www.ijprems.com
editor@ijprems.com

INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)

Vol. 03, Issue 02, February 2023, pp : 181-192

e-ISSN : 2583-1062

Impact Factor : 5.725

A DEEP LEARNING-BASED METHOD TO DETECT SAFTEY HELMETWEARING AT A SATISFACTORY ACCURACY WITH HIGH DETECTION SPEED

A. Swetha[1], Gadam Reesham[2], Gantasala Sowjanya[3], Mallela Siri Chowdary[4]

[1]Assistant professor, department of computer science and engineering Teegala Krishna Reddy Engineering College, Hyderabad, Telangana, India.

[2,3,4]UG students , department of computer science and engineering, Teegala Krishna Reddy Engineering College, Hyderabad, Telangana, India.

ABSTRACT

Wearing safety helmets can effectively protect workers safety on construction sites. However, workers often take off the helmets because of weak security-conscious and discomfort, then hidden dangers will be brought by this behavior. Hence, detecting safety helmet wearing is a vital step of construction sites safety management and a safety helmet detector with high speed and accuracy is urgently needed. Therefore, this paper proposes a deep learning-based method to detect safety helmet wearing at a satisfactory accuracy with high detection speed. Our method chooses YOLO v5 as the baseline, then the fourth detection scale is added to predict more bounding boxes for small objects and the attention mechanism is adopted in the backbone of the network to construct more informative features for following concatenation operations. These results demonstrate the robustness and feasibility of our model. Which means the model is easy to be deployed? At last, after obtaining a satisfactory model, a graphical user interface (GUI) is designed to make our algorithm more user- friendly.

Keywords: safety, deep learning, accuracy.

1. INTRODUCTION

PURPOSE

These results demonstrate the robustness and feasibility of the model. Meanwhile, the size of the trained model is only 16.3 m, which means the model is easy to be deployed. At last, after obtaining a satisfactory model, a graphical user interface (GUI) is designed to make our algorithm more user-friendly.

SCOPE

This project proposes a deep learning-based method to detect safety helmet wearing at a satisfactory accuracy with high detection speed. This method chooses YOLO v5 as the baseline, then the fourth detection scale is added to predict more bounding boxes for small objects and the attention mechanism is adopted in the backbone of the network to construct more informative features for following concatenation operations. In order to overcome the defects caused by insufficient data, targeted data argumentation and transfer learning are used. Improvements caused by every modification are discussed in this paper. Finally, this model achieves 92.2% mean average precision, up 6.3% compared to the original algorithm, and it only takes 3.0ms to detect an image at 640×640.

PROBLEM STATEMENT

A safety helmet detector with high speed and accuracy is urgently required since seeing workers wearing safety helmets is an essential part of managing safety on construction sites. However, standard manual monitoring requires a lot of labour, and popularizing techniques for mounting sensors on safety helmets is challenging.

2. LITERATURE SURVEY

Deep learning has achieved impressive results in many machines learning tasks such as image recognition and computer vision. Its applicability to supervised problems is however constrained by the availability of high-quality training data consisting of large numbers of humans annotated examples (e.g. millions). To overcome this problem, recently, the AI world is increasingly exploiting artificially generated images or video sequences using realistic photo rendering engines such as those used in entertainment applications. In this way, large sets of training images can be easily created to train deep learning algorithms. The generated photorealistic synthetic image sets to train deep learning models to recognize the correct use of personal safety equipment (e.g., worker safety helmets, high visibility vests, ear protection devices) during at-risk work activities. The adaptation of the domain to real-world images using a very small set of real-world images. The demonstrated training with the synthetic training set generated and the use of the domain adaptation phase is an effective solution for applications where no training set is available.

www.ijprems.com
editor@ijprems.com

INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)

Vol. 03, Issue 02, February 2023, pp : 181-192

e-ISSN : 2583-1062

Impact Factor : 5.725

3. SYSTEM ANALYSIS

EXISTING SYSTEM

Various sensors were utilized for safety helmet wearing detection, such as chinstrap sensors, three- axis accelerometer sensor, radio frequency identification (RFID) and pressure sensor. However, these methods increase the detection investment and can be regarded as intrusion to workers. So, workers tend to be reluctant to wear safety helmet with above sensors because of privacy issues. As a result, non-intrusive methods are more acceptable, and among them, computer vision holds immense potential to detect safety helmet wearing condition. Many computers vision-based algorithms have been proposed to detect objects, such as Gaussian mixture model (GMM), histogram of oriented gradient (HOG), and support vector machine. These algorithms were also used in safety helmet detection to deal with the high fatality rates confronting the construction industry.

DISADVANTAGES OF EXISTING SYSTEM

- However, traditional manual monitor is labour intensive methods of installing sensors on safety helmet are difficult to popularize.
- Workers without safety helmets will suffer more injuries in accidents such as falling human body and vertical falling matter.

PROPOSED SYSTEM:

Therefore, this paper proposes a deep learning-based method to detect safety helmet wearing at a satisfactory accuracy with high detection speed. Our method chooses YOLO v5 as the baseline, then the fourth detection scale is added to predict more bounding boxes for small objects and the attention mechanism is adopted in the backbone of the network to construct more informative features for following concatenation operations. In order to overcome the defects caused by insufficient data, targeted data augmentation and transfer learning are used.

ADVANTAGES OF PROPOSED SYSTEM: 1.These results demonstrate the robustness and feasibility of our model.

