Internship Report

On

AI

Submitted to the **Savitribai Phule Pune University** In partial fulfilment for the award of the Degree of **Bachelor of Engineering**

In

# Artificial Intelligence & Machine Learning

By

## Saurabh sanap (T190813426)

Under the guidance of

Prof. Kirti Randhe



Artificial Intelligence & Machine Learning Department

# ISBM College of Engineering Nande, Pune

**Academic Year 2022-2023**

1 | P a g e



# CERTIFICATE

This is to certify that the Internship Report entitled “AI” being submitted by

## Saurabh sanap (T190813426)

is a record of bona fide work carried out by him under the supervision and guidance of Prof. Kirti Randhe in partial fulfillment of the requirement for **TE (Artificial Intelligence & Machine Learning)** a 2019 course of Savitribai Phule Pune University, Pune in the academic year 2022- 2023.

|  |  |  |
| --- | --- | --- |
| Date: / /2023 |  |  |
| Place: PUNE |  |  |
| Prof. Kirti Randhe | Prof. Kirti Randhe | Dr. P.K.Srivastava |
| Guide | HOD | Principal |

This Internship Report has been examined by us as per the Savitribai Phule Pune University, Pune requirements at ISBM College of Engineering, Nande Pune on 2022-2023.

Internal Examiner External Examiner

**INTERNSHIP COMPLETION CERTIFICATE**





# ACKNOWLEDGEMENT

I feel great pleasure in expressing my deepest sense of gratitude and sincere thanks to my guide Prof. Kirti Randhe for their valuable guidance during the Internship work, without which it would have been very difficult task. I have no words to express my sincere thanks for valuable guidance, extreme assistance and cooperation extended to all the Staff Members of my Department.

This acknowledgement would be incomplete without expressing my special thanks to Prof. Kirti Randhe Head of the Department (AI & ML) for their support during the work.

I would also like to extend my heartfelt gratitude to my Principal, Dr.

P.K. Srivastava who provided a lot of valuable support, mostly being behind the veils of college bureaucracy.

Last but not least I would like to thanks all the Teaching, Non- Teaching staff members of my department, my parents and my colleagues those who helped me directly or indirectly for completing of this Internship successfully.

Saurabh sanap

# TABLE OF CONTENTS

Title Page 1

[Certificate 2](#_TOC_250007)

Internship Complication Certificate 3

[Acknowledgement 4](#_TOC_250006)

Learning Objectives/Internship Objectives 6

[Introduction 7](#_TOC_250003)

[Internship Place Details 8](#_TOC_250004)

Weekly overview of internship activity 9

Software requirement specification 12

Motivation and Rationale of the Study 13

[Methodological Details 14](#_TOC_250002)

Technology 15

Project 19

[Results and Conclusion 27](#_TOC_250001)

[List of Reference 2](#_TOC_250000)8

# Learning Objectives/Internship Objectives

Internships are generally thought of to be reserved for college students looking to gain experience in a particular field. However, a wide array of people can benefit from Training Internships in order to receive real world experience and develop their skills.

An objective for this position should emphasize the skills you already possess in the area and your interest in learning more

Internships are utilized in a number of different career fields, including architecture, engineering, healthcare, economics, advertising and many more.

Some internship is used to allow individuals to perform scientific research while others are specifically designed to allow people to gain first-hand experience working.

Utilizing internships is a great way to build your resume and develop skills that can be emphasized in your resume for future jobs. When you are applying for a Training Internship, make sure to highlight any special skills or talents that can make you stand apart from the rest of the applicants so that you have an improved chance of landing the position.

**INTRODUCTION**

During my internship at Pantech solutions, I was exposed to various aspects of Data Analysis. The company has a strong focus on Data scince and has a team of experienced professionals who provided me with the guidance and support needed to learn and grow in this field.

In the first month of my internship, I learned the fundamentals of Python, which is a popular programming language used extensively in Data science and machine learning. I gained an understanding of basic programming concepts and how they can be applied to solve real-world problems. I delved deeper into Python and learned about various libraries such as NumPy, SciPy and Pandas as well as scikitlearn and matplotlib. These libraries are commonly used in Data manipulation and different file handling as well as building machine learning model and visualizing the data graphically, which are important areas of Data Analysis.

