Workplace Safety using AI and ML

Amaan D. Attar, A. Rahim A. R. Gharade, Israr A. Khan, Owais A. Shekasan
Department of Mechanical Engineering
A.I. Kalsekar Technical Campus
Mumbai, India

***Abstract*—Artificial intelligence and machine learning models are mathematical algorithms trained using specific data and human expert input to replicate a decision, which would be made by a human when provided with that same information. Furthermore, computer vision will be the predominant part of our project. Computer vision is a field of artificial intelligence (A.I.) that allows computers and systems to obtain meaningful information from images, videos, and other visual inputs and make changes or provide suggestions based on that information. If artificial intelligence allows computers to think, then computer vision allows them to see, observe and comprehend. It works similarly to human vision. The system is designed and developed to reduce accident rates in the workplace. This technique will increase system efficiency in reducing accident rates, increasing the safety of operators and machines. This proposed system uses OpenCV libraries, deep-learning algorithms, and python programming language. It has various detection modes like apron detection, which ensures operators dressed code, machine components detection that ensures operation safety, and operator safety. Whenever an unsafe action occurs, or the operator does not follow safety rules, it is detected by the algorithm, sensed by the camera, which will trigger an alarm, ultimately lowering the risk of accidents.**

Keywords*— *Artificial Intelligence; Machine Learning; Workplace Safety; YoloV5; Custom Dataset

I. INTRODUCTION

The interaction between humans and machines has been there for many decades, which started with the wheels and contemporarily with lathes, CNCs, automobiles, and much more. The interaction sometimes leads to unavoidable accidents due to unstable machinery conditions, poor ergonomics, un-intentional operator behavior, and poor design. Most of the design, strength, and ergonomics problems have been overcome due to rapid advancement in technology. However, accidents occurring due to the own will of the operator, which could be intentional or unintentional, persist. To avoid such mishaps, constant supervision by a professional person is necessary. However, practically it is not easy to supervise for long hours and multiple operators. Computer vision has a vast application base and solution to contemporary problems. With the help of artificial intelligence and machine learning, we can get constant supervision for infinite time and operators in a single processing unit, which costs negligible. This technology can be used in workshops and school laboratories, machine shops, construction sites, commercial buildings, and much more. Numerous steps and iterations are involved in making a highly efficient model with high accuracy. It begins with creating a database that contains the images and supporting files. After that, the photos are trained with the help of YOLO V5, and it gives the best PT file, which is used with the program to initiate the object detection of the area of interest. The expected output is then displayed with the help of Arduino.

This paper presents our work in detail, including all the steps. The output of our system is to detect the undesired behavior of the operator, which includes not wearing an apron in the workshop, spillage of oil, and not removing the chuck key from the chuck. The following actions have been the primary reason for accidents. The remainder of the paper is organized as follows. Section 2 proposes the model. Section 3 is the detailed description of our model, which includes the layout, dataset creation and training, object detection algorithm and libraries. Section 4 is about hardware used. Section 5 is the result of our testing, which includes accuracy and total integrations. Section 6 is the conclusion, and section 7 presents the scope for improvement.

II. LITERATURE SURVEY

The Authors in this paper [1] have developed a system to detect human face and eyes with the help of a PIR sensor, which gives feedback with the use of a buzzer. The buzzer is connected with Arduino, and it only sends the alert when the face is detected. The authors believe that their model needs some improvisations to increase detection efficiency. For instance, their model cannot see the human face when facing sideways or looking upward, or wearing spectacles.

In this paper [2], the authors aimed to develop a program that detects different shapes with the help of sensors and sends the feedback to the Arduino, which in return drives the actuator according to the command. The authors believe their system is very efficient and can be employed in the manufacturing industry for sorting. Moreover, they state that they can add multiple objects to detect at once and improve the model's efficiency.

The authors of this paper [3] have provided a detailed review of various deep learning-based models for the tasks of generic object detection, specific object detection, and object tracking, evaluating the detection and tracking both individually and in combination. Moreover, it approximates various detectors and proposes the best ones. In addition to it, they have provided the conventional and contemporary trends in object detection and tracking. The authors believe that future examination could be done on multiple object detection models in real-time by combining single and two-stage detectors.

The objective of the authors of this paper [4] was to monitor workers' activity and detect violations that trigger real-time voice alerts on the shop floor. A camera video feed is streamed to a central server using RTSP. The streams are processed and queued across deep learning models. The output of each model is processed, and relevant alerts are sent to the alerting engine. The result of the approach was successful in that it uses computer vision to monitor workers' activity using CCTV feeds and ensures the safety of the workforce in a manufacturing setup.

