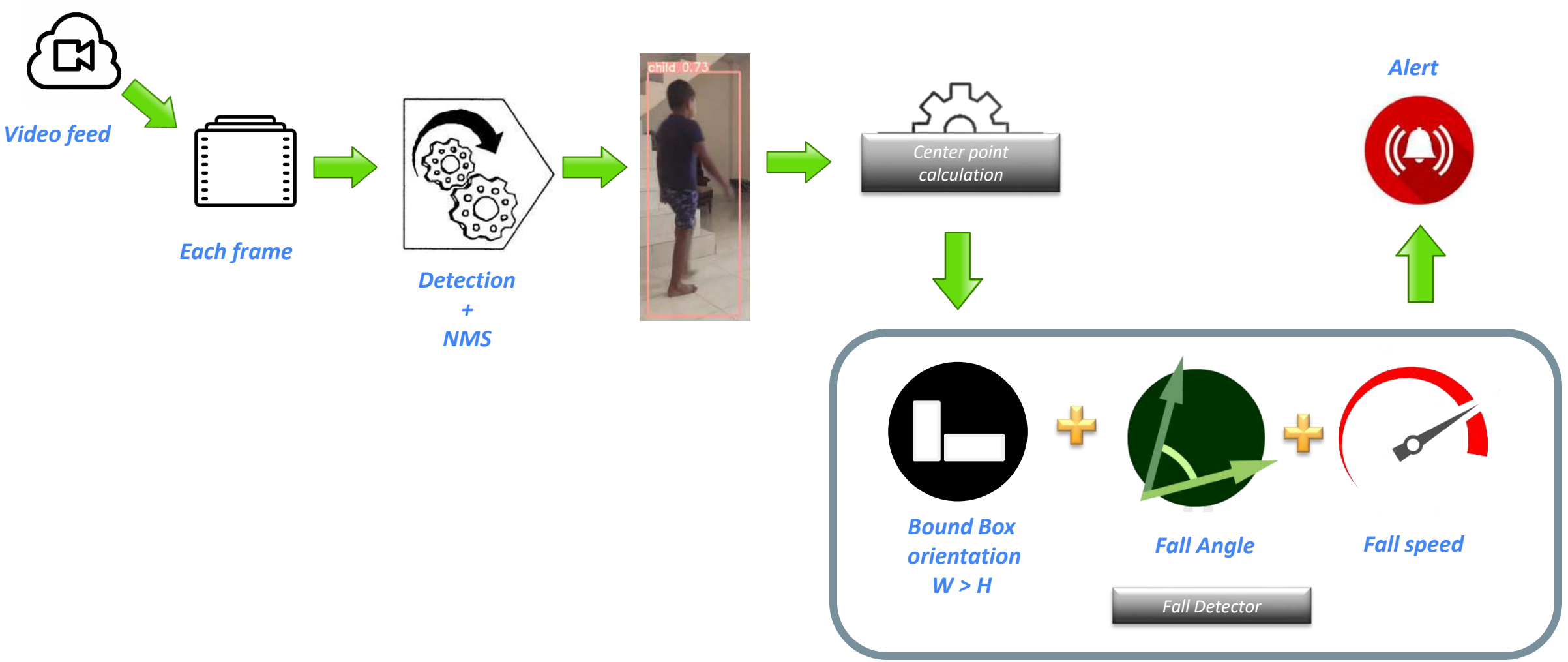
$$speed = \frac{distance}{t}$$

$$speed = \frac{\sqrt{(x_1 - x)^2 + (y_1 - y)^2}}{t}$$

<<Student ID>>| <<Student Name>> | <<Project ID>>

Overall Framework for Child Fall Detection

COMPLETED



<<Student ID>>| <<Student Name>> | <<Project ID>>

Completion of the project

COMPLETED

<<Student ID>>| <<Student Name>> | <<Project ID>>

✓100% - completed

Sub Objectives 4

Climb Detection

Child Climb Detection

