IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Real Time Safety Equipment Kit Detection System

[1]Dhanshree Zade, [2]Yashkumar Atkare, [3]Hitesh Bhagwat, [4]Chetna Kshirsagar

[1]Dr.Sharda Chabbria, [2]Atul Ghumade, [3]Harish Zambani

[1]Department of Artificial Intelligence,

[1]GH Raisoni Institute of Engineering and Technology, Nagpur , Maharashtra , India

HIXAA Technologies Pvt. Ltd. Nagpur , Maharashtra , India

Abstract: Construction had the highest number of fatal work accidents of any industry, due to the high number of accidents each year. There are many solutions to ensure worker safety and reduce accidents, one of which is to ensure the proper use of appropriate personal protective equipment (PPE kit) as defined in safety regulations. However, monitoring the use of personal protective equipment, which is largely based on manual checks, is time-consuming and inefficient. The Several attempts were made. The resulting recognition accuracy on the 12 main armors is up to 98%, while the accuracy of face detection and recognition is 96%. The obtained results showed the ability to identify recognize faces and all the equipment very accurately and remember them in real time. The study uses convolutional neural network (CNN) models developed by applying transfer learning to the basic version of YOLOv7 and The Faster RCNN deep learning. Considering the presence the model predicts fulfillment of requirements in 12 categories, such as jacket, jacket, mask, no mask, shoes, no shoes, helmet, no helmet, Safety Belt, No Safety Belt, No front Safety Belt, Front Safety Belt.

***Index Terms* -Real Time Object Detection ,YOLOv7, Faster RCNN , Machine Learning , Artificial Intelligence**

I. Introduction

Construction safety is a major concern for industry professionals and researchers worldwide. Even after conducting risk analyzes and implementing appropriate controls ,workers in the construction work environment are still exposed to safety risks . The need for personal protective equipment is important in this context. Automatic and real-time detection of employee non-compliance in the use of personal protective equipment is a major concern. Advances in computer vision and data analysis, especially deep learning algorithms, can solve this construction problem. This study developed a framework for real-time identification of PPE safety requirements for construction workers, designed to be integrated into an organization's safety workflow. The study uses convolutional neural network models developed by applying transfer learning to the basic version of YOLOv7 And The Faster RCNN deep learning.

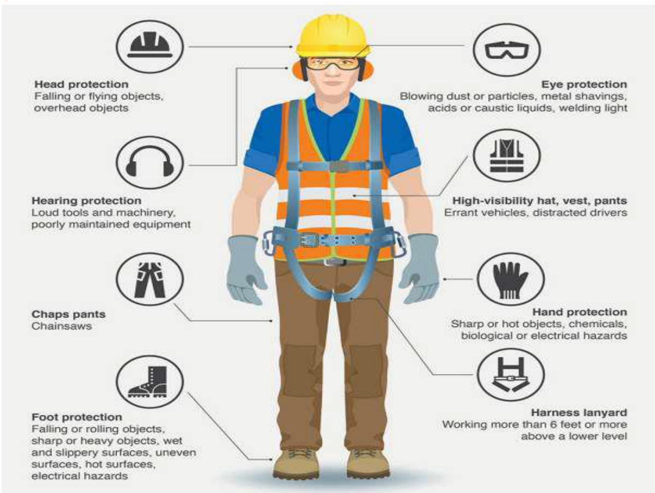


Fig1: Worker with Safety Equipements.

Considering the presence of helmets and safety vests, the model predicts fulfillment of requirements in 12 categories, such as jacket, No jacket, mask, no mask, shoes, no shoes, helmet, no helmet, No safety belt, safety belt, no front safety belt, front safety belt. A dataset of 5000 images was collected from video recordings from several online sources and this web-based collection was used to train the model. The model gives an F1 score of with an average precision of 0.96 and a recall rate of 96% on a test data set of . Overall, the study provides evidence of the feasibility and utility of computer vision-based techniques to automate safety-related compliance processes at , construction sites.

a. Faster RCNN :

Recent advances in object detection have led to the success of region proposal methods and region-based convolutional neural networks (RCNNs). Although region-specific CNNs were computationally expensive when originally developed in , their cost has been greatly reduced due to proposition sharing. The latest incarnation, Rapid R-CNN, achieves near real- time rates by using very deep networks while ignoring the time required for region proposals. Now the proposals are a computational bottleneck in the testing time of state-of-the art detector systems. Area proposal methods are usually based on favorable characteristics and economic reasoning methods. Selective Search, one of the most popular methods, greedily matches super pixels based on designed low-level features. Compared to efficient detection networks, selective search is orders of magnitude slower, at 2 seconds per frame on a CPU implementation Edge Boxes currently offer the best compromise between presentation quality and speed, at 0.2 seconds per frame. However, the regional recommendation phase still consumes the same amount of runtime as the detection network