In the second month, I learned different analysis technique. Then implement those statistical techniques on real life problem. In shortly working on Image Caption generator Project using Python. This project involved building a machine learning model to generate the caption based on the images. It was a challenging project that allowed me to apply the knowledge and skills I had gained during the previous months of the internship. I used tools like Google Colab and Jupyter Notebook to develop the project and referred to the stream lit documentation for building the User Interface. Additionally, I used Py-Spark, a cloud-based platform, to deploy the project.

Overall, my internship at Pantech solutions was a valuable learning experience. I gained hands-on experience in Data Science and had the opportunity to work on a real-world project. The knowledge and skills I gained during the internship will be invaluable in my future career as a Data Scientist as well as Machine Learning.

**INTERNSHIP PLACE DETAILS**

**Internship Title:** Internship on AI

**Internship Duration:** Jan 24th to March 25th (02 Months)

**Name Of Company:** PANTECH SOLUTIONS

**Internship Place:** Perungudi,Chennai - 600 096,India

**Company Background**: Pantech Solutions Private Limited is an Indian technology company that specializes in providing end-to-end embedded system solutions and industrial IoT (Internet of Things) services. The company was established in 2003 and is headquartered in Chennai, India, with additional offices in the USA and Germany.Pantech Solutions offers a range of products and services, including embedded system development, PCB design, FPGA-based design, IoT gateway solutions, and cloud-based services. The company has a strong focus on innovation and research and development and has partnerships with leading technology providers such as Xilinx, Texas Instruments, and Intel.Pantech Solutions has served clients across a variety of industries, including aerospace, defense, automotive, healthcare, and education. The company has a strong reputation for delivering high-quality, reliable, and cost-effective solutions to its clients and has received numerous awards and accolades for its contributions to the technology industry in India.

Overall, Pantech Solutions has established itself as a leading player in the embedded systems and IoT space and continues to innovate and evolve its offerings to meet the evolving needs of its clients and the technology industry as a whole.

**Supervisor Details:**

My supervisor for the internship was **Sanjay kumar**, who is the Expertise in Data analysis**.** he has 10+ years of training experience in Open-Source Technologies like Python, Databases & Machine learning/Data Analysis.

**WEEKLY OVERVIEW OF INTERNSHIP ACTIVITIES**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **PANTECH SOLUTIONS** | | | | | |
| **SAURABH BHAUSAHEB SANAP** | | | | | |
| **INTERNSHIP ON AI** | | | | | |
| **INTERNSHIP LOGBOOK** | | | | | |
| **Week 1** | Date | Day | Topic | Sign | Internal Sign |
| 1/23/2023 | Monday | Introduction Platforms Requirement for online Internship(Hardware, Software) Require applications(Operating system,IDE) etc.. |  |  |
| 1/24/2023 | Tuesday |
| 1/25/2023 | Wednesday |
| 1/26/2023 | Thursday |
| 1/27/2023 | Friday |
| 1/28/2023 | Saturday |
| **INTERNSHIP REPORT** | | | | | |
| **Week 2** | Date | Day | Topic | Sign | Internal Sign |
| 1/30/2023 | Monday | Introduction about python  Installing Python  Python data structure |  |  |
| 1/31/2023 | Tuesday |  |
| 2/1/2023 | Wednesday |  |
| 2/2/2023 | Thursday |  |
| 2/3/2023 | Friday |  |
| 2/4/2023 | Saturday |  |
| **INTERNSHIP REPORT** | | | | | |
| **Week 3** | Date | Day | Topic | Sign | Internal Sign |
| 2/6/2023 | Monday | List vs Set vs Tuple vs Dictionary Python statement Problem solving |  |  |
| 2/7/2023 | Tuesday |
| 2/8/2023 | Wednesday |
| 2/9/2023 | Thursday |
| 2/10/2023 | Friday |
| 2/11/2023 | Saturday |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **INTERNSHIP REPORT** | | | | | |
| **Week 4** | Date | Day | Topic | Sign | Internal Sign |
| 2/13/2023 | Monday | Introduction to AI  Machine Learning Deep Learning Natural Language Processing (NLP) Computer Vision |  |  |
| 2/14/2023 | Tuesday |
| 2/15/2023 | Wednesday |
| 2/16/2023 | Thursday |
| 2/17/2023 | Friday |
| 2/18/2023 | Saturday |
| **INTERNSHIP REPORT** | | | | | |
| **Week 5** | Date | Day | Topic | Sign | Internal Sign |
| 2/20/2023 | Monday | Reinforcement Learning Neural Networks Expert Systems Decision Trees Support Vector Machines (SVM) |  |  |
| 2/21/2023 | Tuesday |
| 2/22/2023 | Wednesday |
| 2/23/2023 | Thursday |
| 2/24/2023 | Friday |
| 2/25/2023 | Saturday |
| **INTERNSHIP REPORT** | | | | | |
| **week 6** | Date | Day | Topic | Sign | Internal Sign |
| 2/27/2023 | Monday | Introduction to Computer vision & its Libraries OpenCV (Open Source Computer Vision Library) TensorFlow  PyTorch |  |  |
| 2/28/2023 | Tuesday |
| 3/1/2023 | Wednesday |
| 3/2/2023 | Thursday |
| 3/3/2023 | Friday |
| 3/4/2023 | Saturday |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **INTERNSHIP REPORT** | | | | | |
| **Week 7** | Date | Day | Topic | Sign | Internal Sign |
| 3/6/2023 | Monday | Introduction to Deep Learning & it’s libraries  Face detection & Tracking |  |  |
| 3/7/2023 | Tuesday |
| 3/8/2023 | Wednesday |
| 3/9/2023 | Thursday |
| 3/10/2023 | Friday |
| 3/11/2023 | Saturday |
| **INTERNSHIP REPORT** | | | | | |
| **Week 8** | Date | Day | Topic | Sign | Internal Sign |
| 3/13/2023 | Monday | Reinforcement Learning AI Snake Game Road Sign Recognition |  |  |
| 3/14/2023 | Tuesday |
| 3/15/2023 | Wednesday |
| 3/16/2023 | Thursday |
| 3/17/2023 | Friday |
| 3/18/2023 | Saturday |
| **INTERNSHIP REPORT** | | | | | |
| **Week 9** | Date | Day | Topic | Sign | Internal Sign |
| 3/20/2023 | Monday | AI Hardware AI Cloud |  |  |
| 3/21/2023 | Tuesday |
| 3/22/2023 | Wednesday |
| 3/23/2023 | Thursday |
| 3/24/2023 | Friday |
| 3/25/2023 | Saturday |