COMPLETED

- **Furniture Object Detection**

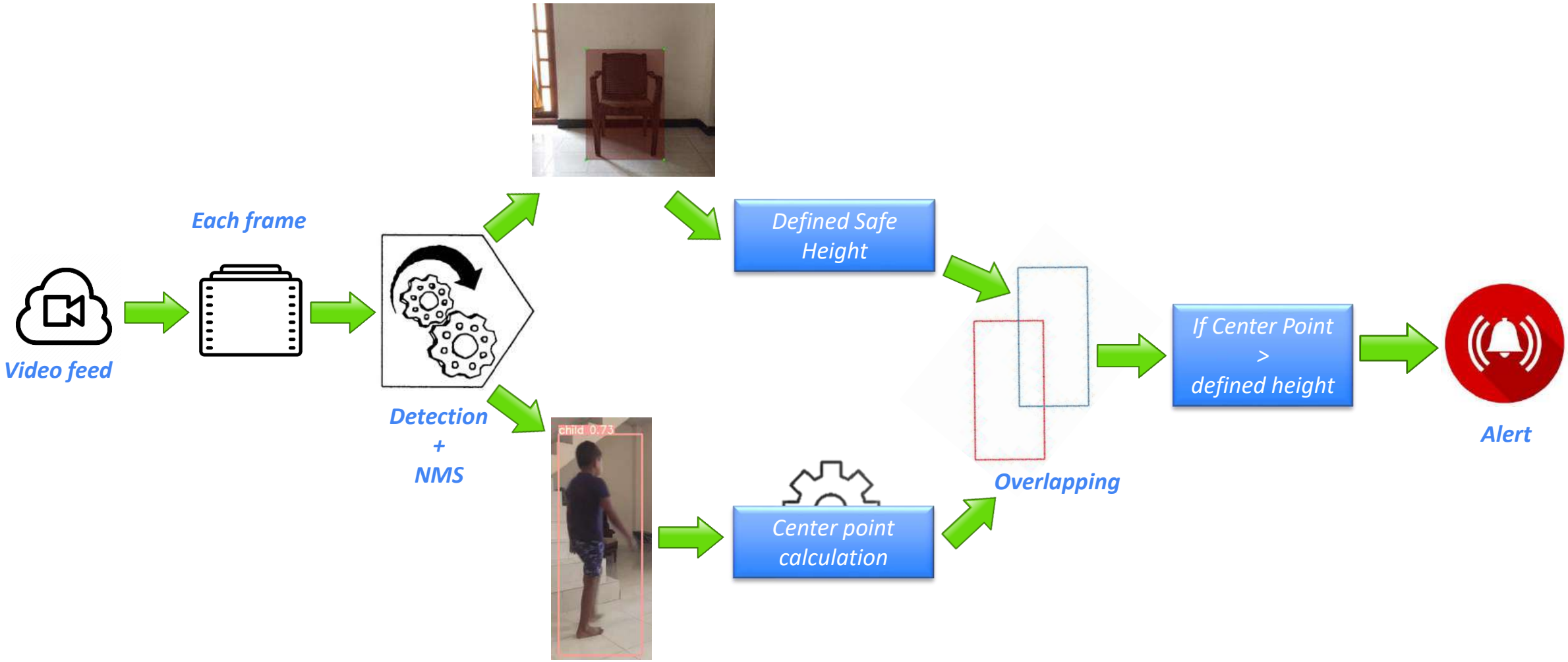
(a) table (b) sofa (c) chair

Matrices	YOLOv4	Tiny-YOLOv4
Mean Average Precision (mAP)	98.81%	98.88%
Precision	0.99	0.96
Recall	1.00	0.99
F1 Score	0.99	0.98

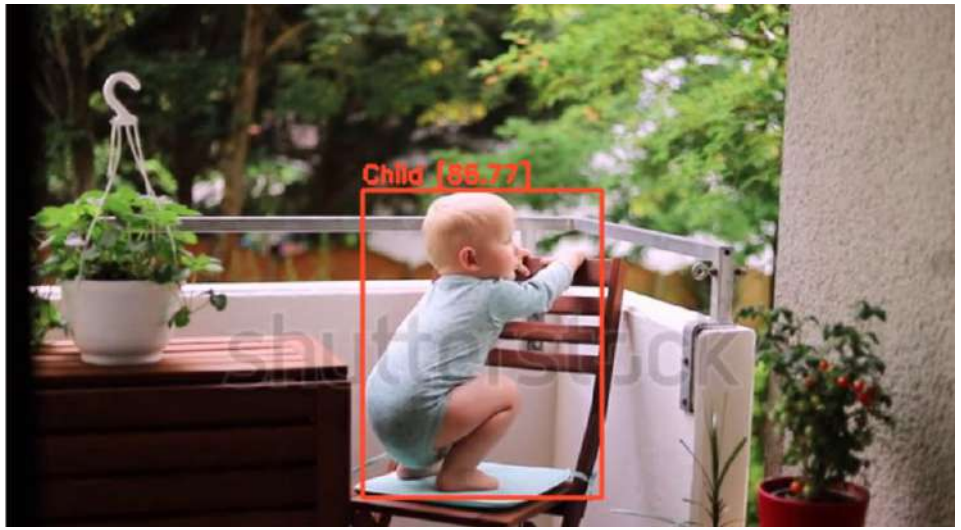
YOLOv4 was chosen as the best model after thorough comparison and evaluation.

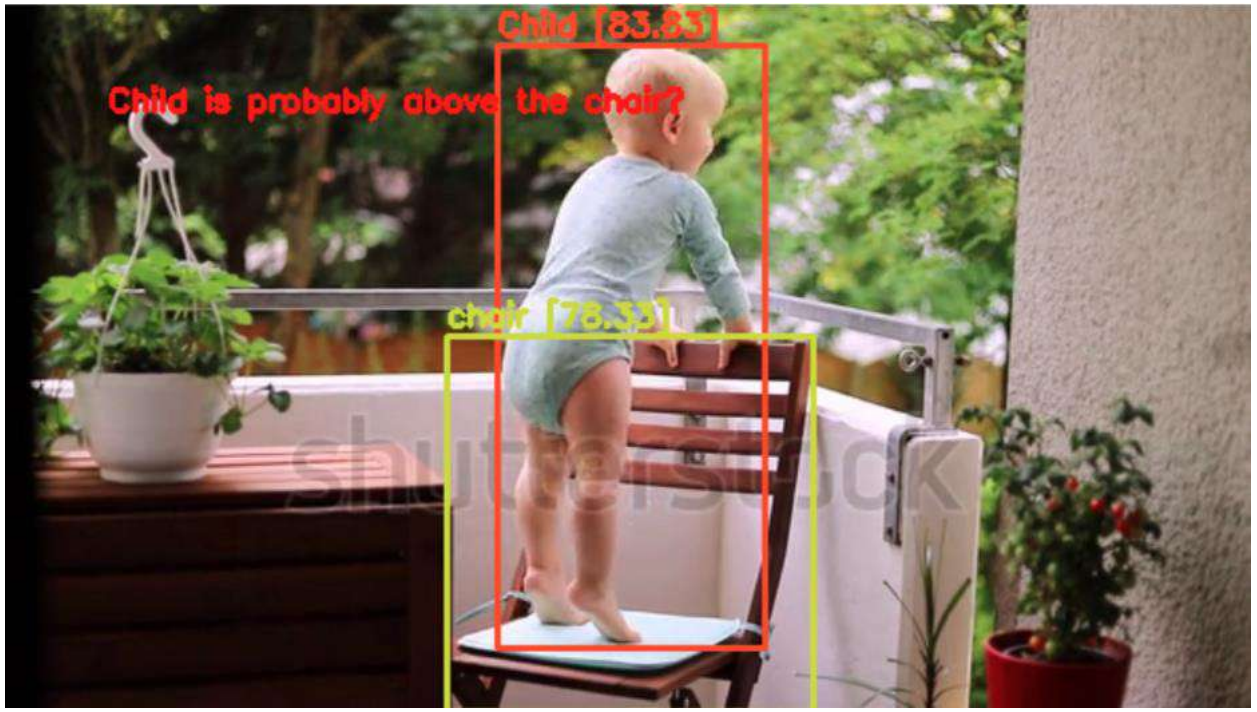
Overall Framework for Child Climb Detection