The purpose of this paper [5] was to explore computer vision-related technologies for occupational health and safety (OHS). The authors review the practice of computer vision technologies for OHS to understand the status of applying OpenCV to construction health and safety. The research concluded by focusing on object recognition, tracking, and health & safety assessment in the construction industry. Computer vision technology is already widely employed in the healthcare industry. Innovations in computer vision will reveal new occupational health and safety applications, like keeping live track of patients.

III. METHODOLOGY

A. System Layout

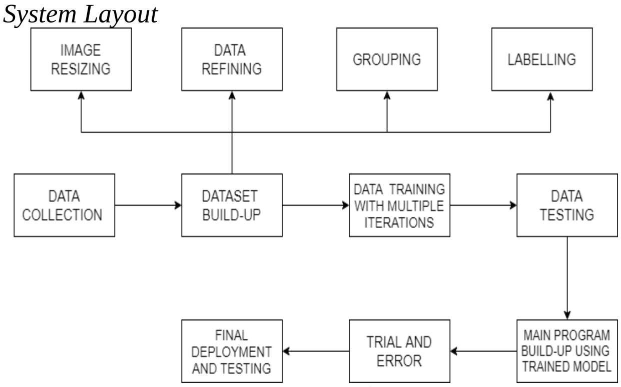


IMAGE RESIZING
DATA REFINING
GROUPING
LABELLING
DATA COLLECTION
DATASET BUILD-UP
DATA TRAINING WITH MULTIPLE ITERATIONS
DATA TESTING
FINAL DEPLOYMENT AND TESTING
TRIAL AND ERROR
MAIN PROGRAM BUILD-UP USING TRAINED MODEL

Figure 1: Flowchart of system

In this paper, our system consists of one camera mounted on the wall to obtain a complete view of the workplace, including the operators and machines. The camera mounted on the wall is connected to the operating system, and the algorithm monitors the live feed to detect undesired actions. If any dangerous activity is detected, the alarm gets triggered by the signal sent to Arduino.

The whole system comprises seven distinctive processes. The first step of the process is data collection. Data collection is a simple step in which digital images are collected of the object, which is to be supervised. Images are gathered in multiple orientations, different environments, and massive quantities. The total number of images reckons in thousands—the greater the number of images, the greater the system's accuracy. The next step is the data set build-up. The collected data is processed for the training phase in data set build-up.

Once the data is built, we proceed to the next step, which is data training. The data created in an earlier phase is then used for training. Created data is trained with the help of YOLOV5. YOLOV5 is an object detection architecture that trains the images by performing hundreds of iterations and providing the “best.pt” file. The total time required for the iterations depends upon the number of photographs and the size of the built data. Once the pt file is generated, it is tested for accuracy and performance evaluation. The next step is to set up the main program. The main program is built-up with the help of the pt file.

After setting up the main program, the next step is trial and error for improvisation. The system is improved by a trial and error approach. The trial and error mechanism helps increase the model's accuracy and efficiency, thereby making the system more compatible in real-time. The last stage is the implementation of the system in the workplace, where it must provide supervision.

B. Dataset Creation and Training

Once the data is collected, the further steps are data set creation and training. There are four discrete steps to create a data set. The images contained in the data collection phase are resized. The dimensions used are 500*500 Pixels, and all the photos have the exact measurements and a total number of pixels that is 250,000. The images are resized with the help of a tool named "Faststone Image Resizer." The tool helps to resize images as per the requirement.

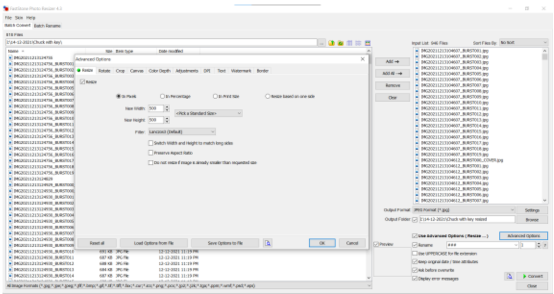


Figure 2: Screenshot of Faststone Image Resizer Software

Further, all the images are resized, renamed, and saved at the specified location. The process of resizing images with the help of the tool saves much time instead of performing the process individually on each picture. The photos are resized because each was taken with a different orientation in the data collection process, leading to various dimensions. Also, the memory of the pictures is in megabytes. Thus it is necessary to resize images.