**SOFTWARE REQUIREMENTS SPECIFICATIONS**

**System configurations**

The software requirement specification can produce at the culmination of the analysis task. The function and performance allocated to software as part of system engineering are refined by established a complete information description, a detailed functional description, a representation of system behavior, and indication of performance and design constrain, appropriate validate criteria, and other information pertinent to requirements.

**Software Requirements:** Pycharm IDE, Python,

**Operating system** : Windows 11 Ultimate. Coding Language : Python

Front-End : Visual Studio Professional, Google colab, Jupyter notebook

Data Base : SQL .

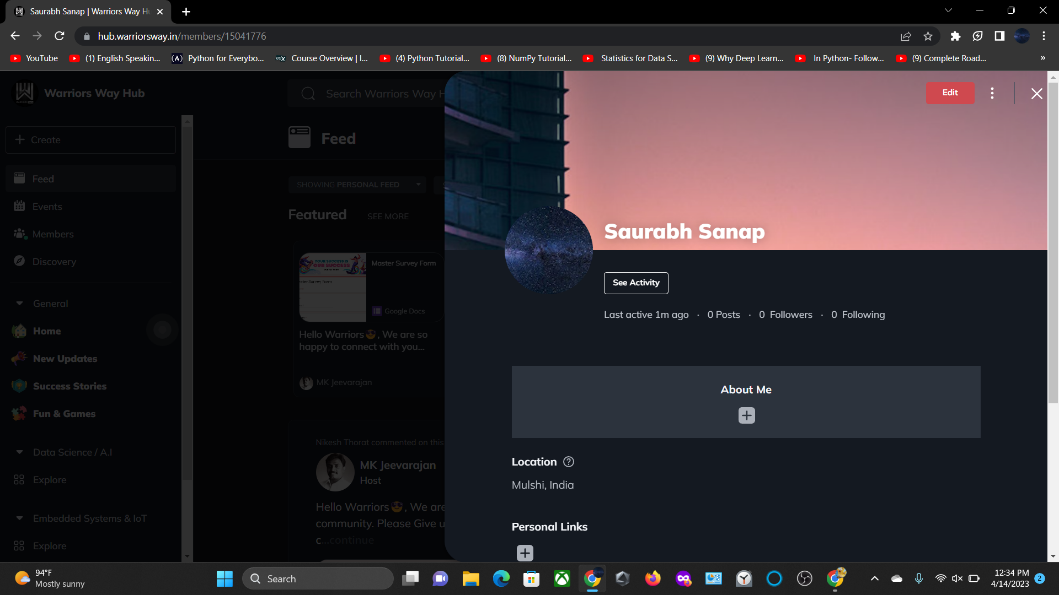
**Hardware Requirement**:

System : intel i5.