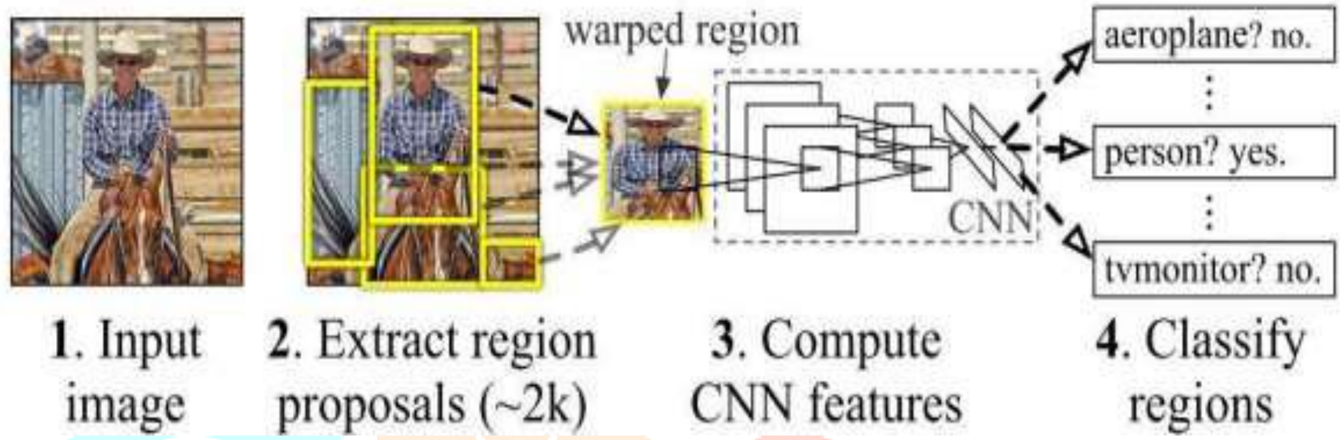


Fig no 2 Faster RCNN

b. YOLOv7 :

YOLOv7 is the fastest and most accurate real-time object recognition model for computer vision tasks. In July 2022, Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao published the official YOLOv7 paper titled "YOLOv7: Trainable Free Bag Sets New State-of-the-art Real-time Object Detectors". The YOLOv7 algorithm is making waves in the computer vision and machine learning communities. It requires many times cheaper hardware than other neural networks and can be trained much faster on small data sets without pre-trained weights. Therefore, YOLOv7 is expected to become the industry standard for object detection in the near future, surpassing the current state of art in real-time applications. Our background: Viso.ai offers the most advanced enterprise computer vision implementation platform, Viso Suite. Leading organizations use a no-code software infrastructure to train YOLOv7 models and build computer vision applications. In this article, we present how YOLOv7 works and what makes it the best object detection algorithm available.

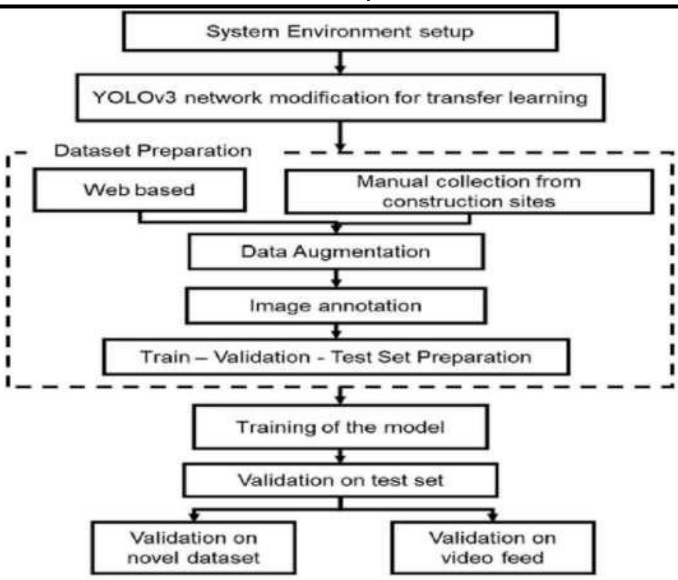


Fig no 3 YOLOv7

c. Jetson Nano Development Kit :

Jetson nano is a small AI computer made for beginners, learners, and developers. If you want to enter into AI world, this board will be the best start for you. Jetson nano is built with 64 quadcore ARM Cortex-A57 CPU. Its running speed is 1.43 GHz and has a maxwell GPU with 128 CUDA cores capable of 472 GFLOPs (FP16). Talking about processing memory, it has 4GB of 64-bit LPDDR4 RAM on-board and 16GB of eMMC storage and runs Linux for Tegra.