COMPLETED



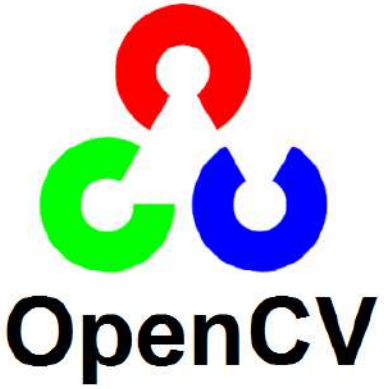
Completion of the project





If the location of the camera is changed, the result of the child climbing the furniture may not be obtained correctly. Considering this, the solution was implemented assuming the camera is in a fixed location.

Technologies



To do....

- ✓ Alert Management - WIP

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

Data Science

Background

- In today childcare has become a daily challenge for many families since parents cannot continuously monitor their babies' conditions either in normal or abnormal situations. Because of this reason, babies are more likely to be exposed to various hazards nowadays.

- **One of them is the damage caused to Children by touching unsafe electrical appliances.**

- The parent can prevent this by monitoring the baby’s behavior using a cctv camera, but there is no immediate way to notify the parent in the event of an accident and it can be difficult for the parent to keep an eye on the cctv camera screen continuously.

- This component aims to address this issue by introducing an IoT based real time baby behavior detection and monitoring system.

Research Problem

- 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 cause of unsafe electrical appliances sustained at home.

The question is ,

- How to prevent the child from electrical items before injury happening?
- How to keep track child activities.?
- How to implement Electrical injuries preventing assistance actions?

Objectives

What is the main Objective ?

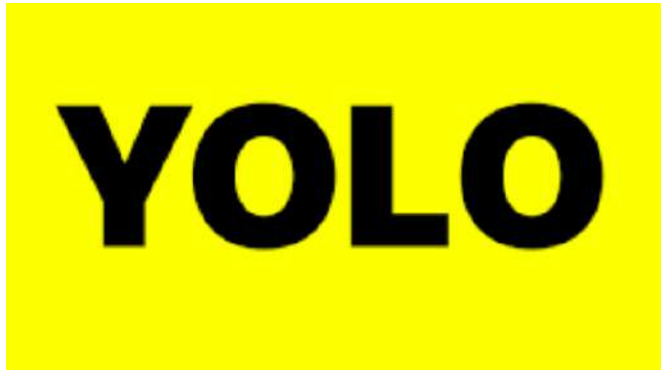
- The main objective is to ensure that keep track child activities which are in the early development stage are safe from electrical injuries due to unsafe electrical appliances, by applying surveillance cameras integrated with computer vision.

Objectives

- **What are the specific objectives ?**
  - ✓ Electrical Object Detection.
  - ✓ person detection
  - ✓ Child detection (Adult child classification)
  - ✓ Child and electrical object Proximity Calculation
  - ✓ Proper warning (Injury preventive action)