REQUIREMENT SPECIFICATIONS:

HARDWARE REQUIREMENT SPECIFICATION:

Minimum hardware requirements are very dependent on the particular software being developed by a given though Python

/ Canopy / VS Code user. Applications that need to store large arrays/objects in memory will require more RAM, whereas applications that need to perform numerous calculations or tasks more quickly will require a faster processor.

- **Operating system: Windows 10**
- **Processor: Intel i5**
- **RAM :4GB**
- **HARD DISK: 250 GB**

SOFTWARE REQUIREMENTS:

The functional requirements or the overall description documents include the product perspective and features, operating system and operating environment, graphics requirements, design constraints and user documentation.

The appropriation of requirements and implementation constraints gives the general overview of the project in regards to what the areas of strength and deficit are and how to tackle them.

- **Operating System -Windows7/8**
- **Programming Language -Python**
- **7 3.Visual studio community version**
- **nodejs (version 12.3.1)**

FUNCTIONAL REQUIREMENTS:

Functional requirements are represented or stated in the form of input to be given to the system, the operation performed and the output expected.

System should collect the data from any resources. All the collected data should be processed for proper use, some analysis should be done for understanding the data properly.

Upload Helmet Dataset:

The machine should be able to know the recognition of the helmet so the dataset of helmets will be uploaded.

Image Pre-processing:

After the uploading the dataset the images of the dataset will be processed.

www.ijprems.com
editor@ijprems.com

INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)

Vol. 03, Issue 02, February 2023, pp : 181-192

e-ISSN : 2583-1062

Impact Factor : 5.725

Feature Extraction:

The feature of the images like the shape of the helmet the way of inserting the helmet is correct or not everything will be extracted.

Upload Test Image:

To detect the person whether he is wearing the helmet or not will be inspected by uploading the images.

Non-Functional Requirements:

Usability:

Usability is the main non-functional requirement for the Deep Learning-Based Workers Safety Helmet Wearing Detection on Construction Sites Using Multi-Scale Features the UI should be simple enough for everyone to understand and get the relevant information without any special training. Different languages can be provided based on the requirements.

Accuracy:

Accuracy is another important non-functional requirement for the Deep Learning-Based Workers Safety Helmet Wearing Detection on Construction Sites Using Multi-Scale Features.

The dataset is used to Train and Test Model in python. Prediction should be correct, consistent, and reliable.

Availability:

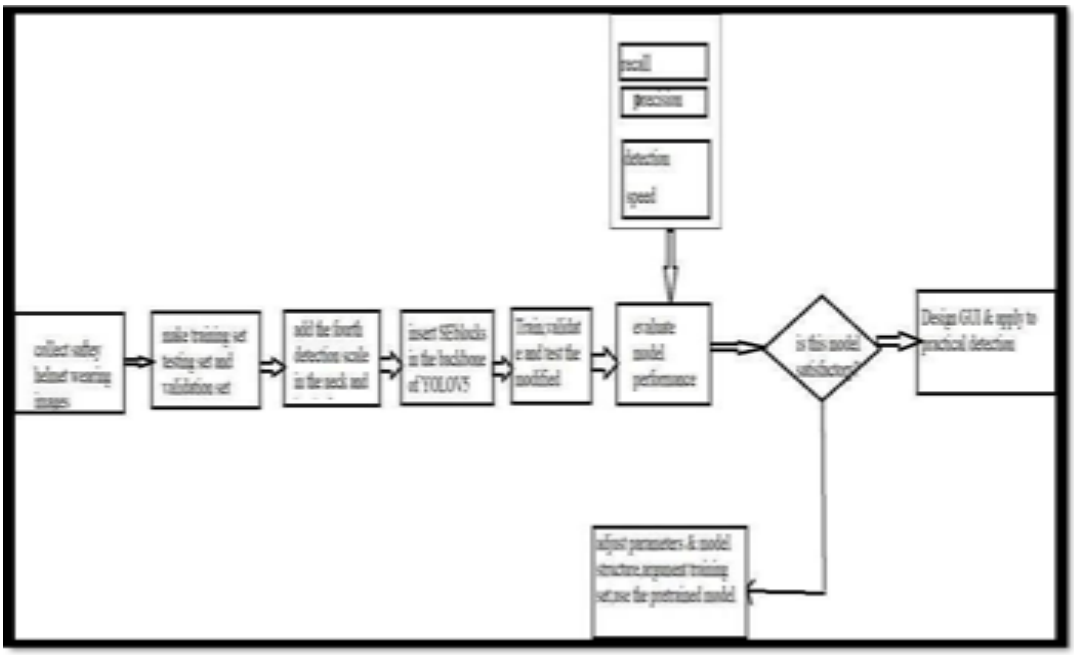
The System should be available for the duration when the user operates and must be recoveredwithin an hour or less if it fails. The system should respond to the requests within two seconds or less.

Maintainability:

The software should be easily maintainable and adding new features and making changes to the software must be as simple as possible. In addition to this, the software must also be portable.

SYSTEM DESIGN:

System Architecture:



This process includes collecting different helmet wearing images which is given in the form of dataset. Make the training set, testing set and validation set. We then add the fourth detection scale and then insert SE blocks in the backbone of YOLO V5. Train, Validate and test the modify Evaluate model performance with recall, precision, detection speed, and checks weather it is satisfactory or not if not, it will adjust parameters and model structure, argument training set, use the pertained model finally design GUI and apply the practical detection.