Hard Disk : 1TB.

Ram : 8GB.

Platform for Internship



# MOTIVATION AND RATIONAL OF THE STUDY

The motivation behind this study was to enhance my knowledge and skills in software development and to apply them to real-world problems. The scope of the study was to learn and understand the basics of Python programming, Data Analysis Process, and different Machine Learning Libraries, and to implement this knowledge in a practical project. The rationale behind this study was to build a tool that could be used to analysis the data and predict the output, which is a common problem faced by many individuals and organizations.

**METHODOLOGICAL DETAILS**

The methodology of my internship at Pantech solutions was structured in two phases. During the first month, I was introduced to the fundamentals of Python programming, including syntax, data structures, and control structures. I was also introduced to object-oriented programming concepts and principles.

In the second month, I learned various libraries commonly used in Data science and machine learning model development. These included Matplotlib, which is used for Visualizing data graphically, pickle, which is used for object serialization, and sklearn, which is a popular Machine Learning library. I learned how to use these libraries to preprocess data, train Machine Learning Models, and evaluate their performance.

Throughout the internship, I followed a structured approach to learning and developing Machine Learning Model, which included researching and understanding the problem statement, breaking down the problem into smaller sub-tasks, designing and implementing solutions for each sub-task, testing and evaluating the solution, and documenting the entire process.

Overall, the methodology of the internship was designed to provide a comprehensive learning experience that included both theoretical and practical aspects of Data science, from fundamental programming concepts to advanced Machine Learning Techniques.

**TECHNOLOGY**

**Python**

Python programming language is widely used in the field of Artificial Intelligence (AI) due to its simplicity, ease of use, and flexibility. Python provides a vast array of libraries and frameworks that can be used for developing AI applications, including machine learning, natural language processing, computer vision, and more.

Python is an open-source language, which means it is free to use, distribute, and modify. It is available on a variety of platforms, including Windows, Mac OS, and Linux, and has a vast array of libraries and frameworks that make it useful for a wide range of applications.

Python has a vast array of libraries and frameworks that make it a powerful language for a variety of applications. Here are some of the most important libraries in Python:

* NumPy: NumPy is a library for scientific computing that provides support for large, multi-dimensional arrays and matrices. It also provides a variety of mathematical functions for working with these arrays.
* Pandas: Pandas is a library for data manipulation and analysis that provides data structures for efficiently storing and manipulating large datasets.
* Matplotlib: Matplotlib is a library for data visualization that provides a variety of plot types and customization options.
* Scikit-learn: Scikit-learn is a library for machine learning that provides a variety of algorithms for classification, regression, and clustering.
* TensorFlow: TensorFlow is a library for machine learning that provides a variety of tools for building and training neural networks.
* Keras: Keras is a high-level API for building and training deep learning models that can run on top of TensorFlow, Theano, or CNTK.
* OpenCV: OpenCV is a library for computer vision that provides tools for image and video processing, object detection, and feature extraction.
* NLTK: NLTK is a library for natural language processing that provides tools for tokenization, stemming, and sentiment analysis.
* In Python, a module is a file that contains a collection of related functions, classes, and variables. Here are some of the most important modules in Python:
* math: The math module provides a variety of mathematical functions such as trigonometric functions, logarithmic functions, and more.
* os: The os module provides a variety of functions for interacting with the operating system, such as creating and deleting files, working with directories, and more.
* sys: The sys module provides access to system-specific parameters and functions, such as command-line arguments, and the ability to exit the program.
* datetime: The datetime module provides functions for working with dates, times, and time intervals.
* random: The random module provides functions for generating random numbers, shuffling lists, and more.
* csv: The csv module provides tools for working with comma-separated values (CSV) files.
* re: The re module provides regular expression matching operations for working with strings.
* socket: The socket module provides low-level networking tools for creating client-server applications.