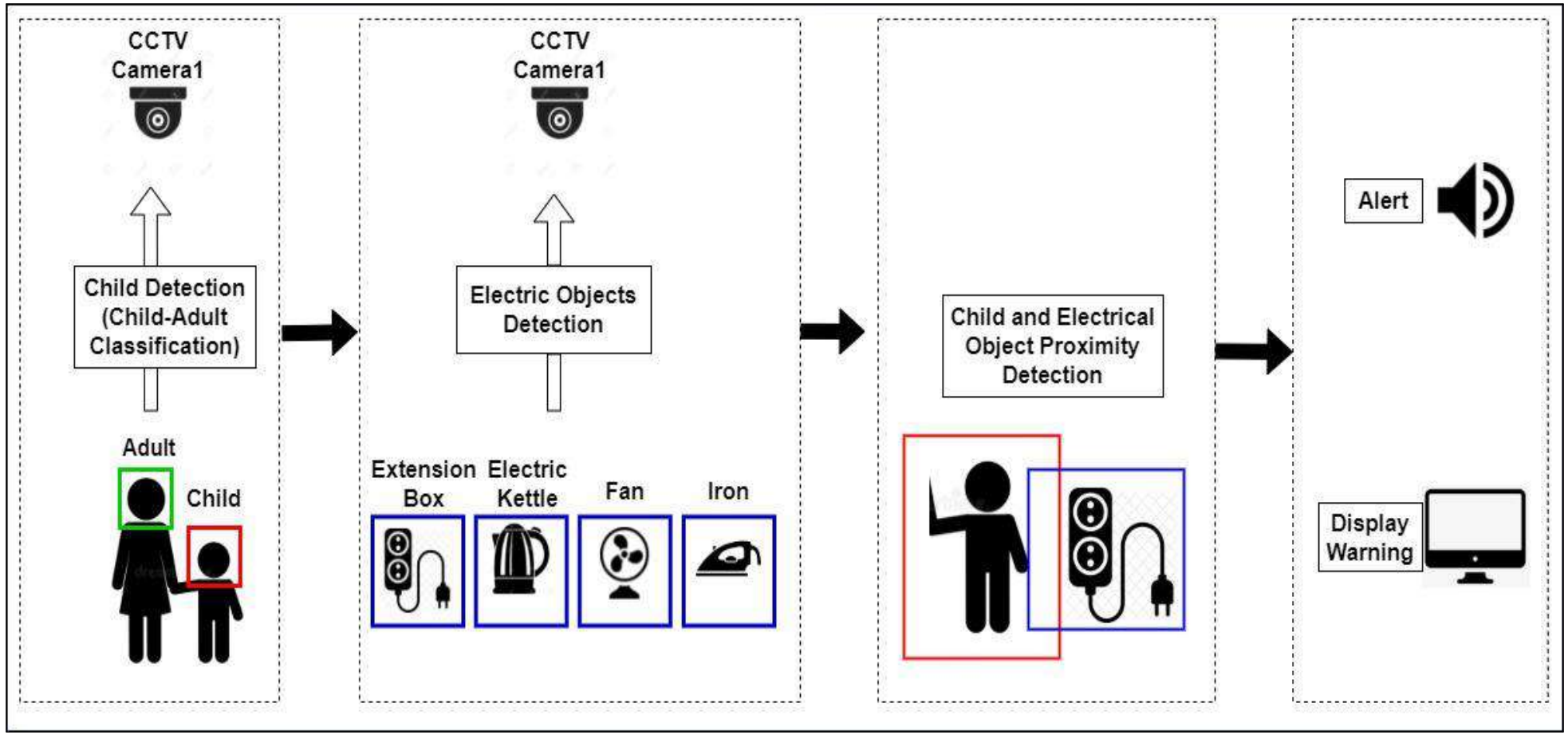
Tools, Technologies and algorithms used

- OpenCV, Keras, NumPy, YOLO, TensorFlow, Python



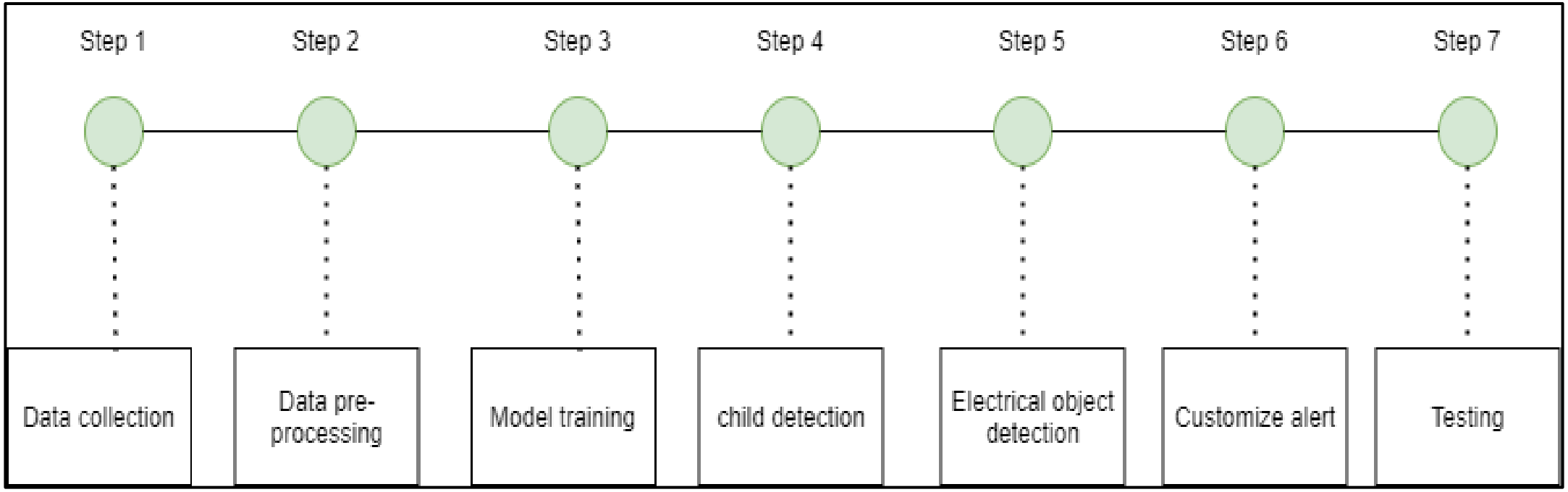
Methodology

- System Architecture



Methodology

- Road map



Methodology

Data Collection

Dataset	Sources
Child and adult images	Kaggle dataset
Electrical objects images (Fan, Electrical kettle, Iron, Electrical extension box)	Custom images and google images

- The Dataset consists with two types of images.