Fig no 4. Jetson Nano Development Kit

d. UI (User Interface) :

In this project we also develop & deploy UI. We develop UI with the help of PYQT UIframework software. PYQT is a Python binding of the cross-platform GUI toolkit Qt, implemented as a Python plug-in. PyQt is free software developed by the British firm Riverbank Computing.PyQt5 is cross-platform GUI toolkit, a set of python bindings for Qtv5. One can develop an interactive desktop application with so much ease because of the **tools** and simplicity provided by this library. A GUI application consists of Front-end and Backend. With the help of PYQT is we design our UI of our application and import in .ui format and then convert into .py format with the help of PYQT designer framework. After that we pass parameter to ui via python as function and taking response from it back to backend for processing that input.

Fig no 5. This Ui control panel of this PPE detection tool

II. Methodology

This study implemented the YOLOv7 model with a network architecture using a spherical backend. Keras is an advanced neural web API written in Python that enables seamless implementation of algorithms using the Tensorflow framework, providing an easy way to implement deep learning algorithms and quickly experiment with the results. Expresses the research method of this study, which follows the standard method of deep learning models. The main processes include algorithm definition, dataset processing, training and model validation. The following sections describe these processes in detail. Currently, monitoring the use of personal protective equipment is still done manually. It I costly and inaccurate because a large number of workers need to be checked over a period of time. PPE detection and face detection and recognition. The main aim of PPE detection is to determine the presence of required PPEs while face detection and recognition aim to determine the identity of the workers. Inspired from impressive results of deep convolutional neural network for different computer vision tasks, in this paper, PPEs and face are detected thanks to YOLO (You only look once) network while Face Net is fine-tuned for face recognition.