**Google colab**

Google Colab is a free online platform that provides a Jupyter notebook environment with access to a GPU and TPU to execute machine learning and data analysis tasks. Colab allows users to write and run Python code using a browser, without having to install any software or hardware locally. Here are some of the features of Google Colab:

* Free access to a virtual machine with GPU and TPU support.
* Pre-installed libraries such as TensorFlow, Keras, PyTorch, OpenCV, and more.
* Ability to install additional libraries using pip or conda.
* Integration with Google Drive for storing and accessing data.
* Collaborative editing and sharing of notebooks with others.
* Ability to run code cells interactively and visualize results.
* Support for Markdown and LaTeX for creating rich text documents.
* Integration with GitHub for version control and collaboration.

**Machine learning steps:**

Problem Definition: Define the problem you want to solve and what you want to achieve with machine learning.

Data Collection: Gather relevant data from various sources such as databases, APIs, or web scraping.

Data Cleaning: Clean and preprocess the data to remove inconsistencies, missing values, and outliers.

Feature Selection: Select the most relevant features for the problem at hand.

Data Splitting: Split the data into training, validation, and testing sets.

Model Selection: Select the appropriate machine learning model that is best suited for the problem you are trying to solve.

Model Training: Train the machine learning model using the training dataset.

Model Validation: Validate the model using the validation dataset to check for overfitting and fine-tune the model.

Model Testing: Test the model using the testing dataset to evaluate its performance.

Model Deployment: Deploy the model in a production environment and monitor its performance.

Model Maintenance: Continuously update the model and retrain it with new data to improve its performance over time.

**PROJECTS**

**Face Detection In Python Using OpenCV**

**OpenCV**

OpenCV is an open source computer vision and machine learning software library. It is a BSD-licence product thus free for both business and academic purposes.The Library provides more than 2500 algorithms that include machine learning tools for classification and clustering, image processing and vision algorithm, basic algorithms and drawing functions, GUI and I/O functions for images and videos. Some applications of these algorithms include face detection, object recognition, extracting 3D models, image processing, camera calibration, motion analysis etc.

OpenCV is written natively in C/C++. It has C++, C, Python and Java interfaces and supports Windows, Linux, Mac OS, iOS, and Android. OpenCV was designed for computational efficiency and targeted for real-time applications. Written in optimized C/C++, the library can take advantage of multi-core processing.

**Face Detection**

Face detection has gained a lot of attention due to its real-time applications. A lot of research has been done and still going on for improved and fast implementation of the face detection algorithm. Why is face detection difficult for a machine? Face detection is not as easy as it seems due to lots of variations of image appearance, such as pose variation (front, non-front), occlusion, image orientation, illumination changes and facial expression.

OpenCV contains many pre-trained classifiers for face, eyes, smile etc. The XML files of pre-trained classifiers are stored in opencv/data/. For face detection specifically, there are two pre-trained classifiers:

Haar Cascade Classifier

LBP Cascade Classifier

We will explore both face detectors in this tutorial.

Haar Cascade Classifier

It is a machine learning based approach where a cascade function is trained from a lot of positive (images with face) and negative images (images without face). The algorithm is proposed by Paul Viola and Michael Jones.

**The algorithm has four stages:**

**Haar Feature Selection:** Haar features are calculated in the subsections of the input image. The difference between the sum of pixel intensities of adjacent rectangular regions is calculated to differentiate the subsections of the image. A large number of haar-like features are required for getting facial features.

**Creating an Integral Image:** Too much computation will be done when operations are performed on all pixels, so an integral image is used that reduce the computation to only four pixels. This makes the algorithm quite fast.

Adaboost: All the computed features are not relevant for the classification purpose. Adaboost is used to classify the relevant features.

**Cascading Classifiers:** Now we can use the relevant features to classify a face from a non-face but algorithm provides another improvement using the concept of cascades of classifiers. Every region of the image is not a facial region so it is not useful to apply all the features on all the regions of the image. Instead of using all the features at a time, group the features into different stages of the classifier.Apply each stage one-by-one to find a facial region. If on any stage the classifier fails, that region will be discarded from further iterations. Only the facial region will pass all the stages of the classifier.

LBP Cascade Classifier

LBP is a texture descriptor and face is composed of micro texture patterns. So LBP features are extracted to form a feature vector to classify a face from a non-face. Following are the basic steps of LBP Cascade classifier algorithm:

**LBP Labelling:** A label as a string of binary numbers is assigned to each pixel of an image.

Feature Vector: Image is divided into sub-regions and for each sub-region, a histogram of labels is constructed. Then, a feature vector is formed by concatenating the sub-regions histograms into a large histogram.