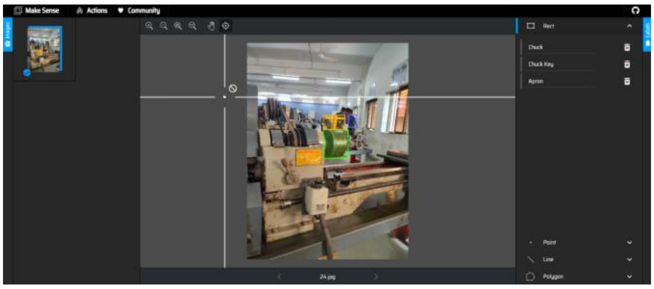
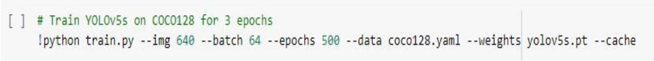


Figure 3: Screenshot of Make sense Webpage

After resizing the images, the next step is to label the pictures. Marking the images means creating the bounding box around the region of interest. The tool named "Make sense" is used to carry out the process. The tool helps to make the bounding boxes around the area of interest. The device can make multiple bounding boxes in a single image. Each label and the corresponding bounding box can be associated with a different color, simplifying the process.

The complete process is done by selecting manually. After that, the tool permits three options to convert the file, which is required for training. The available options are text, CSV, and XML; our requirement is a text file. The text file contains the four corner coordinates of the bounding boxes; each image has one txt file. The data set folder is created, containing two folders named 'train' and 'validation' (val); each includes two subfolders named images and labels. The image folder in the train folder comprises 80% of the processed pictures, and the label folder contains the corresponding text files. The remaining 20% of the processed data is in the subfolder of the validation (val) folder.



```
[ ]  # Train YOLOv5s on COCO128 for 3 epochs
     !python train.py --img 640 --batch 64 --epochs 500 --data coco128.yaml --weights yolov5s.pt --cache
```

Figure 4: Command line for training custom dataset

!python train.py --img 640 --batch 64 --epochs 500 --data coco128.yaml --weights yolov5s.pt --cache

Training of data set begins in command prompt. The above code is used to train the data set. The controlling parameters in the code are epochs, batch, image, and weights. The epoch is the parameter that defines the number of times the learning algorithm will function through the complete training dataset. One epoch means that each piece in the training dataset has had a chance to revamp the standard interior parameters.

The batch size is a parameter that defines the number of samples to work through before revising the internal model parameters. In general, a batch size of 32 is a good starting juncture, and we should also try 64, 128, and 256. Other values (lower or higher) may be exemplary for some data sets, but the given range is generally the best to start experimenting.

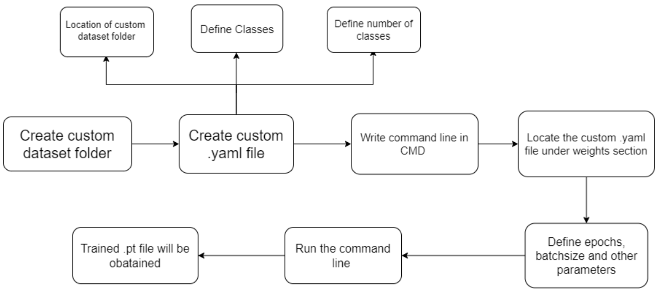
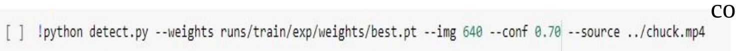

Location of custom dataset folder
Define Classes
Define number of classes
Create custom dataset folder
Create custom .yaml file
Write command line in CMD
Locate the custom .yaml file under weights section
Define epochs, batchsize and other parameters
Run the command line
Trained .pt file will be obatained

Figure 5: Flowchart for training a custom data

C. *Object Detection, Libraries and Algorithm*

The mentioned code is used for object detection; the code is run in command prompt. The controlling parameters of code are weights, confidence, and pt file. The "best.pt" file is generated after the training of the dataset, and it helps the algorithm interpret the live feed or the recorded feed. The code mentions the file's directory on which the detection would be done.



```
[ ]  !python detect.py --weights runs/train/exp/weights/best.pt --img 640 --conf 0.70 --source ../chuck.mp4
```

Figure 6: Command line for detection

!python detect.py –weights best.pt --img 640 --conf 0.70 --source ../chuckmp4

The mentioned code is used for object detection; the code is run in command prompt. The controlling parameters of code are weights, confidence, and pt file. The "best.pt" file is generated after the training of the dataset, and it helps the algorithm interpret the live feed or the recorded feed. The code mentions the file's directory on which the detection would be done. A total of 640 images were used.