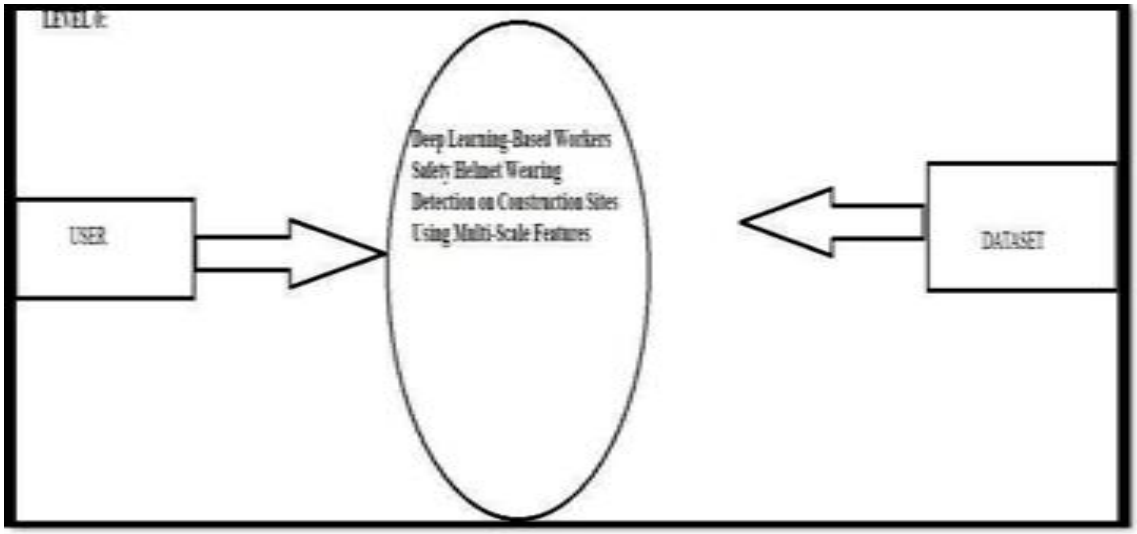
Data Flow Diagram:

The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of input data to the system, various processing carried out on this data, and the output data is generated by this system.

The data flow diagram (DFD) is one of the most important modeling tools. It is used to model the system components.

INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)
www.ijprems.com
editor@ijprems.com
Vol. 03, Issue 02, February 2023, pp : 181-192
e-ISSN : 2583-1062
Impact Factor : 5.725

These components are the system process, the data used by the process, an external entity that interacts with the system and the information flows in the system.



4. METHODOLOGY

Implementation Algorithm:

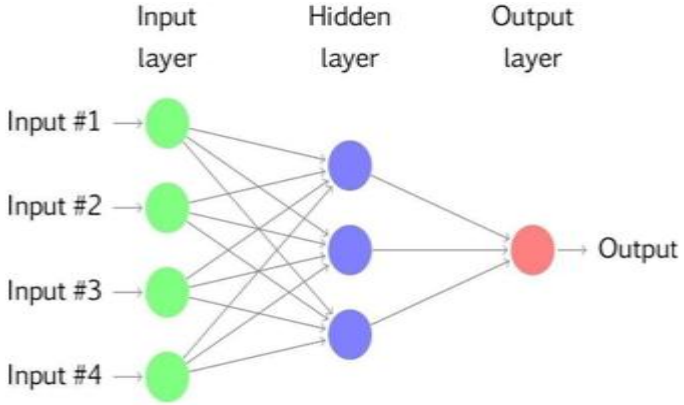
CNN (Convolution Neural Network):

To demonstrate how to build a convolutional neural network- based image classifier, we shall build a 6-layer neural network that will identify and separate one image from other. This network that we shall build is a very small network that we can run on a CPU as well. Traditional neural networks that are very good at doing image classification have many more parameters and take a lot of time if trained on normal CPU. However, our objective is to show how to build a real-world convolutional neural network using TENSORFLOW.

CNN is designed to automatically and adaptively learn spatial hierarchies of features through back propagation by using multiple building blocks, such as convolution layers, pooling layers, and fully connected layers.

Convolutional neural networks (CNNs) have been widely applied to many computer vision applications such as image classification, face recognition, object detection, and so on. This chapter introduces some of the most popular CNN architectures, including LeNet, AlexNet, VGG, GoogLeNet, and ResNet.They are made of neurons, the basic computation unit of neural networks. A neuron takes an input (say x), do some computation on it (say: multiply it with a variable w and adds another variable b) to produce a value (say; z= wx+b). This value is passed to a non-linear function called activation function (f) to produce the final output(activation) of a neuron. There are many kinds of activation functions. One of the popular activation function is Sigmoid. The neuron which uses sigmoid function as an activation function will be called sigmoid neuron. Depending on the activation functions, neurons are named and there are many kinds of them like RELU, TanH.

If you stack neurons in a single line, it's called a layer; which is the next building block of neural networks. See below image with layers to predict image class multiple layers operate on each other to get best match layer and this process continues till no more improvement left.