Methodology

Data pre-processing

1. Dataset resizing.
2. Dataset normalization.
3. Dataset annotation.

LabelImg

Annotation

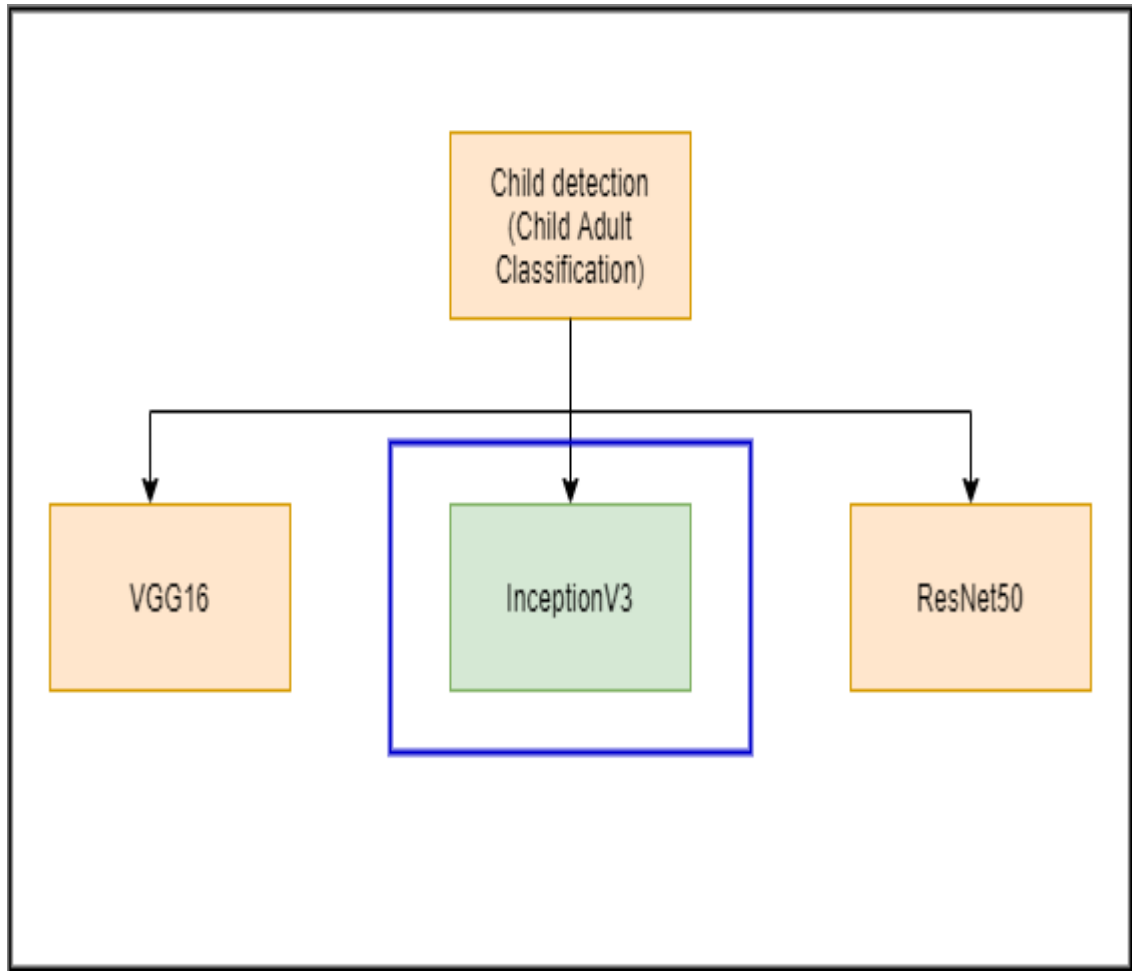
Normalization

```
child_training_X = np.array(child_training_X)
child_training_X = child_training_X.astype('float32')
child_training_X /= 255

nonchild_training_X = np.array(nonchild_training_X)
nonchild_training_X = nonchild_training_X.astype('float32')
nonchild_training_X /= 255
```

Methodology

Child, Adult Classification Model Selection.

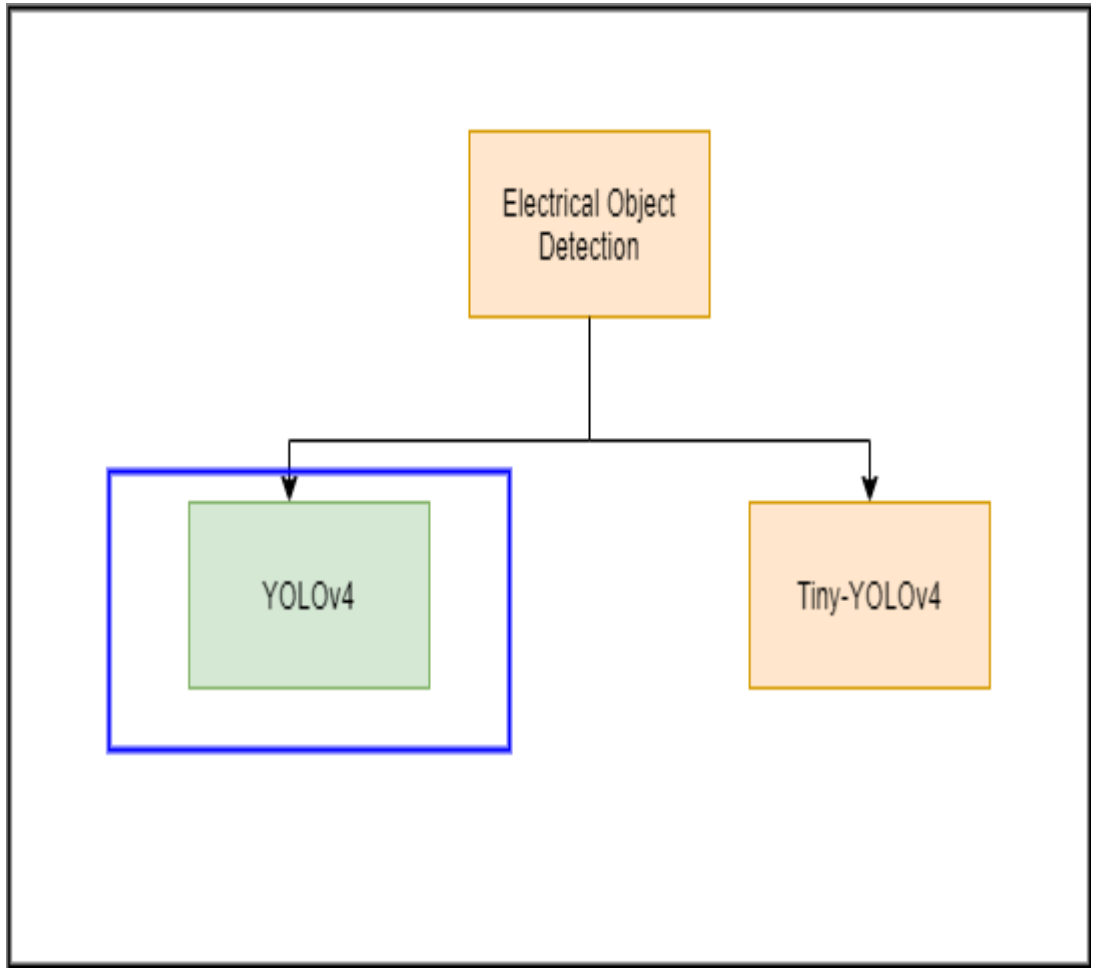
- VGG16, ResNet50 and InceptionV3 models have been tested.
- Number of Training Objects: 3082
- Number of Testing Objects: 771
- Optimizer : Adam Optimizer
- Activation function: Sigmoid, Relu



