Project 2: Hazardous Behavior and Condition Alarm System

Where did this idea come from?
I came up with the idea for this project because, in my last manufacturing engineering internship, I was responsible for writing programs to automate the production line. But due to irregularities by the production line employees, it caused a huge problem for my work. One of my goals is to minimize the losses caused by the irregularities, and ultimately improve production safety and company profit profitability.

"SafeGuard Pro" —— A smart hazard detection and alarm system
A dependable system for detecting hazardous behaviors and unsafe conditions in a real-time manner is necessary to prevent accidents and ensure compliance with safety regulations.

The product's users include workshop managers, safety officers and industrial business owners who are responsible for maintaining a safe working environment. They need reliable systems to monitor and respond to safety risks in real time.

Main purpose:

1. Create an intelligent real-time dangerous behavior and condition detection system for the industrial company.
2. Ensure worker safety and prevent accidents using artificial intelligence and sensor technology.
3. Provide actionable alerts that enable safety officers or shift managers to take immediate corrective action.

User Stories：

Starting from the customer's needs
As a safety officer, I need to receive real-time alerts and detailed reports of hazardous behaviors and conditions on the production line so that immediate corrective action can be taken to prevent accidents.

1. Real-time hazard detection and alerts
As a safety officer, I need to receive real-time alerts when the system detects dangerous behavior or unsafe conditions so that corrective action can be taken immediately to prevent accidents. For example, in an automated production line, an employee breaks into the area where a robotic arm is running, which can cause great harm to people and equipment.

Alerts need to be received immediately. I would like my product to be able to do this by sending an alert to the safety officer while also signaling an alarm at the scene. So that employees can realize their wrongdoings and stop the accident immediately. It can also be used as a record of the company's employee rewards and punishments for future production safety education.

2. Behavior Monitoring

As a line manager, I need to systematically monitor worker behavior and detect unsafe acts and SOP violations. For example, in a clean manufacturing plant, an employee removes his or her dust suit or mask. Such behavior can cause damage to high-precision equipment and degrade the quality of the product. There may also be employees jumping steps and violating operation requirements, stealing company and customer confidential documents or products, such as taking photos and samples on the production line.

3.Environmental Condition Monitoring

As a safety officer, I would require systems to detect unsafe environmental conditions, such as gas leaks and fires. This ensures a safe working environment. As a company supervisor, the temperature and humidity in the production location must be controlled in the right zones to guarantee the quality of the products.

4.Automated Reporting and Log Grading

As a safety supervisor, the system is expected to automatically log all detected hazardous conditions. These hazardous conditions are to be graded according to the degree of urgency:

Level A situations: Actions that will cause significant damage to the enterprise or accidents that will result in injuries or fatalities

Level B situations: loss of control of environmental temperature and humidity, employee violation of operating procedures

Level C situations: Employee not dressed appropriately, sleeping while on the production line

5. Mobile and Web Alerts

Receive real-time alerts via mobile apps so you can stay informed even when you're not on the shop floor.

6.Customizable alarm thresholds

Ability to customize the thresholds that trigger alarms (e.g. noise levels, machine operating speeds) so that the system can meet specific safety requirements.

7.Actionable insights and recommendations

The ability to provide actionable recommendations based on detected hazards so that the root cause of the problem can be addressed, preventing similar issues from occurring in the future.

8.Multi-User Access

As the head of the company, it was expected that multiple team members (managers, safety officers) would all have access to the system with different permissions so that we could collaborate on safety monitoring and response.

MVP

For the MVP, focus on delivering essential functionalities that will provide immediate value while laying the foundation for future enhancements.
First Priorities:

1. Real-Time Hazard Detection and Alerting (High Priority)

This is the core functionality of the product. Without real-time alerts, the system won't deliver its main value.

2. Behavioral Monitoring (High Priority)

Detecting unsafe worker behavior is critical to preventing accidents before they happen.

3. Environmental Condition Monitoring (High Priority)

Detecting unsafe environmental conditions adds another layer of accident prevention.

4. Mobile and Web Alerts (Medium Priority)

Ensures that safety officers and managers can respond quickly, even if they're not physically present.