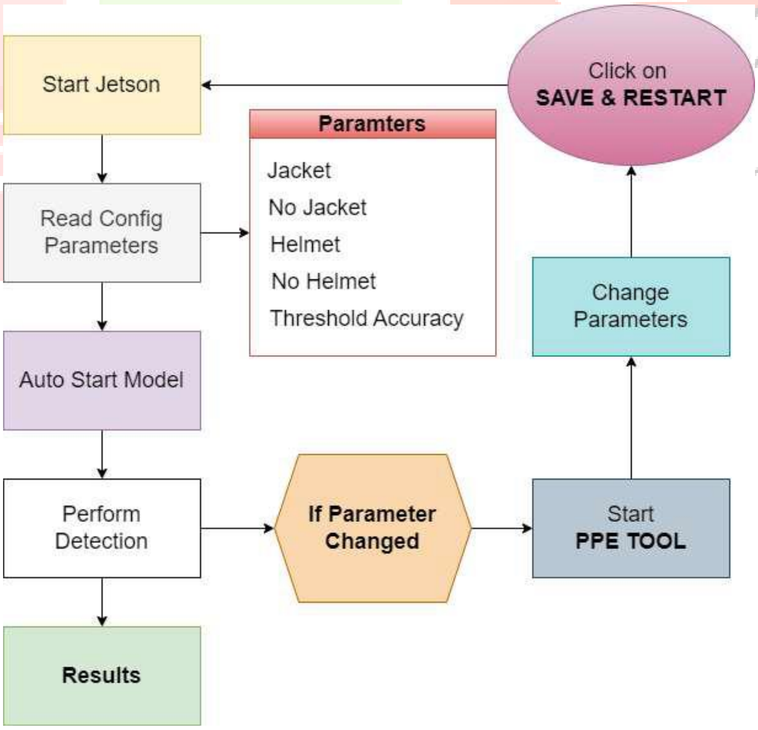


Fig no 6. Project Working Flow

The project is deployed on Yolov7 ML model on Jetson Engine. NVIDIA Jetson is used by professional developers to create breakthrough AI products across all industries, and by students and enthusiasts for hands-on AI learning and making amazing projects. Nvidia Jetson is a series of embedded computing boards from Nvidia. The Jetson TK1, TX1 and TX2 models all carry a Tegra processor (or SoC) from Nvidia that integrates an ARM architecture central processing unit (CPU). Jetson is a low-power system and is designed for accelerating machine learning applications. Nvidia shipped the Nvidia Jetson TK1 development board containing a Tegra K1 SoC in the T124 variant and running Ubuntu Linux. 32 In this project, model develop & deploy in jetson engine & it's running on Ubuntu 18.04 LTS version. As explain about installation of library of python and machine learning in earier chapters. Now are let's discuss about working flow of this project. Firstly the jetson engine is booted with Ubuntu 18.04 LTS version then the program is booted in the OS after that all the configuration data read by the program and pass directly to the Yolov7 AI/ML model which connected to this application. The parameters are Jacket, No Jacket, Helmet & No Helmet after taking this input from application user interface (UI), it directly save that configuration. Save and reboot the device to pass and save the .config file. Auto start model the to start detection the object and perform detection directly get results. If parameter is changed it started program PPE tool then the user can change parameters and then again save and restart to get results onboard.

III. ENVIRONMENT SETUP

System hardware was used to test the algorithm's real-time processing with field data. It consisted of Intel Core i7- 790@3.60GHz 8 CPU and Intel HD Graphics GPU with 8GB 1600MHz RAM. Ubuntu 16.0 base environment consists of python3, pip, OpenCV and Tensorflow. Image annotation is performed on the label using Image-master software, an open source software available on GitHub. The development was done with jupyter notebook and atomic text editor was used. The YOLOv7 model is used to convey learning in the spherical framework. The final output layer is modified to produce 12 classes, namely NO BELTS, SEAT BELT, FIELD, JACKET, NO HELMET, HELMET, NO FRONT BELT, FIRST BELT, MASK, NO MASK, SHOE, NO SHOE - by changing the filter sizes. The YOLOv7 trained weights are used as the initial set of CNN weights, and the convolutional and fully connected layers are opened for training using construction data. This is true for both Yolov7 and Faster RCNN models.

IV. DATA SET PREPARATION

Roboflow is a computer vision platform that enables users to build computer vision models faster and more accurately by providing better data collection, pre-processing and model training techniques. Roboflow allows users to load custom datasets, draw annotations, edit image orientations, resize images, change image contrast, and perform data augmentations. It can also be used to train models. As I mentioned, Roboflow also has a generic conversion tool that allows users to load and convert annotations from one format to another without having to write conversion scripts for custom object detection datasets. First we need to login at https://roboflow.com/. Note that if you select the "Public Workspace" option when you sign up for a free tier account, you can upload image sets of up to 10,000 source images. After creating a project, the next step is to load a dataset containing both images and existing annotation files (can be in JSON, Txt or XML format) or design annotations from scratch. The last step after preparing the dataset is to export the data in ZIP format.

		Train set	Validation set	Test set
Number of images		2217	238	54
Data points per class	NOTSAFE	882	96	19
	NoHardhat	779	72	17
	NoJacket	970	112	23
	SAFE	1040	115	27

Table no 1: Data point distribution among train-validation-test datasets

		Human			
		Notsafe	NoHardhat	NoJacket	Safe
MODEL	NOTSAFE	85	0	1	0
	NoHartdhat	0	75	0	2
	NoJacket	5	0	105	2
	SAFE	0	0	4	96

Table no 2: Confusion matrix for the validation set.

	TP	TN	FP	FN	Precision	Recall	F1 score
NOTSAFE	85	284	1	5	0.99	0.94	0.97
NoHardhat	75	298	2	0	0.97	1.00	0.99
NoJacket	105	258	7	5	0.94	0.95	0.95
SAFE	96	271	4	4	0.96	0.96	0.96
Average					0.96	0.96	0.96

Table no 3 : Classification report for the validation set.

		Human			
		Notsafe	NoHardhat	NoJacket	Safe
MODEL	NOTSAFE	19	0	1	0
	NoHartdhat	0	17	0	0
	NoJacket	0	0	21	1
	SAFE	0	0	1	26

Table no 4 : Confusion matrix for the test set.

	TP	TN	FP	FN	Precision	Recall	F1 score
NOTSAFE	19	66	1	0	0.95	1.00	0.97
NoHardhat	17	69	0	0	1.00	1.00	1.00
NoJacket	21	62	1	2	0.95	0.91	0.93
SAFE	26	58	1	1	0.96	0.96	0.96
Average					0.97	0.97	0.97

Table no 5: Classification report for the test dataset.

