Internship Report on

Advanced Object Detection using YOLO for Diverse Applications

Submitted in partial fulfilment of the requirement for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE & ENGINEERING

Institution: Graphic Era University

Date: 2 July - 14 August 2024

Submitted by: Anshika Singhal

Submitted to: Mr. Vipul Kumar Bharti

(Scientist ‘C’)

Acknowledgments

I would like to express my deepest gratitude to those who have supported me throughout my 6-week internship.

First and foremost, I would like to thank Mr. **Vipul Kumar Bharti (Scientist 'C'),** my mentor at **IRDE, DRDO**, for their invaluable guidance, insightful feedback, and constant encouragement throughout this project. Theirexpertise in object detection and computer vision was instrumental in the successful completion of this work.

I would also like to extend my appreciation to **Tarun Purohit**, who provided their support and assistance whenever needed. Their collaboration and shared knowledge were vital in overcoming the challenges faced during the development of the models.

Lastly, I would like to acknowledge the contributions of everyone for their help with the technical infrastructure, as well as my family and friends for their continuous moral support.

Thank you all for making this experience an invaluable learning opportunity.

Table of Contents

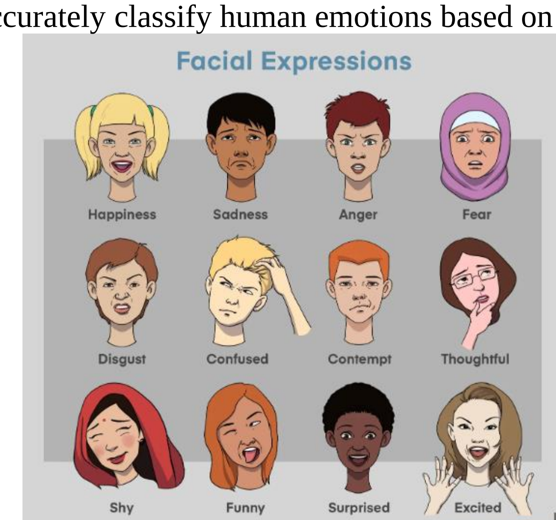
Abstract

Personal Protective Equipment

As seen in Petrochemical Industries, employees are to follow certain safety protocols. And the traditional method of monitoring and checking for any discrepancies in compliance to wear PPE by employees is old and matter of fact timeconsuming. There is always room for error in judgement and may ultimately lead tohurting employees. We propose a solution that will cut back on all these disadvantages and bring an automated system which will act as the gateway to safetyin any industry required. This will cut down any need to attend to wounds as it won'toccur in the first place, any disruption in workflow saving time and any need to employ separate personnel saving money.

Emotion detection using facial expressions

This project aims to develop a Emotion Detection using facial expression recognition with the help of Convolutional Neural Networks (CNN). The primary objective is to accurately classify human emotions based on their facial expressions into multiple classes, such as happy, sad, angry, neutral, surprised, etc. The proposed system holds potential applications in various fields, including market research, customer feedback analysis, and human computer interaction.



Introduction

Background

Object detection is a crucial domain within computer vision, offering practical solutions for various applications such as autonomous vehicles, security surveillance, and human-computer interaction. This process involves not just identifying but also locating objects within images or video frames. Traditionally, object detection relied on handcrafted features and classifiers, but recent advancements have largely shifted the focus to deep learning methods, which significantly enhance both accuracy and processing speed.

A major breakthrough in this field is the YOLO (You Only Look Once) algorithm. YOLO transformed object detection by addressing it as a single regression problem. Instead of the conventional multi-stage approaches, YOLO predicts bounding boxes and class probabilities simultaneously in one pass, which greatly boosts detection speed. This feature makes YOLO especially suitable for real-time applications.

Over time, YOLO has evolved through several versions, each improving upon the previous in terms of accuracy and efficiency. Its ability to maintain a balance between speed and precision has made YOLO a preferred choice for diverse applications, including those tackled in this internship: vehicle number plate recognition, emotion detection, and PPE detection. These examples illustrate YOLO's versatility and its capability to handle real-time processing in varied scenarios.

Personal Protective Equipment

We live in a world where uncertainty and unwarranted actions take place all the time. But we also live in a world where we know prevention is better than cure. This is the same in aspects like safety and health. Most industries/workplaces have some protocols that the workers have to follow to ensure safety. Especially in manufacturing industries where workers are always at risk of physical injuries. The protocols that they have to follow include wearing PPE(Personal Protective Equipment), cellphone violation, entering restricted areas and so on. It varies from one place to another. But most often than not workers often do not comply with these safety regulations and standards. There have been many cases where worker injuries have been reported but yet they seem not to care for their own safety. Most cases for non-compliance include comfort, improper communication, negligence. These problems have been there from when these industries have started but yet there is no solution to properly make sure these accidents never happen again. There have been solutions like manual surveillance with logs, where one person is appointed to make sure workers are adhering to rules and regulations. Then the solution went one step further with the invention of CCTV cameras, again where one person is incharge of overlooking all the monitors that have direct live feed of what was happening in the industries. For other reasons there have been scheduled training classes to educate workers of the safety regulations and why they have to wear PPE.

Why is PPE important?

Making the workplace safe includes providing instructions, procedures, training, and supervision to encourage people to work safely and responsibly. Even where engineering controls and safe systems of work have been applied, some hazards might remain. These include injuries to

- the lungs, e.g., from breathing in contaminated air;
- the head and feet, e.g., from falling materials;
- the eyes, e.g., from flying particles or splashes of corrosive liquids;
- the skin, e.g., from contact with corrosive materials;
- the body, e.g., from extremes of heat or cold.

PPE is needed in these cases to reduce the risk.

What should employers do ?