5. Automated Reporting and Log Grading (Medium Priority)

This feature provides immediate value by ensuring a record of all incidents for compliance and future analysis.

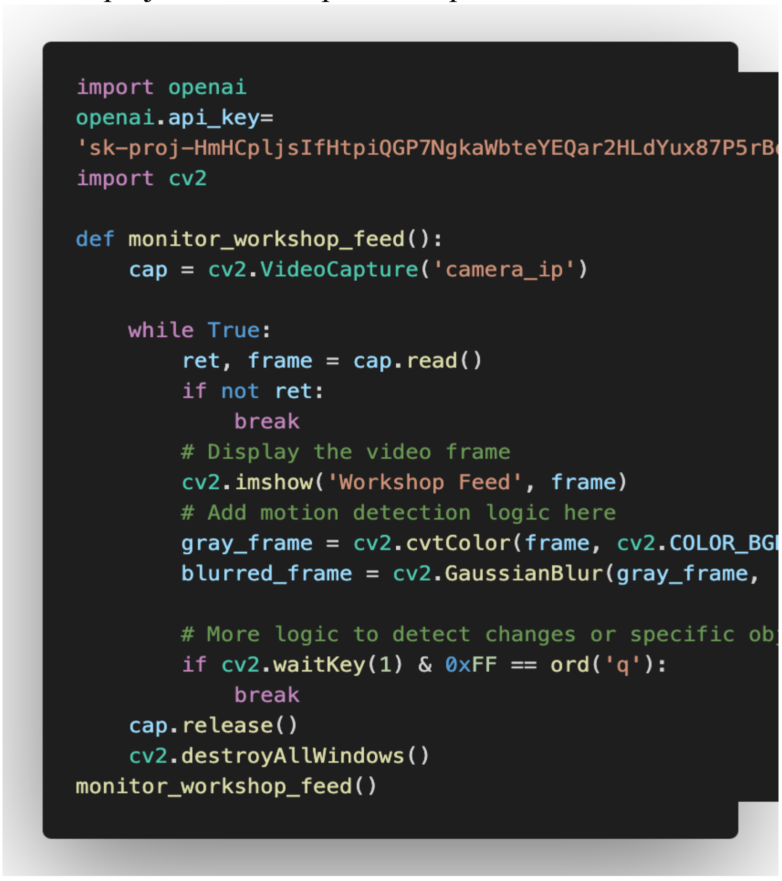
Secondary Priorities:

6. Customizable Alert Thresholds
7. Actionable Insights and Recommendations
8. Multi-User Access

1.The Card:
"As a safety officer, I need to receive real-time alerts for unsafe conditions."
2.The Conversation:
Clarify what constitutes "unsafe conditions," delivery mechanisms for alerts, and expected reaction time.
3.The Details:
Define how to implement the detection system (AI/sensors), APIs for triggering alerts, and integration with frontend systems (web and mobile).
4.Acceptance Criteria:
Real-time alerts, alert customization, handling of false positives, and logs of past alerts.

OpenAI API to realize function

1.Behavioral Monitoring Using API:
Cameras installed in the workshop can capture real-time footage of workers and equipment, CCTV cameras with IP can help to locate the position of different production lines. This project can use OpenCV to process the video streams information.