Methodology

Electrical Object Detection Model Selection.

- YOLOv4
- Tiny-YOLOv4



Development

Process

- **Step 1-** Check if there is a person or no one in the room.
- **Step 2-** If only person has been detected execute the adult child classification as well as electrical object detection in order to track hazardous child activities.

```
cascPath = '../drive/MyDrive/haarcascade_frontalface_default.xml'

def face_detection(frame):
    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)

    return faces
```

Development

- **Step 3-** Calculate proximity between person and electrical object.
- **Step 4-** If the detected person is child and his/her bounding box is overlap with the electrical object bounding box then only trigger a proper alert.
- **Step 5-** if the detected person is adult there will be no trigger an alert.

Bounding boxes overlapping

```
# Calculate the intersection between two bounding boxes
def intersection(a,b):
  x = max(a[0], b[0])
  y = max(a[1], b[1])
  w = min(a[0]+a[2], b[0]+b[2]) - x
  h = min(a[1]+a[3], b[1]+b[3]) - y
  if w<0 or h<0: return False # or (0,0,0,0) ?
  return True
```

Results

Child adult classification model training results

- VGG16 model: 52.65% accuracy
- ResNet model: 92.99% accuracy
- InceptionV3 model: 94.42% accuracy

Highest accuracy obtained by inceptionv3 model.

```
Accuracy Score: 94.42282749675745
Precision Score: 95.03722084367246
Recall Scroe: 94.33497536945814
F1 Score: 94.6847960444994
Confusion Matrix
[[345  20]
 [ 23 383]]
Classification Report
              precision    recall  f1-score   support

       child       0.94      0.95      0.94       365
   non-child       0.95      0.94      0.95       406

    accuracy                           0.94       771
   macro avg       0.94      0.94      0.94       771
weighted avg       0.94      0.94      0.94       771
```

Results

- **Electrical object detection model training results**

Matrices	YOLOv4	Tiny-YOLOv4
Accuracy	81.77%	80.00%
Precision	81%	80%
Recall	85%	77%
F1 Score	83%	81%

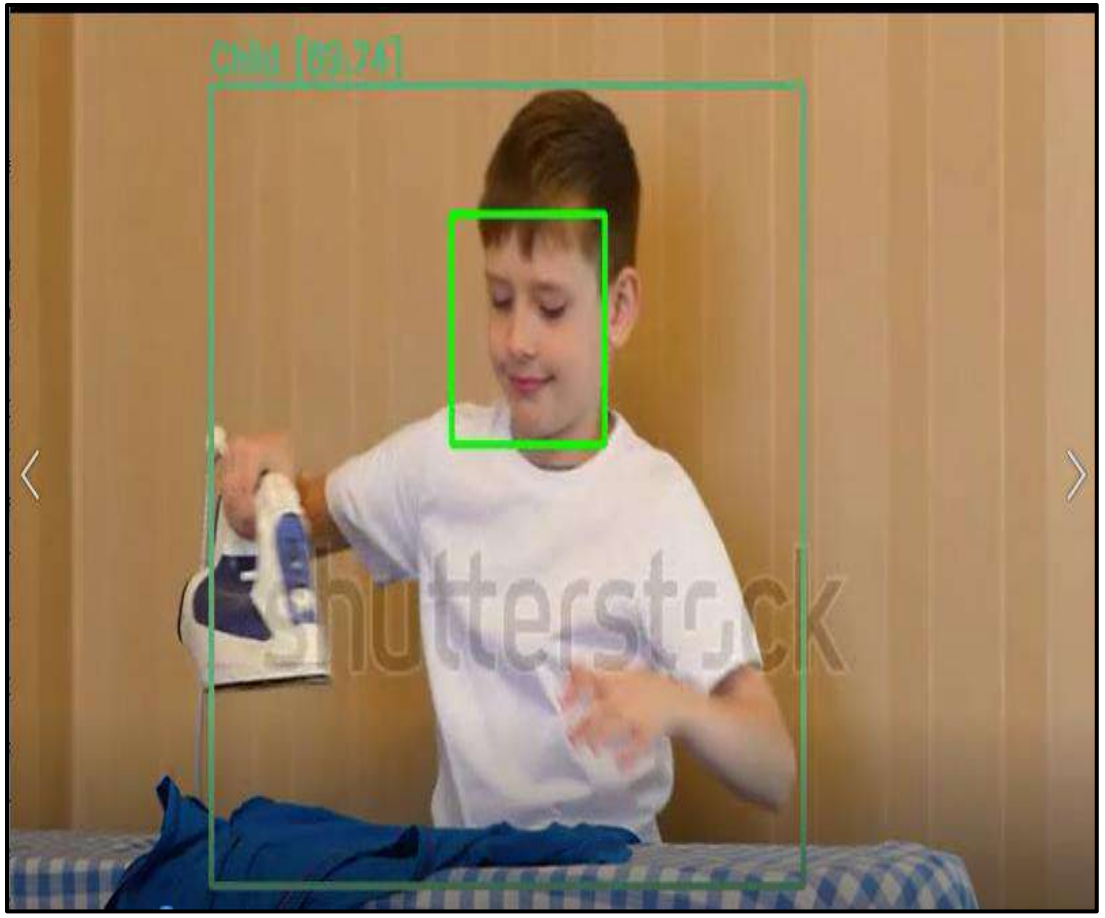
There are overall 4 different classes have been used.