- Only use PPE as a last resort.
- If PPE is still needed after implementing other controls (and there will be circumstances when it is, e.g., head protection on most construction sites), they must provide this for their workers free of charge.
- They must choose the equipment carefully (see selection details below) and ensure workers are trained to use it properly and know how to detect and report any faults.
- like safety harnesses or life jack

Emotion Detection Using Facial recognition

Almost every second of every day, the amount of data on the web grows exponentially. Most of these text, audio, and video files come from web users, who share more and more information through social media, blogs, and web forums. Information is being shared about various subjects, such as health, business, education, travel, and tourism. Organizations can improve their interactions with customers in ways that matter to them if they understand how customers feel about customer service. Sentiment analysis removes the guesswork from such a process by clarifying whether your team's service decision is good or bad. Emotion and sentiment analysis can help marketers connect with potential customers better, no matter where they are.

Emotion Detection using Facial Recognition is an essential area of research in artificial intelligence and computer vision sector, focusing on understanding human emotions and opinions expressed through various mediums, such as text, speech, and images. anges of head position are the main factors that affect the quality of emotion recognition systems using cameras [5]. Especially sensitive for these factors are methods based on 2D image analysis. Methods in which 3D face models are implemented are far more promising. In our experiments we used Microsoft Kinect for 3D face modeling mainly because of its low price and simplicity of operation. Kinect has small scanning resolution .In this project, we focus on facial expression recognition as a modality to analyze emotions. The use of deep learning techniques, specifically Convolutional Neural Networks, has shown significant improvements in image-based classification tasks, making it an ideal choice for this project. Analyzing an image's emotional content is a difficult job in artificial intelligence, particularly in the machine learning subfield of that field. Face detection is the first step of locating or detecting face(s) in a video or single image in the FER process. The images do not consist of faces only, but instead present with complex backgrounds. Indeed, human beings can easily predict facial emotions and other facial features of an image, but these are difficult tasks for machines without excellent training. The primary purpose of face detection is to separate face images from the background (non-faces). Some face detection domains are gesture recognition.

Literature Survey

3.1 Review of Existing Methods

Object Detection Algorithms: An Overview

Object detection is a critical task in computer vision that focuses on identifying and locating objects within images or video frames. Historically, this task was approached using traditional methods such as the Viola-Jones detector and Histogram of Oriented Gradients (HOG) combined with Support Vector Machines (SVM). While these methods provided a foundation, they often relied on manually designed features, which limited their performance in complex and dynamic scenes.

The advent of deep learning has revolutionized object detection. Convolutional Neural Networks (CNNs) emerged as a powerful tool, leading to significant improvements in both accuracy and speed. The introduction of Region-based CNN (R-CNN) marked a major advancement by utilizing CNNs to extract features from regions of interest, which was further enhanced by Fast R-CNN and Faster R-CNN with the incorporation of Region Proposal Networks (RPNs) for faster detection.

YOLO (You Only Look Once)

YOLO, developed by Redmon et al. in 2016, introduced a novel approach to object detection by treating it as a single regression problem. Unlike R-CNN-based methods that generate region proposals before classification, YOLO performs bounding box and class prediction in one step directly from the full image. This end-to-end approach enables much faster detection, making YOLO ideal for real-time applications.

The YOLO algorithm has seen multiple iterations:

- **YOLOv2 and YOLOv3**: These versions introduced enhancements such as anchor boxes, multi-scale predictions, and the Darknet-53 backbone, which improved both speed and accuracy.
- **YOLOv4**: Further optimized the architecture with Cross-Stage Partial connections (CSP), the Mish activation function, and advanced feature pyramid networks.

Personal protective Equipment

Traditional methods for PPE detection primarily relied on manual inspections and rule-based systems. These methods typically used basic image processing techniques such as color segmentation, edge detection, and template matching. For example, color-based segmentation was often employed to detect PPE items like helmets or vests, which are usually of bright and distinct colors (e.g., yellow, orange). While these methods are relatively straightforward to implement, they are highly sensitive to variations in lighting conditions, background noise, and occlusions. Moreover, they often fail to generalize across different environments, leading to inaccurate or incomplete detections

Early Machine Learning Approaches

With the introduction of machine learning, more advanced PPE detection techniques were developed. Early approaches focused on feature extraction using algorithms like Histogram of Oriented Gradients (HOG), Local Binary Patterns (LBP), or Scale-Invariant Feature Transform (SIFT), combined with classifiers such as Support Vector Machines (SVM) or Random Forests. These methods showed improved accuracy compared to traditional rule-based techniques, as they could learn and recognize more complex patterns in images. However, these approaches still required significant manual effort in feature engineering and were often limited by their dependence on handcrafted features.

YOLO (You Only Look Once) Framework

The YOLO (You Only Look Once) framework stands out as one of the most impactful advancements in deep learning for object detection. YOLO models are designed to predict bounding boxes and class probabilities for objects in an image simultaneously, making them extremely fast and accurate. YOLO's ability to perform real-time object detection has made it particularly suitable for applications in dynamic environments, such as construction sites, where PPE compliance must be continuously monitored.

YOLOv8 for PPE Detection

In your project, YOLOv8 was employed to enhance the detection accuracy and speed of PPE items. YOLOv8 is the latest version in the YOLO series and represents significant advancements over its predecessors, including YOLOv5. YOLOv8 introduces several innovations, such as:

- **Anchor-Free Detection**: Unlike earlier versions that relied on predefined anchor boxes, YOLOv8 adopts an anchor-free approach, which simplifies the model and reduces computational overhead. This change has been particularly beneficial in detecting smaller or irregularly shaped PPE items, which are common challenges in real-world environments.
- **Dynamic Head Architecture**: YOLOv8 features a more flexible head architecture that adapts to different detection tasks, allowing for more accurate localization and classification of PPE items. This flexibility ensures that the model can maintain high performance across various conditions, such as different lighting or cluttered backgrounds.
- **Improved Training Strategies**: YOLOv8 incorporates advanced training techniques, including adaptive learning rates and enhanced data augmentation methods. These strategies contribute to better generalization and robustness, enabling the model to perform well even on unseen data or in challenging scenarios.