The weight files are generally held in the same folder as the topology. A confidence interval, in statistics, refers to the probability that a population parameter falls within a set of values for a certain proportion of times. Confidence intervals measure the degree of indecision or certitude in a sampling method.

YOLOv5 is the modified version of the YOLOv4, released after it. YOLO stands for 'You Only Look Once', the algorithm used for making real-time object detection for the trained dataset with the help of neural networks that work on the Pytorch framework. YOLOv5 was developed by Roboflow and Ultralytic, produced on the GitHub repository. YOLOv5 works on the backbone of Darknet. According to the working of YOLOv5, the collected data (dataset) is loaded in CSPDarknet for extracting data, and then it is loaded in the PANet to combine the features. Finally, it is passed through convolution layers to show the detection results. The detection using YOLOv5 has high accuracy due to its open-source contribution. It is better than its previous versions.

OpenCV is an open-source machine learning library that uses Artificial Intelligence to empower computers and different systems to extract essential data, information, and understanding from the digital videos and images or any visual data for deriving meaningful information and can take the measure on it according to data. OpenCV stands for open-source computer vision. OpenCV is an open-source platform where thousands of comprehensive and state-of-the-art computer vision algorithms can be available. The available algorithms can be used for object detection, recognition, classifying, and image processing. Currently, this open-source library is enormously used in different companies and for research work also by thousands of people. OpenCV supports languages like C++, Python, Java interface and works in Windows, Linux, mac OS, and Android. OpenCV is a modern open-source library focused only on real-time object detection implementation.

IV. HARDWARE USED

In the project's assembly, different hardware were used. The leading hardware is a webcam and the Arduino. Further other accessories are used, like the buzzer, LED bulb, and breadboard, essential for making connections. A webcam is used for video capture so that the algorithm can detect objects. An Arduino is a link between webcam and buzzer to sense output from the webcam for transferring it to the buzzer. A buzzer is used as an alerting device that receives commands from Arduino. Finally, a breadboard is used to make connections of buzzer and LEDs with the Arduino.

A. *Webcam*

The camera that was used is a Logitech C505 HD webcam. It takes snapshots at 1.2 megapixels with a resolution of 720p at 30 Frames per second. This webcam is used because of its ease of use, and it covers a good area of the field. The simple integrated webcam can also be used.

B. *Arduino*

The microcontroller used in this project is Arduino ATmega2560 which has a user-friendly USB programmable microcontroller. It consists of 54 digital I/O pins and 16 analogue I/O pins. The microcontroller has 256 Kbytes of flash memory, 8 Kbytes of static RAM and 4 Kbytes of EEPROM. The Arduino has a clock speed of 16 MHz crystal oscillator. A USB cable powers the Arduino ATmega2560 by an external power supply. The programming is fed by an open-source Arduino IDE, available free of cost. The data from python to Arduino is transferred using the Standard Pyfirmata library.

C. *Buzzer*

The buzzer used is an RS PRO flanged continuous tone buzzer with ABS housing. It has a voltage rating of 12V DC and operates on a 96dB sound level. It has high output levels and operating temperatures of -20°C to +60°C. The maximum supply voltage is 20V DC, and the minimum supply voltage is 1.5V DC. It works in a frequency range of 3.3 kHz to 2300 Hz.

V. RESULTS

According to the algorithm, a dynamic live feed of the workplace is streamed with the help of a camera. It detects chuck, chuck key, and operator’s apron for which that was trained. All the datasets are trained at 500 epochs in a batch of 8 with 100 images—a total of 25 images each of apron, chuck, chuck key, and chuck with key. The results are pretty decent, with an accuracy of 80 to 90 percent. All the parameters are detected in the rectangular box can be seen in the figures. When the chuck key is left on the chuck, the camera makes the detection, and signals are sent to the Arduino, which activates the buzzer for alerting the operator.