YOLO V3:

Software design sits at the technical kernel of the software engineering process is applied regardless of the

INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)

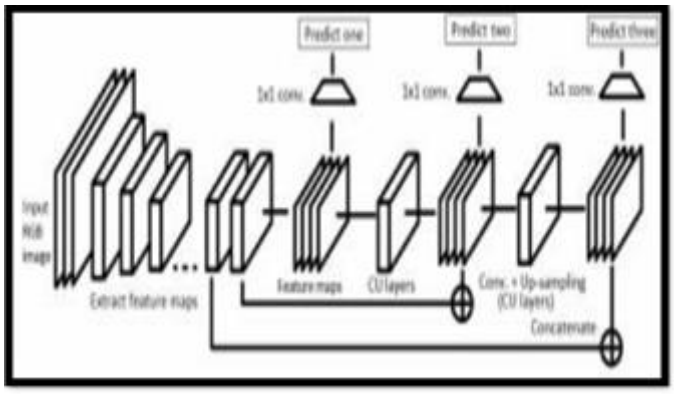
www.ijprems.com
editor@ijprems.com

Vol. 03, Issue 02, February 2023, pp : 181-192

e-ISSN : 2583-1062

Impact Factor : 5.725

development paradigm and area of application. Design is the first step in the development phase for any engineered product or system. The designer's goal is to produce a model or representation of an entity that will later be built. Beginning, once system requirement has been specified and analyzed, system design is the first of the three technical activities- design, code and test that is required to build and verify software.



SCHEMATIC REPRESENTATION OF YOLO V3 ARCHITECTURESOFTWARE ENVIRONMENT:

Python is currently the most widely used multi-purpose, high- level programming language. Python allows programming in Object-Oriented and Procedural paradigms. Python programs generally are smaller than other programming languages like Java.

Programmers have to type relatively less and indentation requirement of the language, makes them readable all the time. Python language is being used by almost all tech-giant companies like – Google, Amazon, Facebook, Instagram, Dropbox, Uber… etc.

The biggest strength of Python is huge collection of standard libraries which can be used for the following

- Machine Learning.
- GUI Applications (KiVy, Tkinter, PyOt etc.).
- Web Frame Works like Django (Used by YouTube, Instagram, Dropbox).
- Image processing (like OpenCV, Pillow)
- Web scraping (like Scrapy, Beautiful Soup, Selenium).
- Test frameworks
- Multimedia

What is Machine Learning?

Before we take a look at the details of various machine learning methods, let's start by looking at what machine learning is, and what it isn't. Machine learning is often categorized as a subfield of artificial intelligence, but I find that categorization can often be misleading at first brush. The study of machine learning certainly arose from research in this context, but in the data science application of machine learning methods, it's more helpful to think of machine learning as a means of building models of data.

Fundamentally, machine learning involves building mathematical models to help understand data. "Learning" enters the fray when we give these models tunable parameters that can be adapted to observed data; in this way the program can be considered to be "learning" from the data. Once these models have been fit to previously seen data, they can be used to predict and understand aspects of newly observed data. I'll leave to the reader the more philosophical digression regarding the extent to which this type of mathematical, model-based "learning" is similar to the "learning" exhibited by the human brain. Understanding the problem setting in machine learning is essential to using these tools effectively, and so we will start with some broad categorizations of the types of approaches we'll discuss here.

Categories of Machine Leaning: -

Machine learning can be categorized into two main types:

- **Supervised learning.**
- **Unsupervised learning.**

Supervised learning:

Supervised learning involves somehow modeling the relationship between measured features of data and some label associated with the data; once this model is determined, it can be used to apply labels to new, unknown data. This is further subdivided into classification tasks and regression tasks: in classification, the labels are discrete categories,

while in regression, the labels are continuous quantities. We will see examples of both types of supervised learning in the following section.

Unsupervised learning:

Unsupervised learning involves modeling the features of a dataset without reference to any label, and is often described as "letting the dataset speak for itself." These models include tasks such as clustering and dimensionality reduction. Clustering algorithms identify distinct groups of data, while dimensionality reduction algorithms search for more succinct representations of the data. We will see examples of both types of unsupervised learning in the following section.

Need for Machine Learning:

Human beings, at this moment, are the most intelligent and advanced species on earth because they can think, evaluate and solve complex problems. On the other side, AI is still in its initial stage and haven't surpassed human intelligence in many aspects. Then the question is that what is the need to make machine learn? The most suitable reason for doing this is, "to make decisions, based on data, with efficiency and scale".

Lately, organizations are investing heavily in newer technologies like Artificial Intelligence, Machine Learning and Deep Learning to get the key information from data to perform several real-world tasks and solve problems. We can call it data-driven decisions taken by machines, particularly to automate the process. These data-driven decisions can be used, instead of using programing logic, in the problems that cannot be programmed inherently. The fact is that we can't do without human intelligence, but other aspect is that we all need to solve real- world problems with efficiency at a huge scale. That is why the need for machine learning arises.

DEEP LEARNING:

Deep learning is a subset of machine learning, which is essentially a neural network with three or more layers. These neural networks attempt to simulate the behavior of the human brain—albeit far from matching its ability—allowing it to "learn" from large amounts of data. While a neural network with a single layer can still make approximate predictions, additional hidden layers can help to optimize and refine for accuracy.

Deep learning drives many artificial intelligence (AI) applications and services that improve automation, performing analytical and physical tasks without human intervention. Deep learning technology lies behind everyday products and services (such as digital assistants, voice-enabled TV remotes, and credit card fraud detection) as well as emerging technologies (such as self-driving cars). Another process called back propagation uses algorithms, like gradient descent, to calculate errors in predictions and then adjusts the weights and biases of the function by moving backwards through the layers in an effort to train the model. Together, forward propagation and back propagation allow a neural network to make predictions and correct for any errors accordingly. Over time, the algorithm becomes gradually more accurate.