Comparative Studies and Benchmarks

Comparative studies have consistently shown that YOLOv8 outperforms other deep learning models such as Faster R-CNN, SSD (Single Shot Multibox Detector), and Retina Net in PPE detection. For instance, a study comparing these models on a dataset of construction site images found that YOLOv8 achieved higher accuracy and faster processing times than its counterparts. These studies highlight the advantages of YOLOv8 in real-time applications, where speed and accuracy are critical for effective PPE monitoring.

EXISTING SYSTEM

As said earlier there are a few existing systems that are being used today but none of them are prominent and can give an efficient solution. For example, the system with manual inspection requires personnel to invigilate the entire industry for any discrepancies and then make their move to solve that problem. This is not efficient in the least. Other systems were also implemented like CCTV surveillance which was a step up but that also required manual inspection. There are certain solutions that are automated, but companies are very wary of these solutions for its cost and knowledge transfer problems.

PROBLEMS

- False Positives and Negatives
- Model Biases
- Misinterpretation of Satire
- Wrong Detection
- Dynamic Environments

- Privacy Concerns
- Scalability

Challenges: Developing scalable solutions that can efficiently monitor PPE compliance in large and diverse environments is an ongoing engineering challenge. It encompasses considerations of cost, infrastructure, and centralized management. In conclusion, these open problems in existing PPE detection systems underscore the complex and evolving nature of this field. Addressing these issues necessitates a multidisciplinary approach, combining computer vision, machine learning, ethics, and engineering. Finding innovative solutions to these challenges is vital for improving workplace safety and ensuring the responsible use of PPE detection technology.

PROPOSED SYSTEM

A system using multi-class object detection to ensure that the safety measures and standards are being worn by the employees at the entry of the workplace. This will be a desktop application maintaining simplicity in all aspects making Suring to have compatibility with existing systems if any.

Emotion Detection Using Facial recognition

Emotion Detection: The Need and Challenges

Emotion detection from visual data is a growing field with applications in security, marketing, and human-computer interaction. The goal is to classify human emotions based on visual cues such as facial expressions and body language.

Traditional Methods

Earlier methods for emotion detection involved extracting features using techniques like the Facial Action Coding System (FACS) and Gabor filters. These were often combined with SVMs or other classifiers. However, these approaches were limited by their sensitivity to variations in lighting, pose, and occlusions, leading to inconsistent results.

Deep Learning in Emotion Detection

Deep learning has brought significant improvements to emotion detection. CNNs are used to learn complex facial representations, leading to more accurate emotion classification. Pre-trained models such as VGGFace and ResNet have been adapted for emotion detection, benefiting from extensive facial datasets.

Challenges in Emotion Detection

Despite these advancements, emotion detection remains challenging due to the subtlety and variability of emotions. Contextual factors and the limitations of visual data alone can affect accuracy. Real-time emotion detection also requires low-latency processing and must handle diverse real-world conditions.

Recent Advances

Recent research has explored combining visual data with other modalities like audio and text to enhance emotion detection accuracy. Developing models that capture compound emotions or temporal dynamics using RNNs or LSTM networks has also been a focus. Future research will likely aim to improve generalization across different populations and environments and integrate emotion detection into broader AI systems.

Methodology

1. Overview of Software Tools

The project leverages a diverse tech stack, integrating various technologies to deliver a desktop application with a user-friendly graphical user interface (GUI). The development and deployment processes involve multiple environments and tools, each contributing uniquely to the project's success.

2. Programming Language: Python

- **Description**: Python is an interpreted, high-level, general-purpose programming language known for its emphasis on code readability through significant indentation. It supports multiple programming paradigms, including procedural, object-oriented, and functional programming, and is dynamically typed and garbage-collected.
- **Role in the Project**: Python serves as the primary programming language due to its simplicity and extensive libraries that cater to scientific computing, machine learning, and data processing needs.

3. Deep Learning Framework: TensorFlow

- **Description**: TensorFlow is a free, open-source software library developed by the Google Brain team for machine learning and deep neural network training and inference. It is a symbolic math library based on dataflow and differentiable programming.
- **Role in the Project**: TensorFlow is utilized for developing, training, and fine-tuning the deep learning models, specifically YOLOv8, to perform PPE detection tasks efficiently and accurately.

4. Computer Vision Library: OpenCV

- **Description**: OpenCV is a cross-platform library for real-time computer vision, initially developed by Intel. It offers GPU acceleration and is free to use under the Apache 2 License.
- **Role in the Project**: OpenCV is utilized for image processing tasks, such as resizing and pre-processing images, to optimize them for model inference.

Other Libraries:

- **os**: Used for interacting with the operating system to manage file paths and execute system-level commands.
- **numpy**: A fundamental package for scientific computing with Python, used for handling large, multi-dimensional arrays and matrices, along with a collection of mathematical functions to operate on these arrays.
- **PIL (Python Imaging Library)**: Utilized for image manipulation and processing, such as loading, modifying, and saving images.

- **sys**: Provides access to some variables and functions that interact with the Python runtime environment.

5. Development and Deployment Environments

- **Google Colab**: Google Colab is used for training the YOLOv8 model, taking advantage of its free GPU support and integration with Google Drive, which facilitates model training and storage of large datasets.
- **VS Code (Visual Studio Code)**: The trained model is imported into VS Code, a versatile IDE that supports Python and provides features such as debugging, syntax highlighting, and version control integration.

This software description highlights the comprehensive set of tools and frameworks employed in the project, providing a clear picture of the development and deployment pipeline for the PPE detection system. It ensures the integration of Python, deep learning models, and GUI development, leading to a robust and user-friendly desktop application.