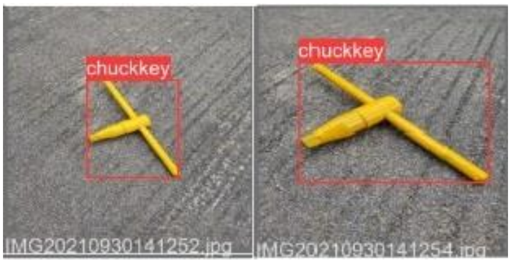


Figure 7: Test result indicates the detection of a chuck key

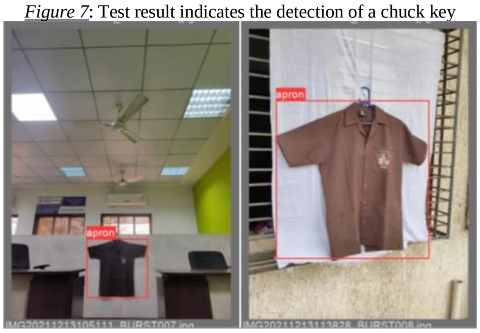


Figure 8: Test result indicates the detection of an apron

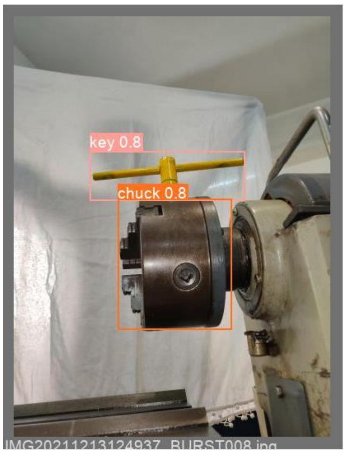


Figure 9: Test result indicates the detection of chuck and chuck key

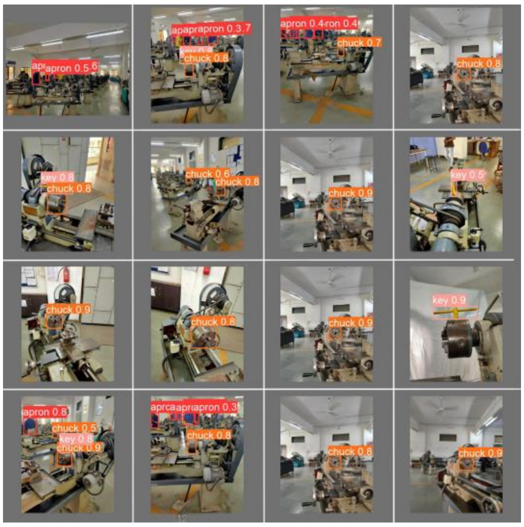


Figure 10: Final results of the detections after combining all datasets

The detection accuracy obtained after the evaluation is about 80 to 90 percent for the 500 epochs in the batch of 8. The camera can continuously detect 25 to 30 frames per second, but sometimes it fluctuates as the frame rate drops.

VI. CONCLUSION

This project aims to improve safety in the workplace as many accidental rates are increasing gradually due to lack of attention towards workplace safety precautions and unsafe actions or irresponsible behavior, and improper machine handling. The focus of work is to overcome the accidents related to the workplace and create a safe working environment for which an Artificial Intelligence algorithm is built that monitors machines and operators continuously by employing a camera feed and alerting system that ensures effective workflow.

As Artificial intelligence develops continuously and the availability of open-source technology, deploying AI algorithms for the workplace's safety is done affordably and effectively. It has been designed to detect machine parts and the operator's apron. The applied algorithm for the detection is working with high accuracy. An Arduino is deployed with this algorithm; whenever the camera detects an unsafe environment, signals are sent to Arduino, which alerts the operator using a buzzer. The experimental results were obtained are good, and this technology can be used in small and large scale production.

VII. SCOPE FOR IMPROVEMENT

Artificial Intelligence is still in the approbatory stage, and nearly in all the significant sectors, it is being used steadily. This project can assist in the growth of Artificial intelligence technology in the small and large scale industry to reduce the accident rate at an affordable cost. Further in the future, this work can be improved by training it for more parts of the machine and more safety accessories related to the operator's safety. In the future, this project can be used for remote monitoring of machines that enables the operator to track various performance parameters and keep a maintenance record of them. Due to the low lighting conditions faced in the workplace, the black-colored shirts are detected as an apron. It could be improved by training the data set with the addition of images taken in low lighting conditions. The project was done by using only one camera that has focus only from one angle.

VIII. REFERENCES

[1] Biometric Identification Using OpenCV Based on Arduino
[2] Shape detection and classification using OpenCV and Arduino Uno
[3] Deep learning in multi object detection and tracking state of the art
[4] Using Computer Vision to enhance Safety of Workforce in Manufacturing in a Post COVID World
[5] A review of the applications of computer vision to construction health and safety
[6] Industry and Object Recognition: Applications, Applied Research and Challenges
[7] Research Paper on Artificial Intelligence
[8] On using AI-based human identification in improving surveillance system efficiency