V. TRAINING

The training was done in three stages. The first stage has an epoch of 500 before and after unfreezing all layers. Later ones had 50 epochs each before and after unfreezing all layers. The final model from previous training was used as the initial model for subsequent training. The first model for training was made by combining the cfg file and weights file of yolov3. An h5 model file was generated, which had 252 layers. The model layers include 23 adding layers, 72 batch normalization layers, 75 2D convolutional layers, 72 leaky relu layers, five 2D zero padding layers, two each concatenate and up sampling layers, and a YOLO loss layer. The data after all epochs before unfreezing all layers and three epochs after unfreezing all layers in training was stored as a new model. Tensor board call backs were added to the code for visualization of training loss. For better training, reducing the learning rate of a change in validation loss was not less than 0.1, considering the last three epochs were incorporated in the code. To avoid overfitting, early stopping of a change in validation loss is nil for the last ten epochs used. During training, only the last ten layers were trainable for the first part, and in the second part, all layers were trainable. The batch size for the first stage of training is 8 and 4 for the first and second parts. The later stages had batch sizes of 4 and 2 for the first and second parts, respectively. Adam optimizer, with an initial learning rate of 0.001, was used for training. As early stoppage was used, with the first training stopped after 509 epochs, the second one after 92 epochs, and the last one after 70 epochs. The training progress of a network was monitored using any of the two parameters, namely loss, and accuracy. In this model, a new loss function named YOLO loss was used to monitor the training. After the first stage the loss was 21.88, after the second it came to 12.87 and the final loss is 12.06. The loss versus the number of epochs curve is as shown in below Figure.

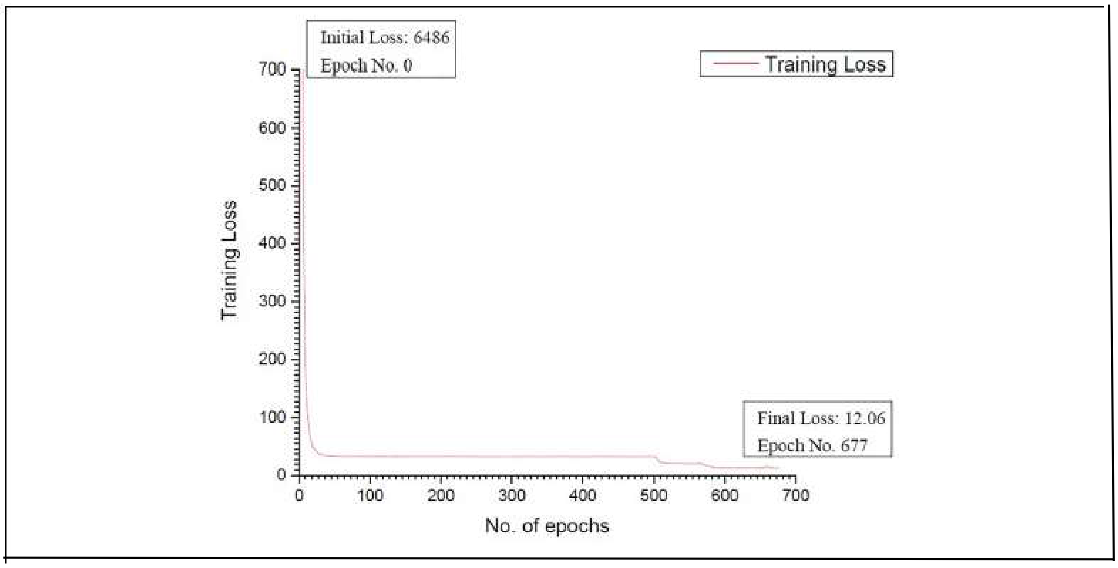


Fig no 7 Training loss vs Epoch number

VI. Result & Discussion

The final loss of the network was 12.06 after three training sessions with steps. A confusion matrix was created to validate and test the dataset, and model accuracy was calculated as the number of true predictions out of the total number of predictions. In addition, a completely new dataset with contexts and backgrounds was created using videos of people wearing PPE. The trained model was then tested on both image and video files to evaluate the performance of algorithms.