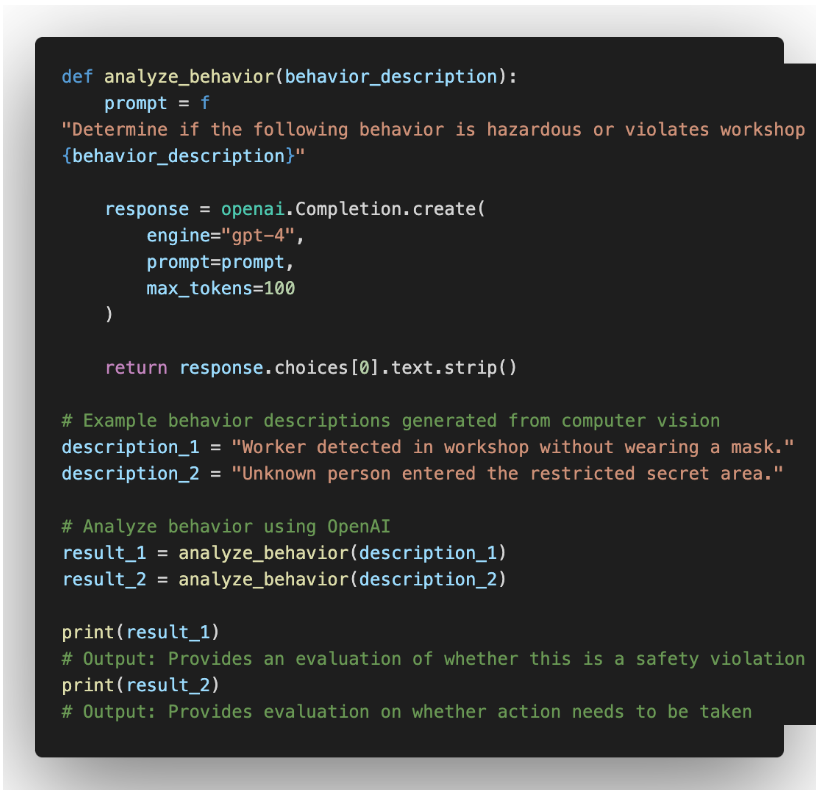
```python
import openai
openai.api_key=
'sk-proj-HmHCpljsIfHtpiQGP7NgkaWbteYEQar2HLdYux87P5rB
import cv2

def monitor_workshop_feed():
    cap = cv2.VideoCapture('camera_ip')

    while True:
        ret, frame = cap.read()
        if not ret:
            break
        # Display the video frame
        cv2.imshow('Workshop Feed', frame)
        # Add motion detection logic here
        gray_frame = cv2.cvtColor(frame, cv2.COLOR_BG
        blurred_frame = cv2.GaussianBlur(gray_frame,

        # More logic to detect changes or specific ob
        if cv2.waitKey(1) & 0xFF == ord('q'):
            break
    cap.release()
    cv2.destroyAllWindows()
monitor_workshop_feed()
```

Figure1. codes for importing camera streams

Then, use OpenAI API to detect the behaviors of the workers. The OpenAI API can determine if the behavior captured is hazardous or not. This project needs to train the system to recognize a wide variety of behaviors.

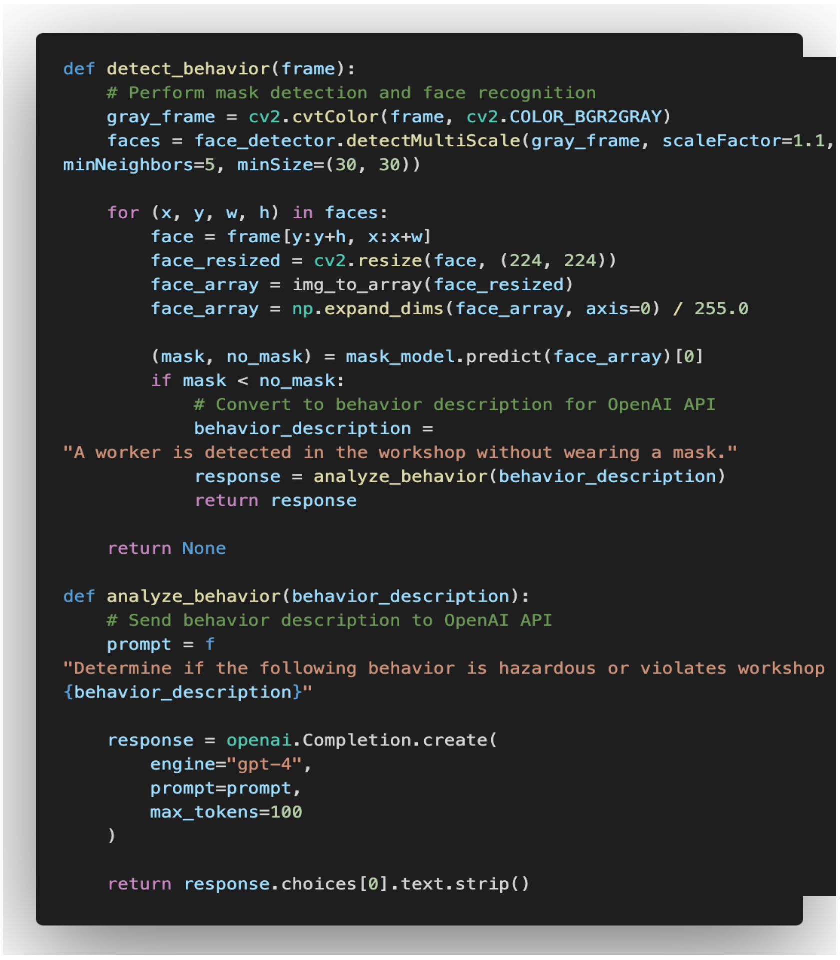
For example, set up a code function to detect whether workers all wear masks:

1. Detect faces in the camera feed.
2. Classify if each face is wearing a mask or not.
3. Raise an alert if a worker is detected without a mask.



```python
def analyze_behavior(behavior_description):
    prompt = f
"Determine if the following behavior is hazardous or violates workshop
{behavior_description}"

    response = openai.Completion.create(
        engine="gpt-4",
        prompt=prompt,
        max_tokens=100
    )

    return response.choices[0].text.strip()

# Example behavior descriptions generated from computer vision
description_1 = "Worker detected in workshop without wearing a mask."
description_2 = "Unknown person entered the restricted secret area."

# Analyze behavior using OpenAI
result_1 = analyze_behavior(description_1)
result_2 = analyze_behavior(description_2)

print(result_1)
# Output: Provides an evaluation of whether this is a safety violation
print(result_2)
# Output: Provides evaluation on whether action needs to be taken
```

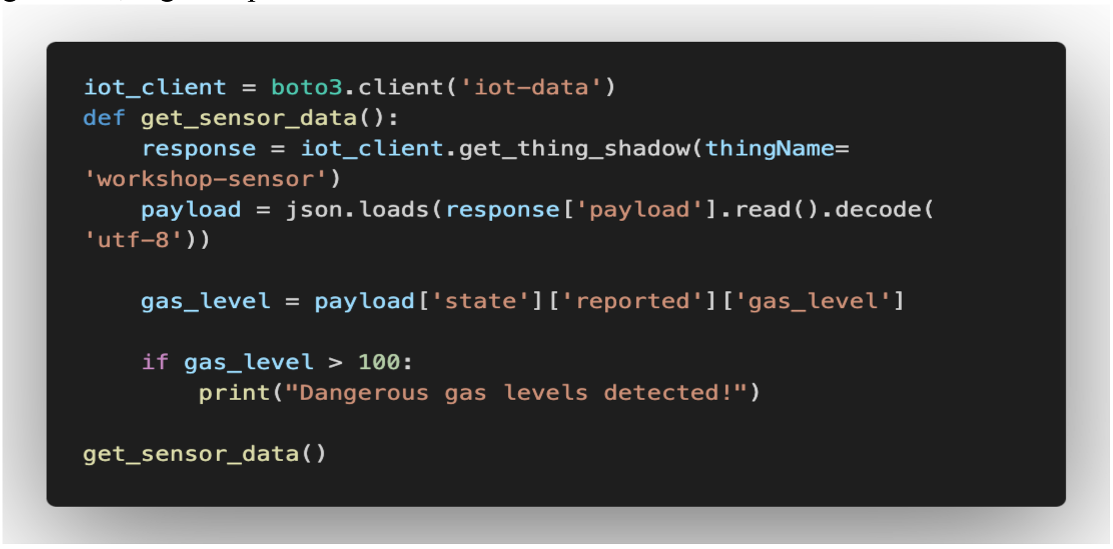
Figure 2. Use API to analyze behavior



```python
def detect_behavior(frame):
    # Perform mask detection and face recognition
    gray_frame = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    faces = face_detector.detectMultiScale(gray_frame, scaleFactor=1.1,
minNeighbors=5, minSize=(30, 30))

    for (x, y, w, h) in faces:
        face = frame[y:y+h, x:x+w]
        face_resized = cv2.resize(face, (224, 224))
        face_array = img_to_array(face_resized)
        face_array = np.expand_dims(face_array, axis=0) / 255.0

        (mask, no_mask) = mask_model.predict(face_array)[0]
        if mask < no_mask:
            # Convert to behavior description for OpenAI API
            behavior_description =
"A worker is detected in the workshop without wearing a mask."
            response = analyze_behavior(behavior_description)
            return response

    return None

def analyze_behavior(behavior_description):
    # Send behavior description to OpenAI API
    prompt = f
"Determine if the following behavior is hazardous or violates workshop
{behavior_description}"

    response = openai.Completion.create(
        engine="gpt-4",
        prompt=prompt,
        max_tokens=100
    )

    return response.choices[0].text.strip()
```