Code for Personal Protective Equipment

To train model in Google Colab

```
!nvidia-smi
!pip install ultralytics
from ultralytics import YOLO
!yolo task=detect mode=predict data=yolov8.pt conf=0.25 source='https://ultralytics.com/images/bus.jpg'
from google.colab import drive
drive.mount('/content/drive')
import os
paths = [
   '../content/drive/MyDrive/DataSet/ppe/train/images',
   '../content/drive/MyDrive/DataSet/ppe/valid/images',
   '../content/drive/MyDrive/DataSet/ppe/test/images'
]
for path in paths:
   if os.path.exists(path):
       print(f"Path exists: {path}")
   else:
       print(f"Path does NOT exist: {path}")
!yolo task=detect mode=train data=yolov81.pt
data=../content/drive/MyDrive/DataSet/ppe/data.yaml epochs=25 imgsz=640
```

After the train model that is best.pt further testing is done in vs code.

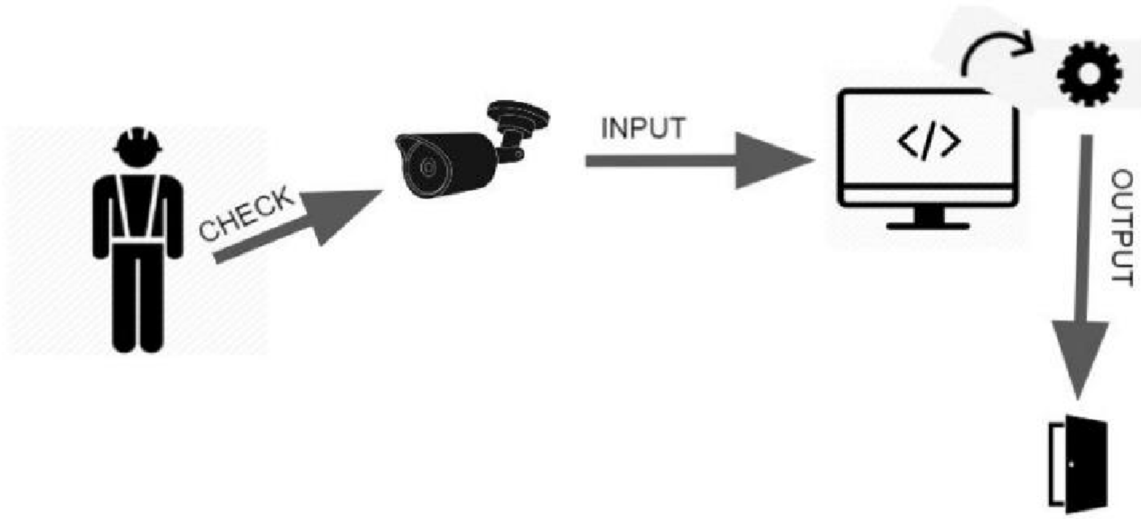
```
from ultralytics import YOLO
import cv2
import cvzone
import math

# cap = cv2.VideoCapture(1)  # For Webcam
# cap.set(3, 1280)
# cap.set(4, 720)
cap = cv2.VideoCapture("../Videos/ppe-1.mp4")  # For Video

model = YOLO("ppe.pt")# saves model best.pt
classNames = ['Hardhat', 'Mask', 'NO-Hardhat', 'NO-Mask', 'NO-Safety Vest', 'Person', 'Safety
Cone',
              'Safety Vest', 'machinery', 'vehicle']
myColor = (0, 0, 255)
while True:
    success, img = cap.read()
    results = model(img, stream=True)
    for r in results:
        boxes = r.boxes
        for box in boxes:
            # Bounding Box
            x1, y1, x2, y2 = box.xyxy[0]
            x1, y1, x2, y2 = int(x1), int(y1), int(x2), int(y2)
            # cv2.rectangle(img,(x1,y1),(x2,y2),(255,0,255),3)
            w, h = x2 - x1, y2 - y1
            # cvzone.cornerRect(img, (x1, y1, w, h))

            # Confidence
            conf = math.ceil((box.conf[0] * 100)) / 100
            # Class Name
            cls = int(box.cls[0])
            currentClass = classNames[cls]
            print(currentClass)
            if conf>0.5:
                if currentClass =='NO-Hardhat' or currentClass =='NO-Safety Vest' or currentClass ==
"NO-Mask":
                    myColor = (0, 0,255)
                elif currentClass =='Hardhat' or currentClass =='Safety Vest' or currentClass ==
"Mask":
```

```
            myColor =(0,255,0)
        else:
            myColor = (255, 0, 0)

        cvzone.putTextRect(img, f'{classNames[cls]} {conf}',
                           (max(0, x1), max(35, y1)), scale=1, thickness=1,colorB=myColor,
                           colorT=(255,255,255),colorR=myColor, offset=5)
        cv2.rectangle(img, (x1, y1), (x2, y2), myColor, 3)

cv2.imshow("Image", img)
cv2.waitKey(1)
```

```
├──.ipynb_checkpoints
├──assets
├──data
├──├─data.yaml
├──├─ppe_data.yaml
│   ├──test
│   │   ├──images
│   │   └──labels
│   ├──train
│   │   ├──images
│   │   └──labels
│   └──valid
│       ├──images
│       └──labels
├──models
├──output
│   └──output_yolov8n_100e
├──results
└──source_files
```

WORKFLOW DETAILS

The below shown workflow diagram will illustrate how our system will work to get a better understanding.



In the above diagram we can see that when a worker enters the industry he is monitored by the camera, and this is taken as the input for the system to make a detection on. Once the worker is detected and seen that he is wearing the required PPE he will be allowed inside if not then he will not be permitted to enter. This will eradicate the need to check manually and make sure that everybody is safe. This can be followed inside of the industry when people are working to make sure that they are adhering to the safety regulations whilst inside the factory/industry. This can be also followed with restricting entry into violated areas, cell phone violation and furthermore.