Sr .no	Models	Accuracy	FPS
1	yolo v7	98%	7 FPS
2	yolo v7 tiny	70%	25 FPS
3	Faster-RCNN	96%	-

1. Performance of Video file

To test the reliability of the trained model, the model was used to predict a video file created from CCTV footage in mp4 format. Processing occurred at 2 frames per second. This reduced speed is due to the limited computing resources of the system. The predicted model recognized people in the frames. Additionally, the algorithm can classify a person as dangerous if the personal protective equipment they are wearing changes, such as removing a helmet or jacket. Thus, a dangerous person was reclassified as SAFE if a person meeting the IKV standard was identified.

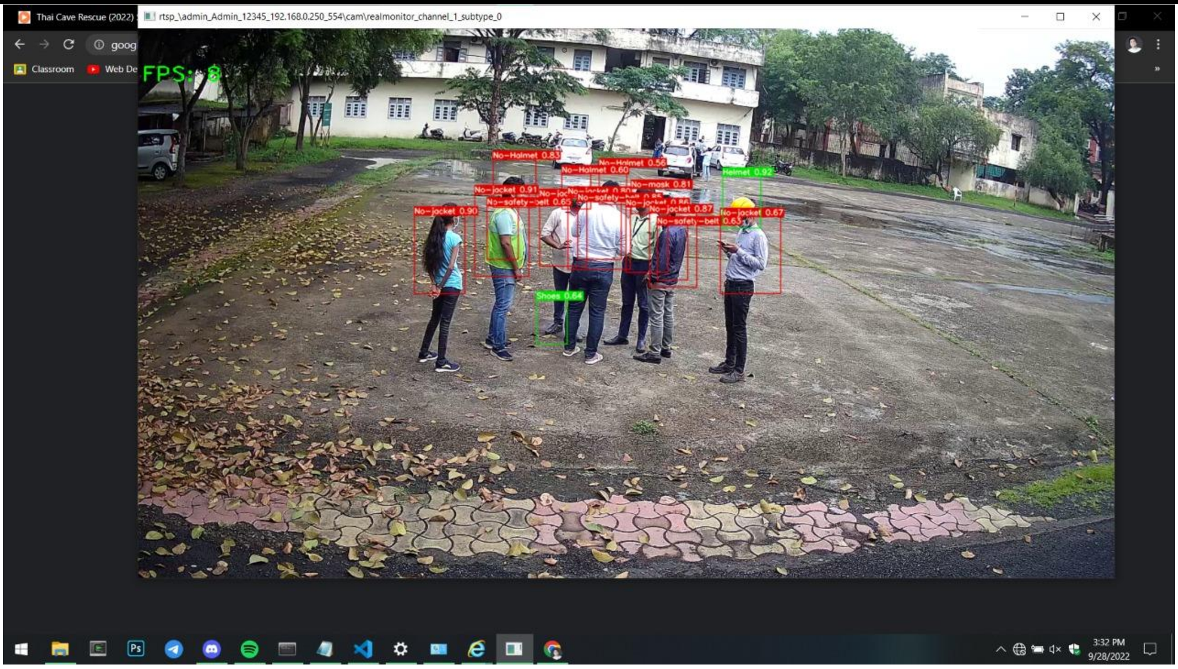


Fig no 8 . Output on Video Feed

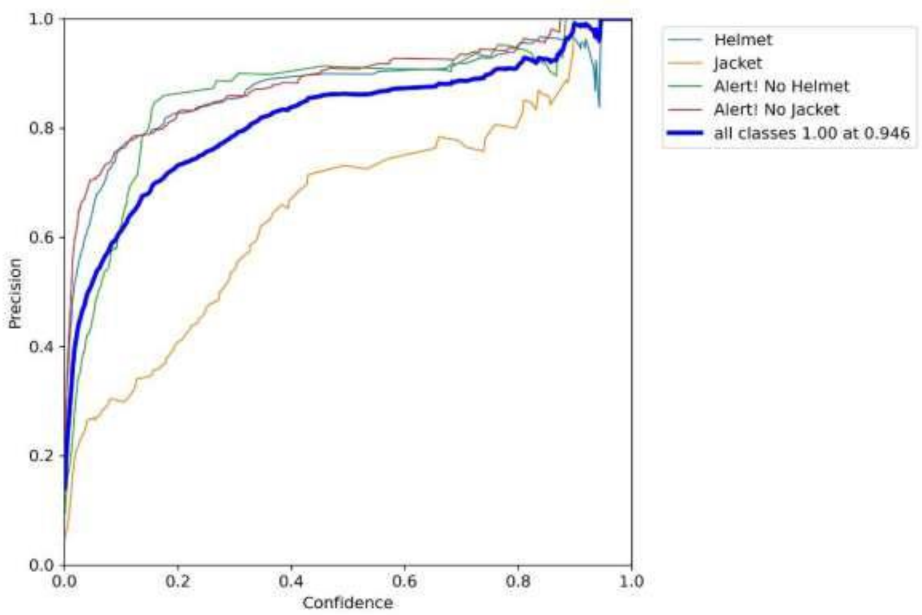
2. Performance of Custom Dataset

Calculating model performance to analyze a new dataset gives an idea of the actual performance of the model. The new dataset was created as a set of images of different wearable safety vests or helmets, of which were different from the ones the model was trained to recognize. A dataset completely new to the model that matched the PPE characteristics of the test dataset was developed through manual collection and image scraping. The observation model is run on the new data set and the results are and a confusion matrix was made. The classification report includes TP, TN, FP and FN , precision, recall and model F1 scores for each class. The models had a precision of 96.92%, an average precision of 0.98, a recall of of 0.95, and an F1 score of 0.96 as well. Given this , the model accurately predicted 96D with the new data set. The accuracy of the current model can be compared with models previously developed in previous studies using the RCNN and SSD algorithms. The detection speed is also comparable to previous models developed to detect devices using the faster R-CNN network.