Figure 3. The main codes for mask detection

2.Enviornment Monitoring Using API:
Use sensors connected with a cloud platform to monitor environmental conditions such as temperature or humidity. This function can ensure track dangerous conditions like gas leaks, high temperatures or excessive noise levels.

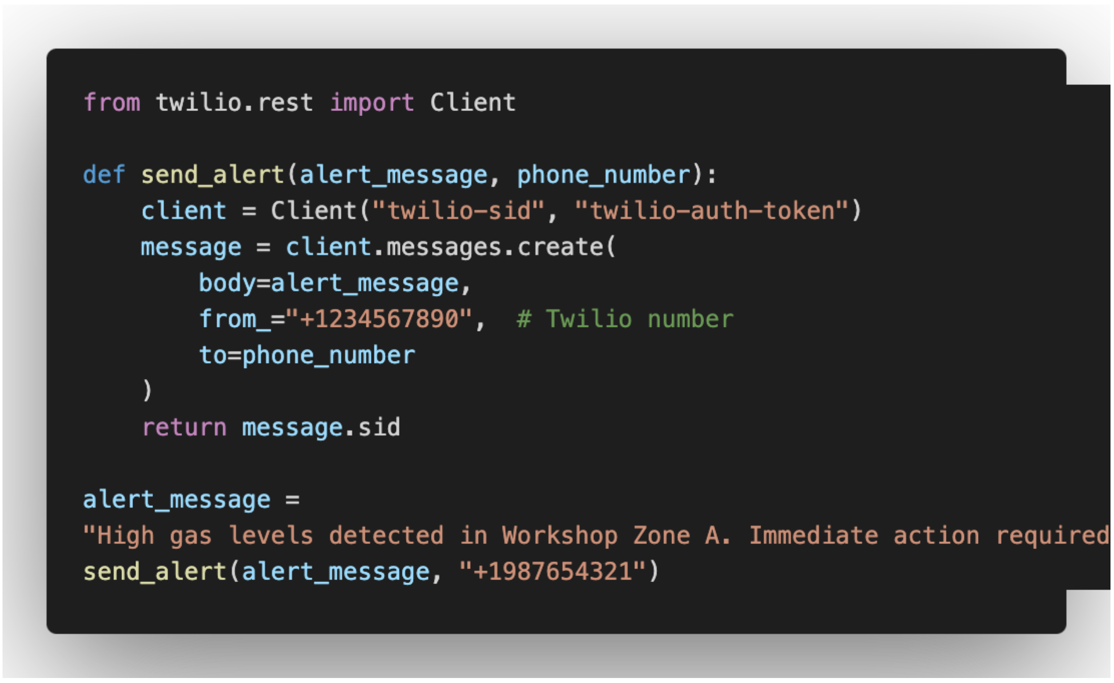


```python
iot_client = boto3.client('iot-data')
def get_sensor_data():
    response = iot_client.get_thing_shadow(thingName=
'workshop-sensor')
    payload = json.loads(response['payload'].read().decode(
'utf-8'))

    gas_level = payload['state']['reported']['gas_level']

    if gas_level > 100:
        print("Dangerous gas levels detected!")

get_sensor_data()
```

Figure 4. use AWS IoT to monitor environment

According to different requirements, safety officer can set different thresholds for environmental conditions like temperature or noise levels.

3.Real-Time Alerts Using APIs
Once a hazardous condition is detected, an SMS alert is sent to responsible personnel.
1.Twilio API for sending SMS or email notifications.
2.Firebase for push notifications.