Code for Emotion Detection using Facial recognition

To train model in Google Colab

```
!nvidia-smi
!pip install ultralytics
from ultralytics import YOLO
!yolo task=detect mode=predict data=yolov8.pt conf=0.25 source='https://ultralytics.com/images/bus.jpg'
from google.colab import drive
drive.mount('/content/drive')
import os
paths = [
```

```
    '../content/drive/MyDrive/DataSet/ppe/train/images',
    '../content/drive/MyDrive/DataSet/ppe/valid/images',
    '../content/drive/MyDrive/DataSet/ppe/test/images'
]
for path in paths:
    if os.path.exists(path):
        print(f"Path exists: {path}")
    else:
        print(f"Path does NOT exist: {path}")
!yolo task=detect mode=train data=yolov81.pt
data=../content/drive/MyDrive/DataSet/ppe/data.yaml epochs=25 imgsz=640
```

After the train model that is best.pt further testing is done in vs code.

```
import cv2
from ultralytics import YOLO

# Load your trained YOLO model for emotion detection
model = YOLO("emotion_detection_model.pt")  # Replace with your model path

def test_emotion_on_video(video_path):
    # Open the video file
    cap = cv2.VideoCapture(video_path)
    if not cap.isOpened():
        print(f"Error: Unable to open video {video_path}")
        return

    # Loop through each frame of the video
    while cap.isOpened():
        ret, frame = cap.read()
        if not ret:
            break

        # Perform emotion detection
        results = model(frame)

        # Annotate the frame with the results
        for result in results:
            for box in result.boxes:
                x1, y1, x2, y2 = map(int, box.xyxy[0])
                confidence = float(box.conf[0])
```

```
            emotion_label = box.cls[0]  # Assuming the label is in cls attribute

            # Draw bounding box and label
            cv2.rectangle(frame, (x1, y1), (x2, y2), (0, 255, 0), 2)
            cv2.putText(frame, f'{emotion_label} ({confidence:.2f})', (x1, y1 - 10),
                        cv2.FONT_HERSHEY_SIMPLEX, 0.9, (0, 255, 0), 2)

        # Display the annotated frame
        cv2.imshow("Emotion Detection", frame)

        # Exit if 'q' is pressed
        if cv2.waitKey(1) & 0xFF == ord('q'):
            break

    # Release the video capture and close windows
    cap.release()
    cv2.destroyAllWindows()
```

YOLO Architecture: An Overview

The YOLO (You Only Look Once) algorithm is a cutting-edge object detection system recognized for its real-time processing capabilities and high accuracy. Unlike conventional object detection methods that involve multiple stages, including region proposals, feature extraction, and classification, YOLO simplifies detection into a single regression problem. This means YOLO can predict bounding boxes and class probabilities directly from the entire image in one go.