Python:

Python is an interpreted high-level programming language for general- purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace.

Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library.

- Python is Interpreted − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
- Python is Interactive − you can actually sit at a Python prompt and interact with the interpreter directly to write your programs.

Python also acknowledges that speed of development is important. Readable and terse code is part of this, and so is access to powerful constructs that avoid tedious repetition of

code. Maintainability also ties into this may be an all but useless metric, but it does say something about how much code you have to scan, read and/or understand to troubleshoot problems or tweak behaviors. This speed of development, the ease with which a programmer of other languages can pick up basic Python skills and the huge standard library is key to another area where Python excels. All its tools have been quick to implement, saved a lot of time, and several of them have later been patched and updated by people with no Python background - without breaking.

www.ijprems.com
editor@ijprems.com

INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)

Vol. 03, Issue 02, February 2023, pp : 181-192

e-ISSN : 2583-1062

Impact Factor : 5.725

Libraries Used in Project:

Tensorflow:

TensorFlow is a free and open-source software library for dataflow and differentiable programming across a range of tasks. It is a symbolic math library, and is also used for machine learning applications such as neural networks. It is used for both research and production at Google.

TensorFlow was developed by the Google Brain team for internal Google use. It was released under the Apache 2.0 open- source license on November 9, 2015.

Besides its obvious scientific uses, NumPy can also be used as an efficient multi- dimensional container of generic data. Arbitrary data-types can be defined using NumPy which allows NumPy to seamlessly and speedily integrate with a wide variety of databases.

Pandas:

Pandas is an open-source Python Library providing high- performance data manipulation and analysis tool using its powerful data structures. Python was majorly used for data munging and preparation. It had very little contribution towards data analysis. Pandas solved this problem. Using Pandas, we can accomplish five typical steps in the processing and analysis of data, regardless of the origin of data load, prepare, manipulate, model, and analyze. Python with Pandas is used in a wide range of fields including academic and commercial domains including finance, economics, Statistics, analytics, etc.

Matplotlib:

Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can

be used in Python scripts, the Python and IPython shells, the Jupiter Notebook, web application servers, and four graphical user interface tool kits. Matplotlib tries to make easy things easy and hard things possible. You can generate plots, histograms, power spectra, bar charts, error charts, scatter plots, etc., with just a few lines of code. For examples, see the sample plots and thumbnail gallery.

For simple plotting the pyplot module provides a MATLAB- like interface, particularly when combined with IPython. For the power user, you have full control of line styles, font properties, axes properties, etc. via an object-oriented interface or via a set of functions familiar to MATLAB users.

Scikit – learn

Scikit-learn provides a range of supervised and unsupervised learning algorithms via a consistent interface in Python. It is licensed under a permissive simplified BSD license and is distributed under many Linux distributions, encouraging academic and commercial use.

SYSTEM TESTING:

Testing strategies:

Unit Testing:

Unit testing, a testing technique using which individual modules are tested to determine if there are issues by the developer himself. it is concerned with functional correctness of the standalone modules. The main aim is to isolate each unit of the system to identify, analyze and fix the defects.

Integration Testing:

Integration testing units or individual components of the software are tested in a group. The focus of this is to expose defects at the time of integrated components units. Once all the components or modules are working independent modules is known as integration testing.

Validation Testing:

The process of evaluating software during the development process or at the end of the development process to determine whether it satisfies specified Validation testing ensures that the product actually meets the client's needs. It can also be defining as to demonstrate that the product fulfill sits intended use when deployed on appropriate environment.

System Testing:

System testing is a technique performed to evaluate the complete system the system's compliance against specified requirements. It is usually carried out by a team that is independent of the development team in order to measure the quality of the system unbiased.

INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)

www.ijprems.com
editor@ijprems.com

Vol. 03, Issue 02, February 2023, pp : 181-192

e-ISSN : 2583-1062

Impact Factor : 5.725

S.NO	INPUT	OUTPUT	RESULT
Unit testing of Dataset	The user gives the input in the form of Helmet Dataset.	An output is Detect Person &Helmet Result.	A result is Detect Person &Helmet Result.
Unit testing of Accuracy	The user gives the input in the form of Helmet Dataset.	An output is Detect Person &Helmet Result.	A result is Detect Person & Helmet using YoloV5 algorithm got Accuracy up to 100%.
Unit testing of Machine Learning Algorithms	The user gives the input in the form of Helmet Dataset.	An output is Detect Person &Helmet Result.	A result is Detect Person & Helmet using YoloV5 algorithm got Accuracy up to 100%.
Integration testing of Dataset	The user gives the input in the form of image test data .	An output is Detect Person &Helmet Result.	A result is Detect Person & Helmet using YoloV5 algorithm got Accuracy up to 100%.
Big Bang testing	The user gives the input in the form of image test data .	An output is Detect Person &Helmet Result.	A result Detect Person & Helmet using DL Algorithm like YoloV5,ImageAI ,Faster RCNN& YoloV4 for Result.