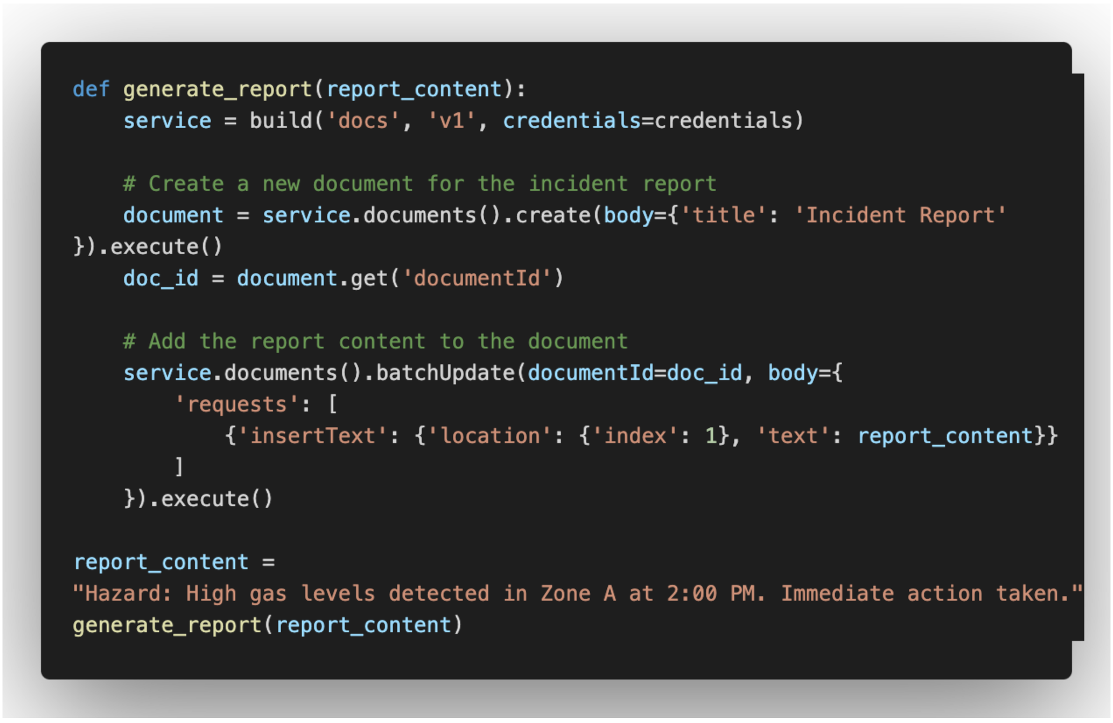


```python
from twilio.rest import Client

def send_alert(alert_message, phone_number):
    client = Client("twilio-sid", "twilio-auth-token")
    message = client.messages.create(
        body=alert_message,
        from_="+1234567890",  # Twilio number
        to=phone_number
    )
    return message.sid

alert_message =
"High gas levels detected in Workshop Zone A. Immediate action required
send_alert(alert_message, "+1987654321")
```

Figure 5. Send SMS message to responsible person

4. Generating an Incident Report Using Google Docs API

Save all detected hazards into a database for later audits and automatically generate an incident report for safety documentation.



```
def generate_report(report_content):
    service = build('docs', 'v1', credentials=credentials)

    # Create a new document for the incident report
    document = service.documents().create(body={'title': 'Incident Report'
}).execute()
    doc_id = document.get('documentId')

    # Add the report content to the document
    service.documents().batchUpdate(documentId=doc_id, body={
        'requests': [
            {'insertText': {'location': {'index': 1}, 'text': report_content}}
        ]
    }).execute()

report_content =
"Hazard: High gas levels detected in Zone A at 2:00 PM. Immediate action taken."
generate_report(report_content)
```

Figure 6. codes to create incident report

Conclusion

The Safety Detection and Alarm System uses OpenCV for video analysis and OpenAI for behavior classification. It is feasible to create a powerful workshop safety monitoring system that detects dangerous behaviors and environmental conditions, sends alerts and logs incidents. This protects people and equipment. In the future, it will be also and continuously improved to fulfill the user's needs.