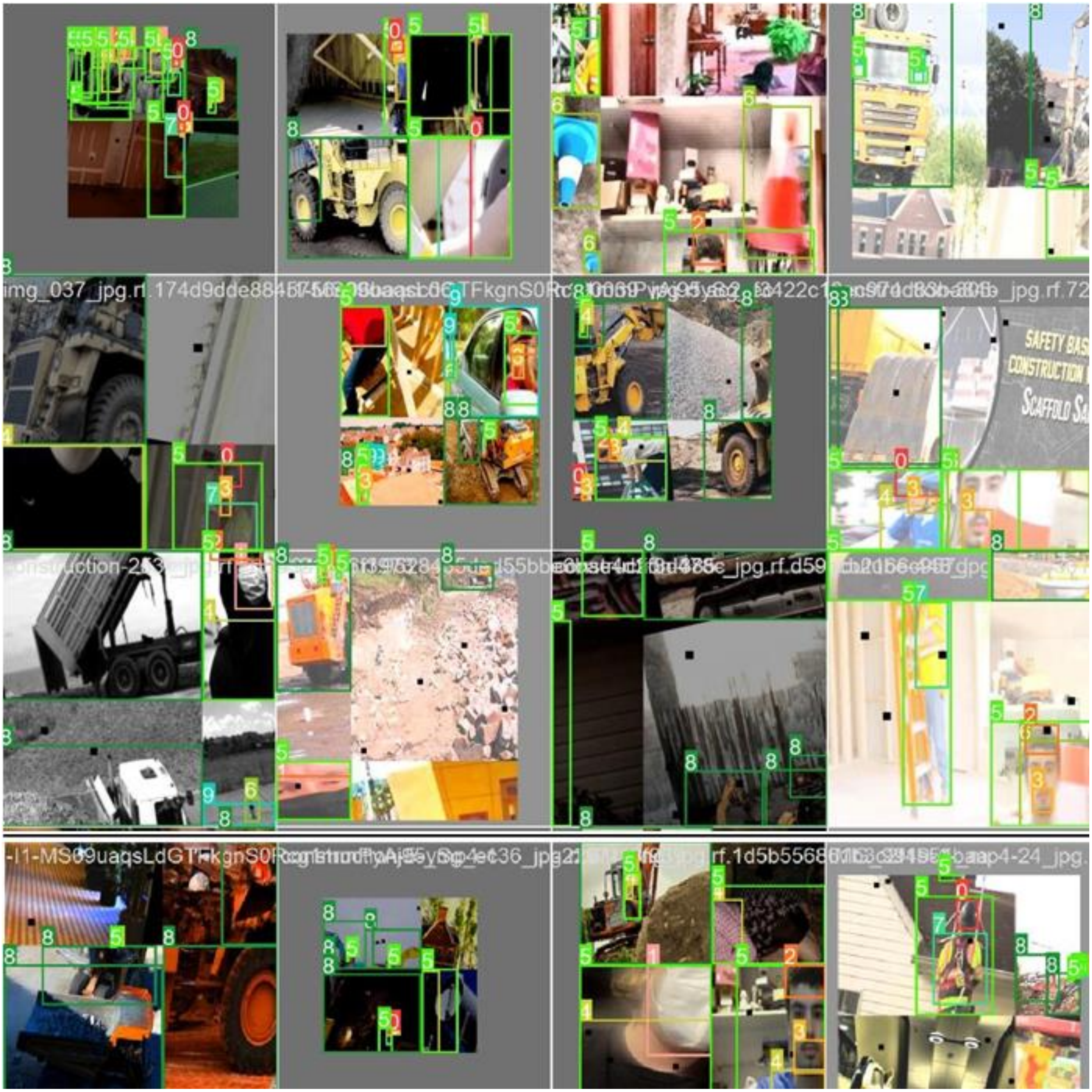
Advantages of YOLO

YOLO's primary strength is its speed, making it well-suited for real-time applications. Its single-stage design allows for rapid processing of images or video frames, leading to high frame rates. Additionally, YOLO's holistic approach, which considers the entire image, helps reduce false positives, particularly for larger objects.

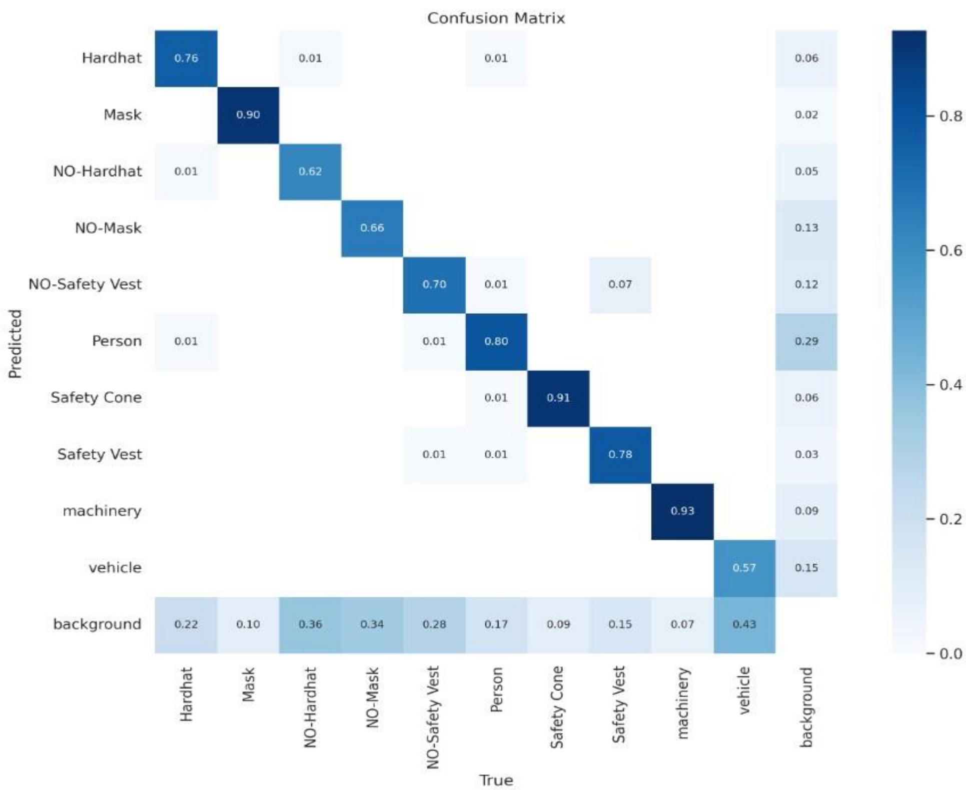
However, YOLO does have some limitations. Its coarse grid structure can make it challenging to detect small objects or objects that are closely packed together. While YOLO is optimized for speed, achieving top-level accuracy may require careful tuning and may involve some trade-offs with model complexity.

Result and Discussion

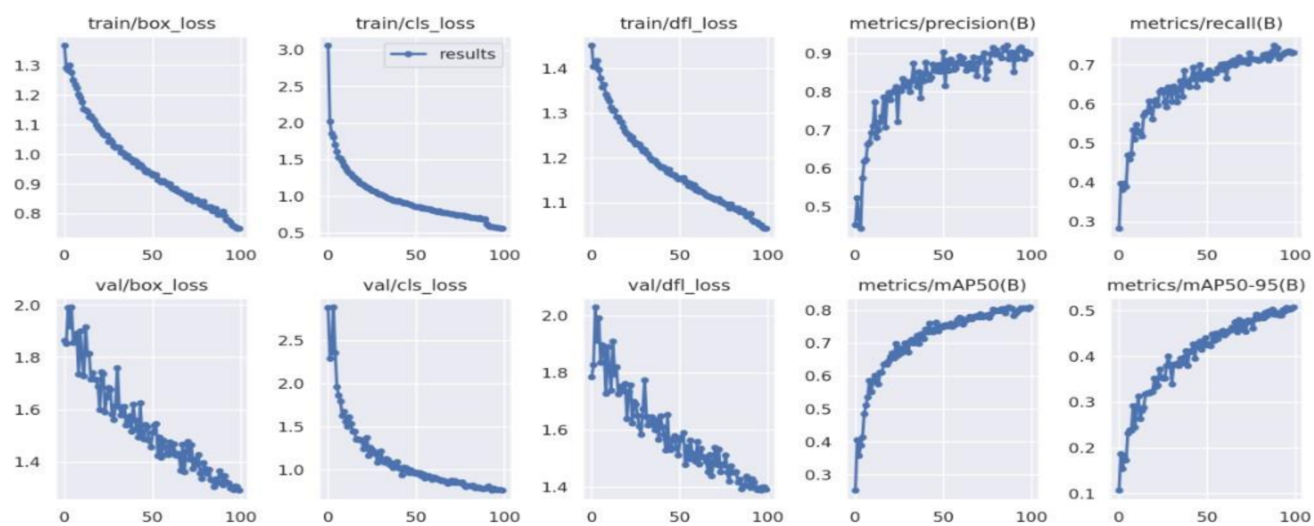
Personal Protective Equipment :training model



Begin with a summary of the key results of your project. This sets the stage for more detail explanations later in the section.