CODING:

Import Libraries:

import numpy as np import pandas as pd

import xml.etree.ElementTree as ET from sklearn.model_selection import train_test_split import matplotlib.pyplot as plt from pathlib import Path import random import os import cv2from google.colab import drive drive.mount('/content/drive') base_path = '/content/drive/MyDrive/helmet/archive/' ann_path = base_path

+ 'Annotations/' img_path

= base_path + 'Images/'

Preparing The Training Dataframe:

def get_train_df(ann_path, img_path):

ann_path_list = get_file_list(ann_path, '.xml') ann_list

= []

for a_path in ann_path_list:

root = ET.parse(a_path).getroot() ann = {} ann['filename'] = Path(str(img_path) + '/'+ root.find("./filename").text) ann['width'] = root.find("./size/width").text ann['height'] = root.find("./size/height").text ann['class'] = root.find("./object/name").text ann['xmin']=int(root.find("./object/bndbox/xmin").text) ann['ymin']=int(root.find("./object/bndbox/ymin").text) ann['xmax'] =int(root.find("./object/bndbox/xmax").text ann['ymax'] = int(root.find("./object/bndbox/ymax").text) return pd.DataFrame(ann_list)

df_train = get_train_df(ann_path, img_path) yellow 147

white 131

red 90

none 79

blue 52

Name: class, dtype: int64

class_dict = {'blue': 0, 'none': 1, 'red': 2, 'white': 3, 'yellow': 3} idx_to_class

= {k:v for k,v in enumerate(list(class_dict.keys()))} df_train['class'] =

INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)
www.ijprems.com
editor@ijprems.com
Vol. 03, Issue 02, February 2023, pp : 181-192
e-ISSN : 2583-1062
Impact Factor : 5.725

df_train['class'].apply(lambda x: class_dict[x]) df_train.head()

Preparing Images Sizes And Bounding Boxes:

def read_image(path): return cv2.cvtColor(cv2.imread(str(path)), cv2.COLOR_BGR2RGB)

def create_bb_array(x): return np.array([x[5],x[4],x[7],x[6]])

def create_mask(bb, x): rows,cols,*_ = x.shape Y

= np.zeros((rows, cols))bb = bb.astype(np.int)

Y[bb[0]:bb[2], bb[1]:bb[3]] = 1. returnY

def mask_to_bb(Y): cols, rows = np.nonzero(Y) if len(cols)==0:

return np.zeros(4, dtype=np.float32)top_row = np.min(rows) left_col = np.min(cols) bottom_row = np.max(rows)right_col

= np.max(cols)

return np.array([left_col, top_row,right_col, bottom_row],dtype=np.float32) def

resize_image_bb(read_path, write_path, bb, sz): im = read_image(read_path)

im_resized = cv2.resize(im, (int(1.49*sz), sz))

Y_resized = cv2.resize(create_mask(bb, im), (int(1.49*sz), sz)) new_path =str(write_path/read_path.parts[-1]) cv2.imwrite(new_path, cv2.cvtColor(im_resized, cv2.COLOR_RGB2BGR)) return new_path, mask_to_bb(Y_resized)

Demo Image With Bounding Boxes

before transformation im = #cv2.imread(str(df_train.values[100][8])) im = cv2.cvtColor(im, cv2.COLOR_BGR2RGB) show_corner_bb(im, df_train.values[100][9]) # after transformation im, bb =

transformsXY(str(df_train.values[100][8]),df_train.values[100] [9],True) show_corner_bb(im,bb)

Training

def train_epocs(model, optimizer, train_dl, val_dl, epochs=10,C=1000):idx = 0 for I in range(epochs):

model.train() total = 0 sum_loss = 0 for x, y_class, y_bb in train_dl: batch = y_class.shape[0] x = x.cuda().float() y_class

= y_class.cuda() y_bb =y_bb.cuda().float() out_class, out_bb

= model(x) loss_class = F.cross_entropy(out_class, y_class, reduction="sum") loss_bb =F.l1_loss(out_bb, y_bb, reduction="none").sum(1) loss_bb = loss_bb.sum()loss

= loss_class + loss_bb/C optimizer.zero_grad() loss.backward() optimizer.step()

idx += 1 total += batch sum_loss

+= loss.item()

train_loss = sum_loss/total val_loss,val_acc =val_metrics(model, valid_dl, C) if (i+1) % 10 == 0: print("i:%4d train_loss:%5.3f val_loss:%5.3f val_acc:%5.3f"

% ((i+1),train_loss, val_loss, val_acc)) return sum_loss/total

YOLO V5

train_image_path = "/content/drive/MyDrive/helmet/train/images" train_label_path = "/content/drive/MyDrive/helmet/train/labels" val_image_path

= "/content/drive/MyDrive/helmet/valid/images" val_label_path = "/content/drive/MyDrive/helmet/valid/labels" if notos.path.isdir(train_image_path): os.makedirs(train_image_path)

if not os.path.isdir(train_label_path): os.makedirs(train_label_path)

if not os.path.isdir(val_image_path): os.makedirs(val_image_path)if not os.path.isdir(val_label_path): os.makedirs(val_label_path) # cycle for train dir for x in range(count_for_train): file_jpg