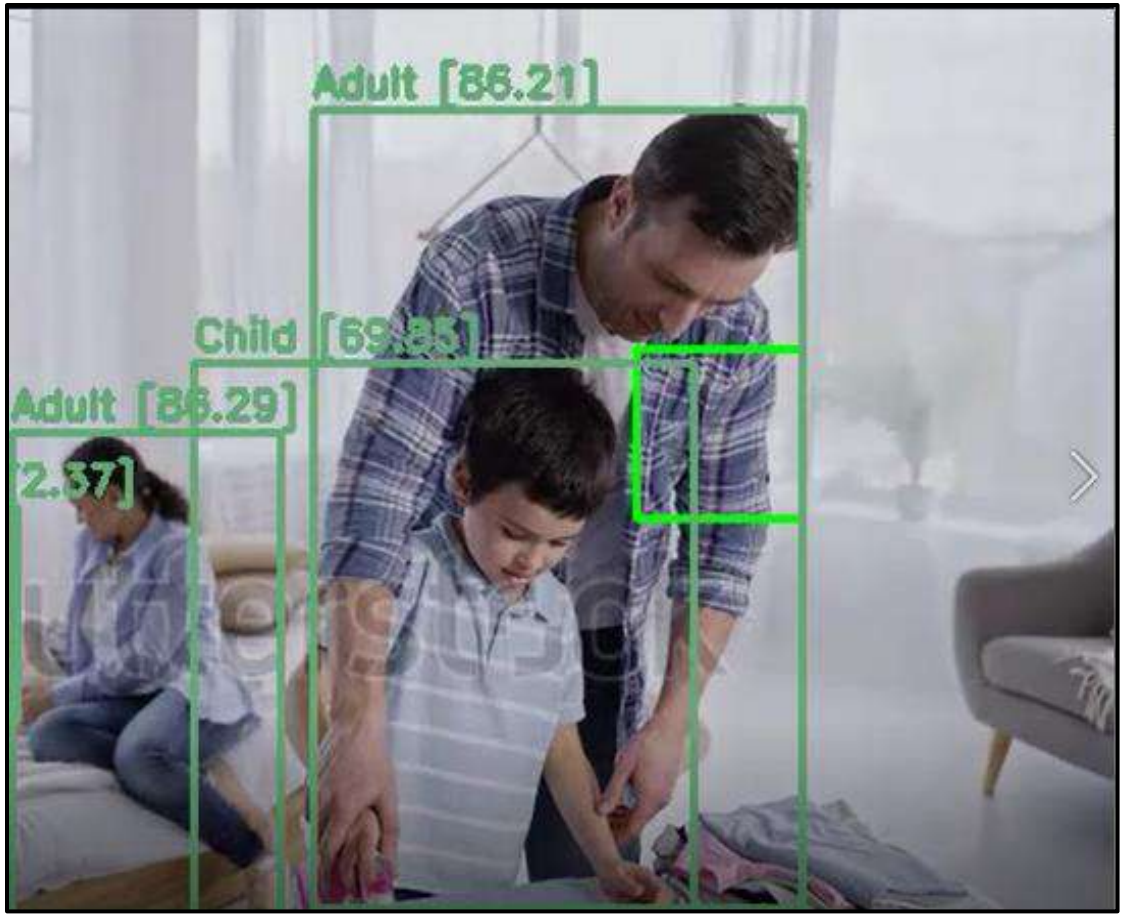
- Electrical kettle.
- Fan
- Iron
- Electrical extension box.

Highest accuracy obtained by YOLOv4 model.