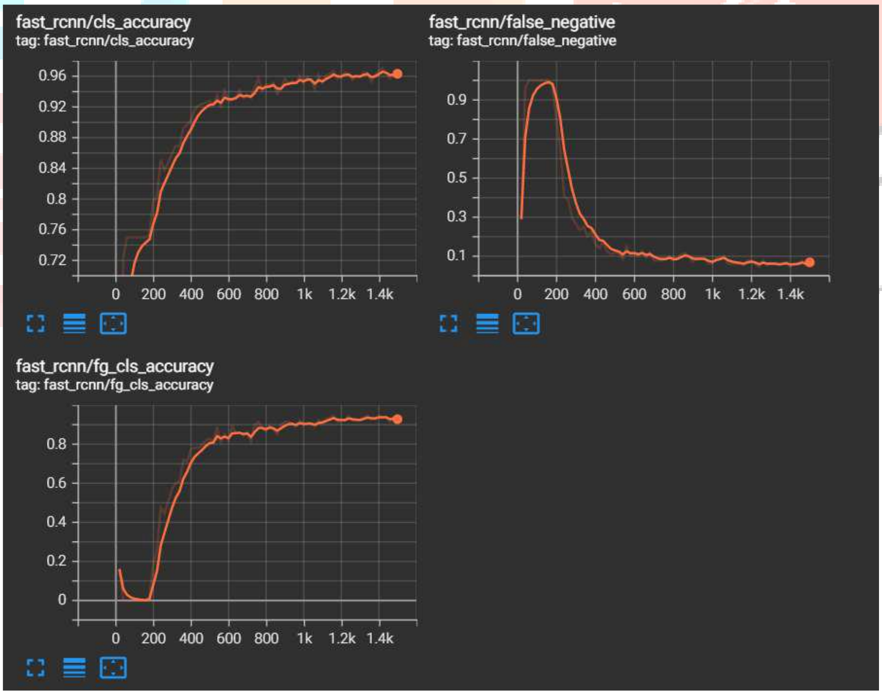
Fig no 9. Output on Custom Dataset

3. Outputs on Graphical Performance

i. YOLOv7



ii. Faster RCNN



4. Discussion

In this Comprehensive study on two algorithms namely : Faster RCNN , and YOLOv7 we Compared their Inference Speeds , Performance and map accuracies .Even though the accuracy of YOLOv7 is higher in comparison to the accuracy of Faster RCNN model . This study demonstrated the usefulness of deep learning based CV frameworks in the accurate monitoring of safety on construction sites. Furthermore, a small module could also be added to this framework to generate a regular report with screenshots of instances of noncompliance. These could be automated and sent out as daily/regular reports, enabling construction managers to escalate and address non-compliance and ensure on-site safety. The study also demonstrated the use of transfer learning in trained algorithms as a way to customize them to particular contexts. Techniques such as this are important in ensuring the scalability and application of the framework, enabling it to incorporate new functionalities such as detecting new classes of PPE, for example, should a job require new or more specialist equipment. There are some limitations to this study concerning the algorithm, which would benefit from a faster frame rate to enable real-time prediction. These research directions present exciting opportunities for enhancing safety in the construction industry. This study has presented a cutting edge deep learning-based computer vision algorithm with substantial implications and applications that opens up possibilities for incorporating machine intelligence that can automatically predict and monitor safety.