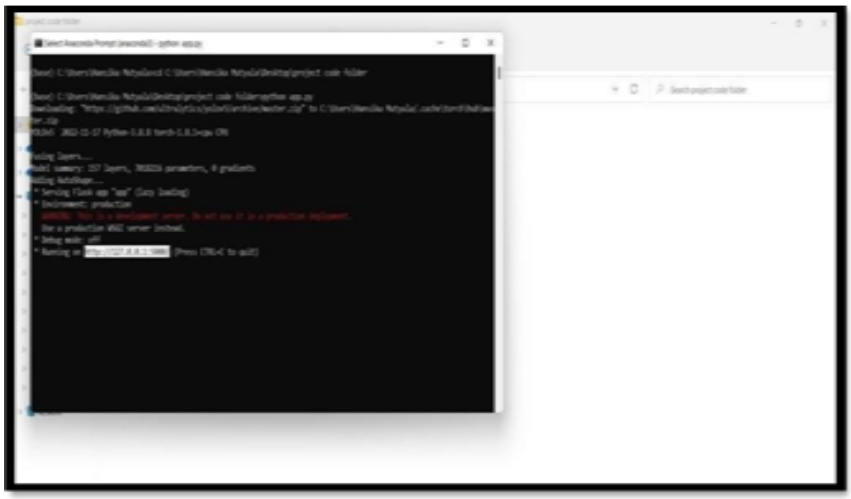
= choice(imgs) file_xml =file_jpg[:-4] + ".txt"

INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)
www.ijprems.com
editor@ijprems.com
Vol. 03, Issue 02, February 2023, pp : 181-192
e-ISSN : 2583-1062
Impact Factor : 5.725

shutil.copy(os.path.join(crs_path,file_jpg),os.path.join(train_i mage_path,file_j pg))
shutil.copy(os.path.join(crs_path,file_xml),os.path.join(train_l abel_path,file_x ml)) imgs.remove(file_jpg)
xmls.remove(file_xml) # cycle for test dir for x in range(count_for_val): file_jpg = choice(imgs) file_xml = file_jpg[:-4] + ".txt" shutil.copy(os.path.join(crs_path,file_jpg),os.path.join(val_im age_path,file_jp g))
shutil.copy(os.path.join(crs_path,file_xml),os.path.join(val_la bel_path,file_x ml))
imgs.remove(file_jpg) xmls.remove(file_xml)# rest of files print("images length
- ",len(imgs)) print(imgs,xmls) forx in imgs: file_jpg = x file_xml = file_jpg[:-4] + ".txt"
shutil.copy(os.path.join(crs_path,file_jpg),os.path.join(val_im age_path,file_j
pg))shutil.copy(os.path.join(crs_path,file_xml),os.path.join(va l_label_path,fi le_xml))

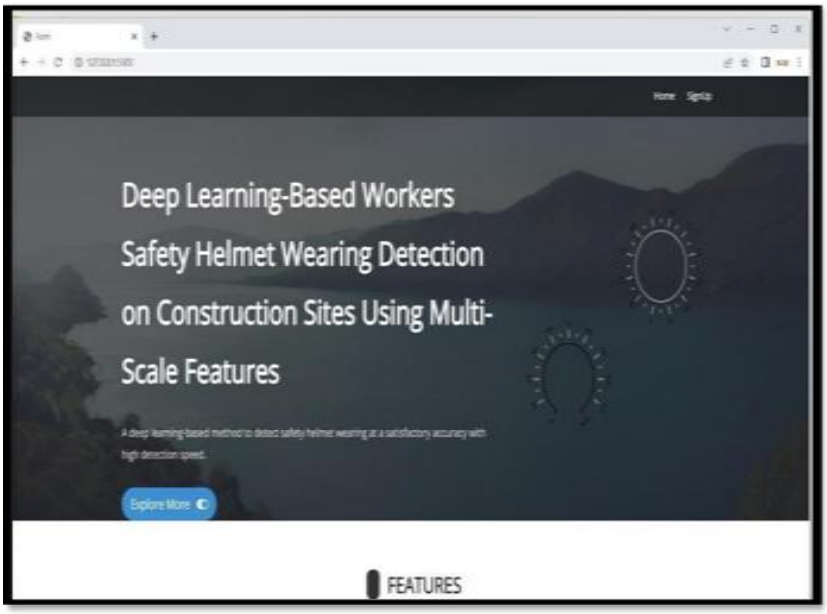
5. RESULT AND DISCUSSION

OUTPUT SCREENS:



URL Screen

Anaconda prompt showing various details to go through it. There is link which is related to collab, pasting the link (which is given above) in google. Users can able to access the Homepages.



Home page

Homepages where users can access to login. In homepage we can or users can sign by using their username and password. If user is a new user, he/she login by using signup option.

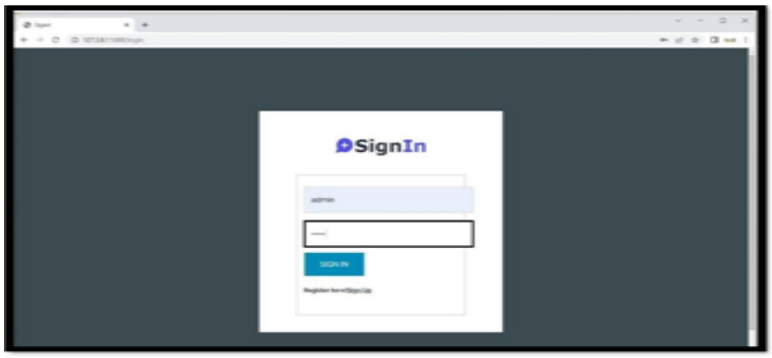
INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)