Results

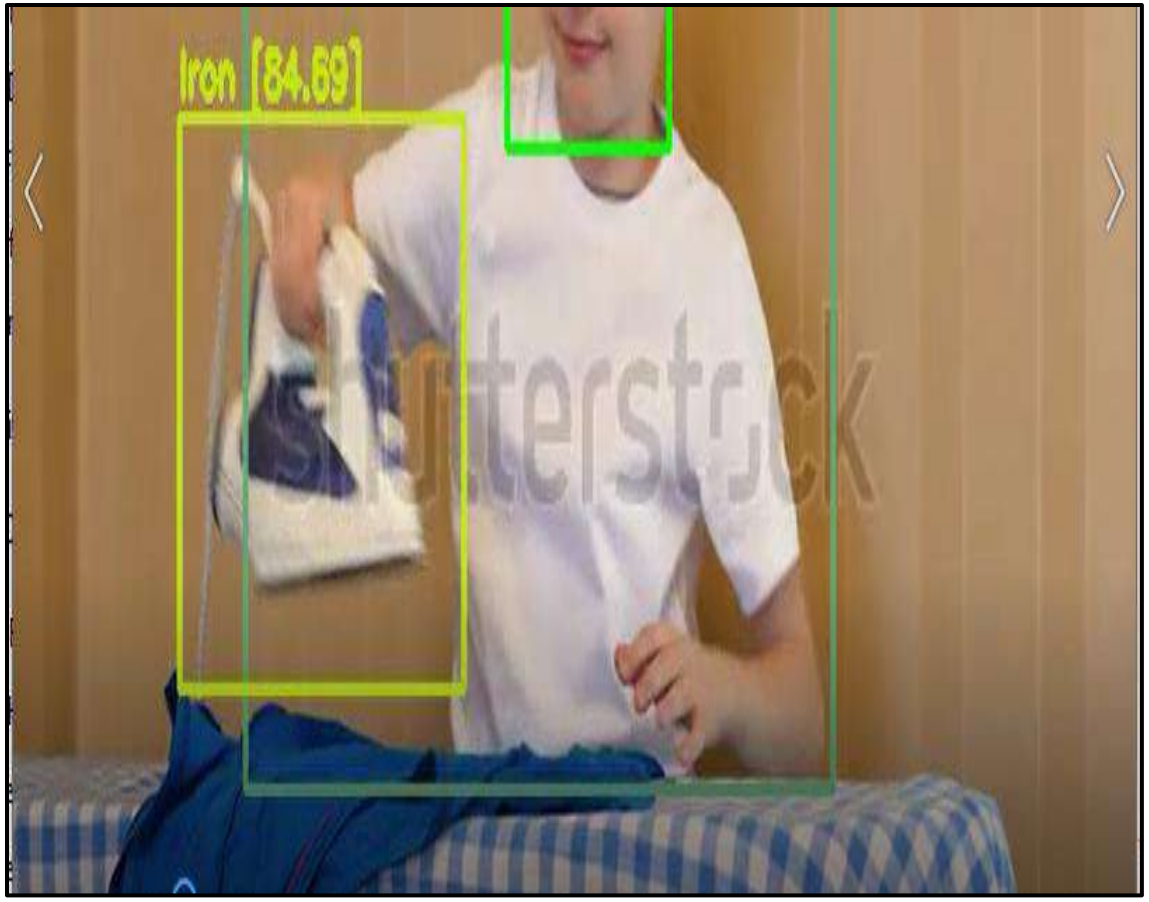
Test Results Child Detection





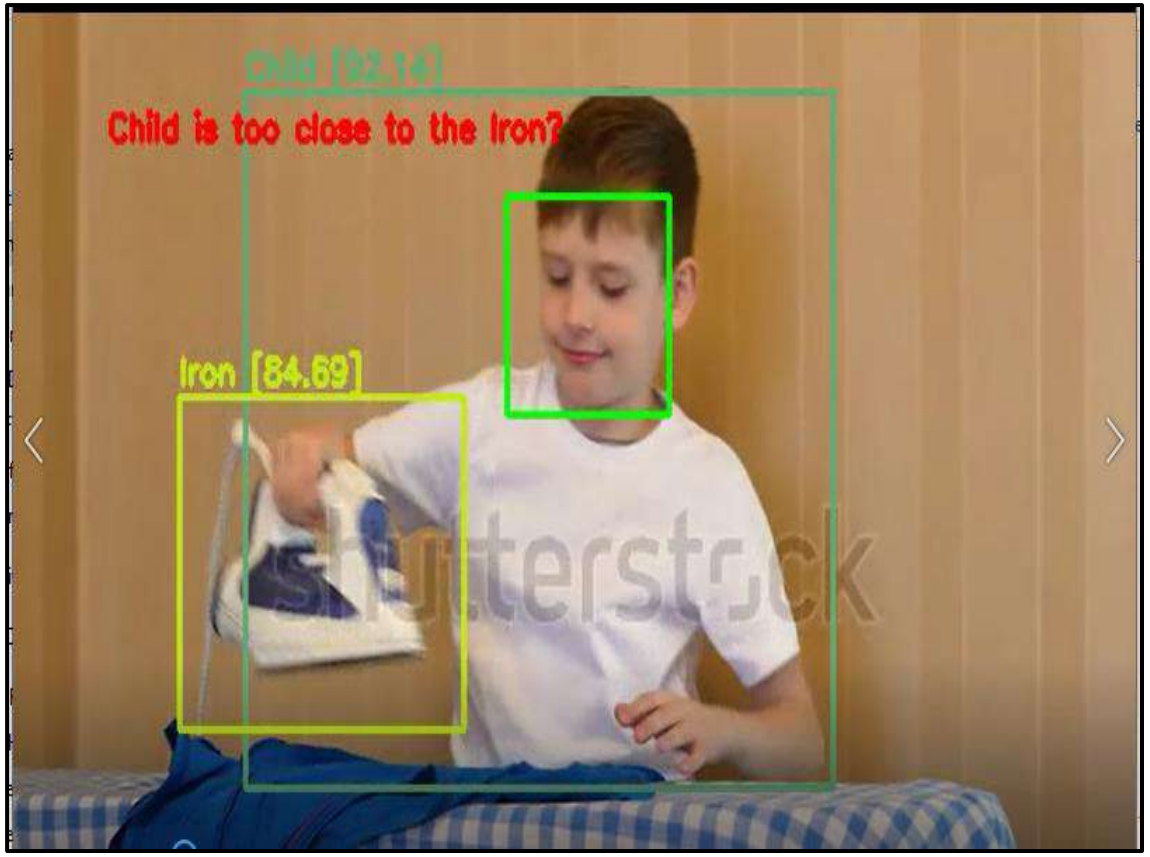
Results

- **Test Results Electrical Object Detection**



Results

- **Test results Overall System**



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