VII. CONCLUSION

This study used deep learning-based computer vision algorithms in the automated detection of the key processes that sustain construction safety and on-site management. Using YOLOv7and Faster Rcnn , a state of art object detection algorithm, this study demonstrates how safety compliance can be automatically detected by using a trained model to examine data from sites. The study demonstrated the deployment of such algorithms on construction sites to aid near real-time detection of safety violations. Systems such as this could be used to deploy similarly trained algorithms to process CCTV video footage from construction sites, generating a dashboard for real-time monitoring. The model developed in this study could be used in frameworks for regularly reporting non-compliance with safety regulations. This study demonstrated the usefulness of deep learning based CV frameworks in the accurate monitoring of safety on construction sites. Furthermore, a small module could also be added to this framework to generate a regular report with screenshots of instances of noncompliance. These could be automated and sent out as daily/regular reports, enabling construction managers to escalate and address non-compliance and ensure on-site safety. The study also demonstrated the use of transfer learning in trained algorithms as a way to customize them to particular contexts. Techniques such as this are important in ensuring the scalability and application of the framework, enabling it to incorporate new functionalities such as detecting new classes of PPE, for example, should a job require new or more specialist equipment. There are some limitations to this study concerning the algorithm, which would benefit from a faster frame rate to enable real-time prediction. The frame rate for processing the video images is about 2 fps which could be improved further by fine-tuning the hyperparameters of the algorithm. Moreover, the capacity for prediction suffers from the usual problems of occlusion and some color mismatching (especially for hard hats) which could be refined by using larger datasets in the future. The study also used supervised machine learning techniques, and future research would benefit from a combination of supervised and unsupervised techniques to generate more intelligent systems. The development of this business intelligence is an ongoing effort. The present models form one piece of larger frameworks, which could be evolved to completely automate safety monitoring without manual interventions. Such frameworks might use sensing technologies beyond vision to create an ability to understand the safety conditions in a more comprehensive manner. Future applications and technologies might make use of techniques such as the Internet of Things (IoT) and other big data to completely automate responses to hazards. These research directions present exciting opportunities for enhancing safety in the construction industry. This study has presented a cutting edge deep learning-based computer vision algorithm with substantial implications and applications that opens up possibilities for incorporating machine intelligence that can automatically predict and monitor safety.

VIII. ACKNOWLEDGMENT

The authors wish to thank HIXAA Technologies Pvt. Ltd. for providing CCTV , JETSON , PPE KIT for compilation of the dataset used for this on-site camera for live testing. They also want to acknowledge G. H. Raisoni Institute of Engineering and Technology, Nagpur, for providing access to NVIDIA GPU for faster and optimized training of the algorithms.

REFERENCES

[1] .T. Q. Vinh and N. T. N. Anh, "Real-Time Face Mask Detector Using YOLOv3 Algorithm and Haar Cascade Classifier," 2020 International Conference on Advanced Computing and Applications (ACOMP), Quy Nhon, Vietnam, 2020, pp. 146-149, doi: 10.1109/ACOMP50827.2020.00029

[2] Cavazza, N., and Serpe, A. (2009). Effects of safety climate on safety norm violations: exploring the mediating role of attitudinal ambivalence toward personal protective equipment. J. Safe. Res. 40, 277–283. doi: 10.1016/j.jsr.2009.06.002

[3] D.Podgórski, K. Majchrzycka, A. Dąbrowska, G. Gralewicz, M. Okrasa, Towards a Conceptual Framework of OSH Risk Management inSmart Working Environments Based on Smart PPE, Ambient Intelligence and the Internet of Things Technologies, International Journal of Occupational Safety and Ergonomics 23(2017)120.

[4] D. Podgórski, K. Majchrzycka, A. Dąbrowska, G. Gralewicz, M. Okrasa, Towards a Conceptual Framework of OSH Risk Management inSmart Working Environments Based on Smart PPE, Ambient Intelligence and the Internet of Things Technologies, International Journal of Occupational Safety and Ergonomics 23 (1) (2017) 1–20. doi:10.1080/10803548.2016.1214431.

[5] Bosche, F., Haas, C.T. and Akinci, B. (2009). Automatic 3D CAD object detection in laser scanning for project 3D spatial rendering and performance control. J. Calculate. Construction technology. 23,311-318. doi:10.1061 (ASCE) 0887-3801 (2009) 23:6 (311)