www.ijprems.com
editor@ijprems.com

Vol. 03, Issue 02, February 2023, pp : 181-192

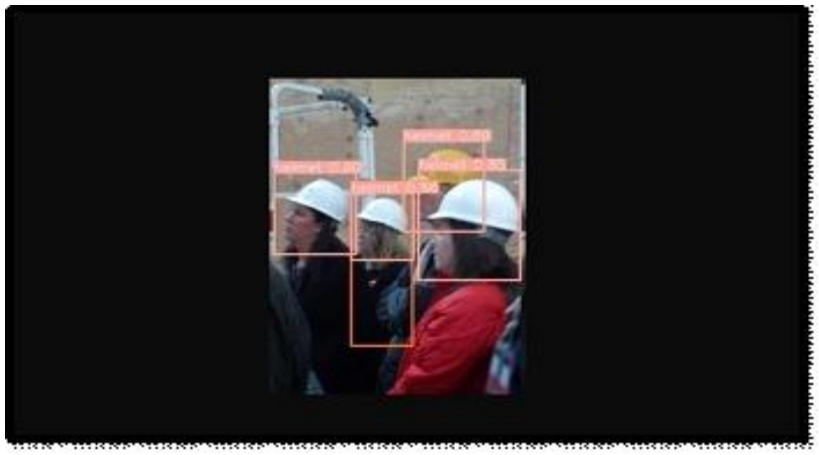
e-ISSN : 2583-1062

Impact Factor : 5.725



User can login through username and password

Web page



Detecting the Helmet

After logging in, our application will be opened. At top of the page, there is helmet detection and YOLOV5. We can selection any of those two for detecting of helmet.

6. CONCLUSION

In the proposed system a fast, accurate and easy-to-use method to detect safety helmet wearing condition. And the main contributions are as follows:

- We design the fourth detection scale based on YOLO v5 s. This modification can help reduce the missed and false detection caused by the challenges of small objects and scale variances.
- Two SE blocks was added in the backbone of the modified model to further improve accuracy at negligible additional computational cost.
- Targeted data augmentation and pre-trained model were adopted to overcome the defects caused by insufficient data.
- A GUI is designed to make our method more user-friendly after obtaining the satisfactory model, which means our model can be applied to practical engineering easily.

7. FUTURE ENHANCEMENT

However, in this we are just detecting weather the worker is wearing the helmet or not but what if the worker is not wearing the helmet? So, in the future enhancement we are going to remind the workers that who is not wearing the helmet.

Finally, reminding workers to wear safety helmet on construction sites in time will be investigated in the future.

8. REFERENCES

[1] M. D. Benedetto, F. Carrara, E. Meloni, G. Amato, and C. Gennaro, ‘‘Learning accurate personal protective equipment detection from virtual worlds,’’ Multimedia Tools Appl., vol. 80, pp. 1–13, Aug. 2020.

[2] Y. Li, H. Wei, Z. Han, J. Huang, and W. Wang, ‘‘Deep learning- based safety helmet detection in engineering management based on convolutional neural networks,’’ Adv. Civil Eng., vol. 2020, pp. 1– 10, Sep. 2020.

[3] R. R. Cabahug, ‘‘A survey on the implementation of safety standards of on-going construction projects in Cagayan de Oro City, Philippines,’’ Mindanao J. Sci. Technol., vol. 12, no. 1, pp. 12–24, 2014.

[4] B. Wang, W. Li, and H. Tang, ‘‘Improved YOLOv3 algorithm and its application in helmet detection,’’ Comput. Eng. Appl., vol. 56, no. 9, pp. 33–40, 2020.

[5] S. H. Kim, C. Wang, S. D. Min, and S. H. Lee, ‘‘Safety helmet wearing management system for construction workers using three- axis accelerometer sensor,’’ Appl. Sci., vol. 8, no. 12, p. 2400, Nov. 2018.

[6] A. Kelm, L. Laußat, A. Meins-Becker, D. Platz, M. J. Khazaee,

[7] Chenchireddy, Kalagotla, V. Jegathesan, and L. Ashok Kumar. "Different topologies of inverter: a literature survey." Innovations in Electrical and Electronics Engineering: Proceedings of the 4th ICIEEE 2019 (2020): 35-43.

[8] A. M. Costin, M. Helmus, and J. Teizer, ‘‘Mobile passive radio frequency identification (RFID) portal for automated and rapid control of personal protective equipment (PPE) on construction sites,’’ Autom.Construct., vol. 36, pp. 38–52, Dec. 2013.

[9] S. Dong, Q. He, H. Li, and Q. Yin, ‘‘Automated PPE misuse identification and assessment for safety performance enhancement,’’ in Proc. Int. Conf. Construct. Real Estate Manage., Sep. 2015, pp. 204– 214.

[10] Chenchireddy, Kalagotla, and V. Jegathesan. "A Review Paper on the Elimination of Low-Order Harmonics in Multilevel Inverters Using Different Modulation Techniques." Inventive Communication and Computational Technologies: Proceedings of ICICCT 2020 (2021): 961-971.

[11] Chenchireddy, K., Goud, B. S., Mudhiraj, C. M. S., Rajitha, N., Kumar, B. S., & Jagan, V. (2022, March). Performance Verification of Full-Bridge DC To DC Converter Used for Electric Vehicle Charging Stations. In 2022 8th International Conference on Advanced Computing and Communication Systems (ICACCS) (Vol. 1, pp. 434-439). IEEE.