Present quantitative metrics that measure the performance of your model or system. This could include accuracy, precision, recall, F1-score, mean Average Precision (Map), or other relevant metrics

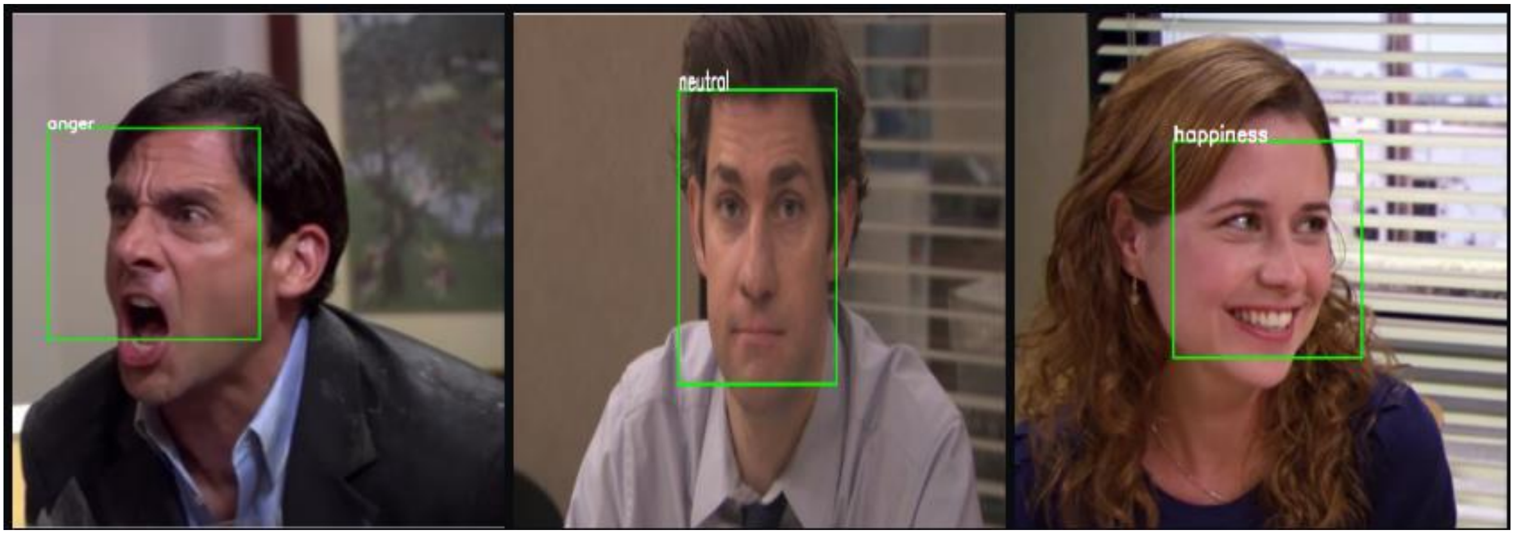


Outputs

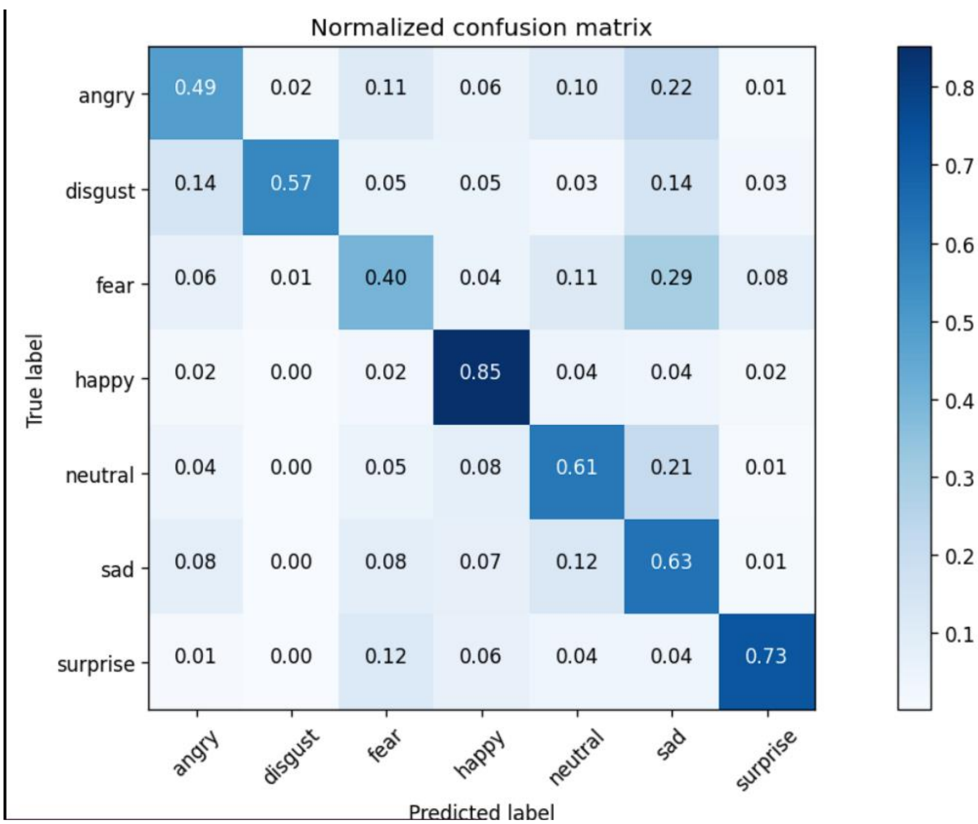

Person 0.87
Hardhat 0.92
Safety Vest 0.86
Safety Vest 0.91


Person 0.89
Hardhat 0.91
Hardhat 0.86
Safety Cone 0.69
NO-Mask 0.94
Safety Vest 0.68
Safety Vest 0.84

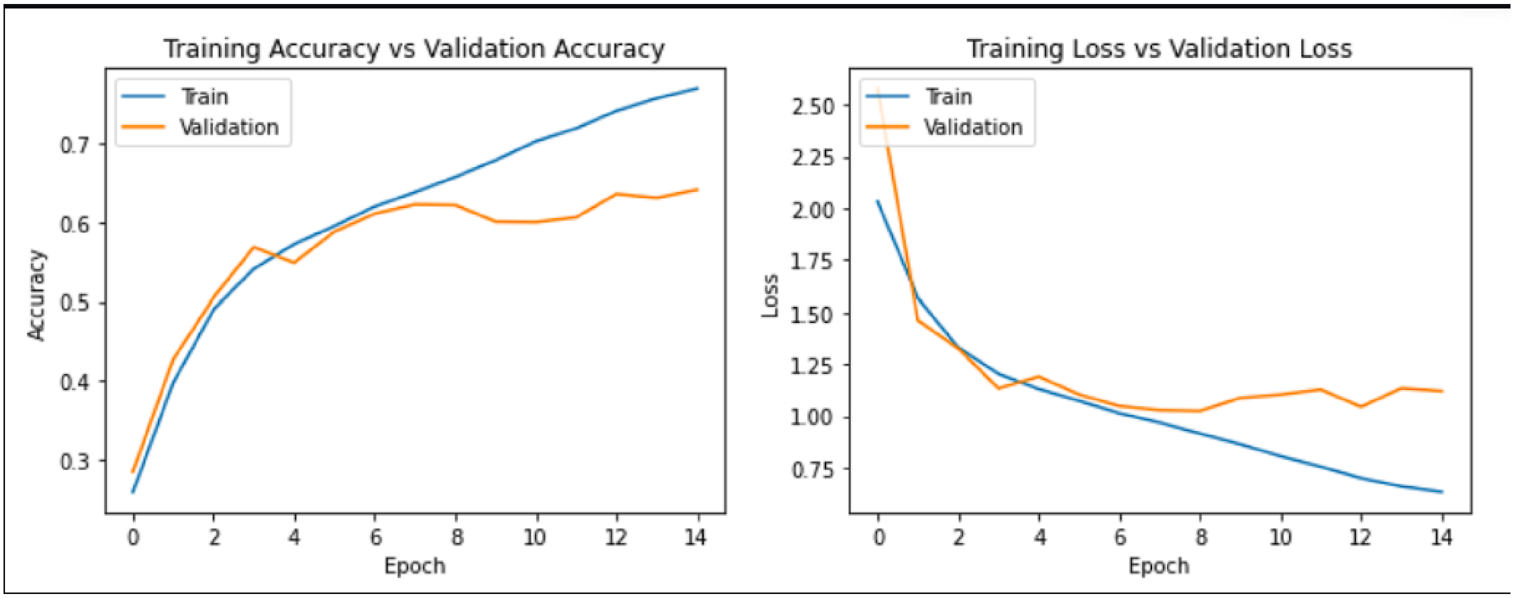
Emotion Detection using Facial recognition



Begin with a summary of the key results of your project. This sets the stage for more detail explanations later in the section



Present quantitative metrics that measure the performance of your model or system. This could include accuracy, precision, recall, F1-score, mean Average Precision (Map), or other relevant metrics



Conclusion

This solution is not the ultimate solution and always has room for improvement. The model that was designed for this purpose meets the conditions and has proved efficient. This can always be grown with the numbers of images being trained. There is always going to be more and more future advancements and we will have to update ourselves and our system to accommodate these changes. But for now, this might be the state of the art as far as Deep Learning is concerned. This project can later be scaled to any environment with the same requirements.
The YOLO model trained for this project demonstrates a significant step forward in achieving the desired results for object detection tasks. While the current solution proves to be efficient and meets the necessary conditions, it is not the ultimate solution and inherently has room for improvement. YOLO, like any other model, benefits greatly from continuous learning and data expansion. As more images are added to the training dataset, the model's accuracy and ability to generalize across different environments are expected to improve. The field of deep learning, particularly in object detection, is rapidly evolving with frequent advancements in algorithms and methodologies. To maintain the model's effectiveness and relevance, it will be crucial to update both our knowledge and the system itself to incorporate these advancements. The scalability of this YOLO-based solution is a key strength, as it can be adapted to various environments with similar requirements.

Future Work

In the future, the following enhancements can beconsidered:

- Train the model for more epochs.
- Compare with 4 other models by YoloV8.
- ML App deployment with alarm triggering

References

[1] Nikolay Filatov et al., “Development of Hard Hat Wearing Monitoring System Using Deep Neural Networks with High Inference Speed” Published in: 2020 International Russian Automation Conference (Ducatoon)., no. 29.Sep. [2] Öner Hatipoğlu et al., “Detection of personal protective equipment” Published in: 2018 26th Signal Processing and Communications Applications Conference (SIU)., no. 2.May. [3] Caio Souto Maior et al., “Personal protective equipment detection in industrial facilities using camera video streaming” Published in: 2018 Book :Safety and Reliability – Safe Societies in a Changing World., no.2 June https://github.com/CiscoDevNet/ppe-detection https://github.com/ciber-lab/pictor-ppe#pre-trained-models https://cv-tricks.com/object-detection/faster-r-cnn-yolo-ssd/