AdaBoost Learning: Strong classifier is constructed using gentle AdaBoost to remove redundant information from feature vector.

Cascade of Classifier: The cascades of classifiers are formed from the features obtained by the gentle AdaBoost algorithm. Sub-regions of the image is evaluated starting from simpler classifier to strong classifier. If on any stage classifier fails, that region will be discarded from further iterations. Only the facial region will pass all the stages of the classifier.

**CODING**

**#import required libraries**

import numpy as np

import cv2

import matplotlib.pyplot as plt

import time

%matplotlib inline

**#load cascade classifier training file for haarcascade**

haar\_face\_cascade = cv2.CascadeClassifier('data/haarcascade\_frontalface\_alt.xml')

**#load test iamge**

test1 = cv2.imread('data/test1.jpg')

**#convert the test image to gray image as opencv face detector expects gray images**

gray\_img = cv2.cvtColor(test1, cv2.COLOR\_BGR2GRAY)

**#display the gray image using OpenCV**

**# cv2.imshow('Test Imag', gray\_img)**

**# cv2.waitKey(0)**

**# cv2.destroyAllWindows()**

**#or if you have matplotlib installed then**

plt.imshow(gray\_img, cmap='gray')



**#let's detect multiscale (some images may be closer to camera than others) images**

faces = haar\_face\_cascade.detectMultiScale(gray\_img, scaleFactor=1.1, minNeighbors=5);

**#print the number of faces found**

print('Faces found: ', len(faces))

Next, let's loop over the list of faces (rectangles) it returned and draw those rectangles using built in OpenCV rectangle function on our original colored image to see if it detected the right faces.

**#go over list of faces and draw them as rectangles on original colored img**

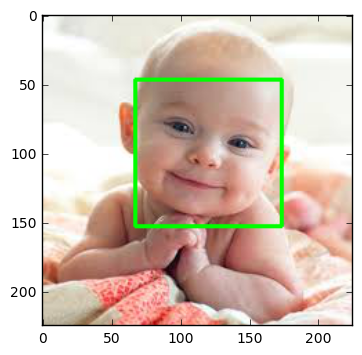
for (x, y, w, h) in faces:

cv2.rectangle(test1, (x, y), (x+w, y+h), (0, 255, 0), 2)

Display the original image to see rectangles drawn and verify that detected faces are really faces and not false positives.

**#conver image to RGB and show image**

plt.imshow(convertToRGB(test1))



def detect\_faces(f\_cascade, colored\_img, scaleFactor = 1.1):

img\_copy = np.copy(colored\_img)

**#convert the test image to gray image as opencv face detector expects gray images**

gray = cv2.cvtColor(img\_copy, cv2.COLOR\_BGR2GRAY)

**#let's detect multiscale (some images may be closer to camera than others) images**

faces = f\_cascade.detectMultiScale(gray, scaleFactor=scaleFactor, minNeighbors=5);

**#go over list of faces and draw them as rectangles on original colored img**

for (x, y, w, h) in faces:

cv2.rectangle(img\_copy, (x, y), (x+w, y+h), (0, 255, 0), 2)

return img\_copy

Now let's try this function on another test image.

**#load another image**

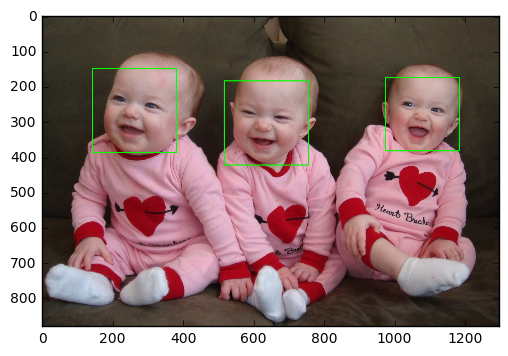
test2 = cv2.imread('data/test3.jpg')

**#call our function to detect faces**

faces\_detected\_img = detect\_faces(haar\_face\_cascade, test2)

**#conver image to RGB and show image**

plt.imshow(convertToRGB(faces\_detected\_img))



detectMultiScale Parameter Details

This function detects the faces in a given test image and following are details of its options.

detectMultiScale: A general function that detects objects. Since we are calling it on the face cascade, that’s what it detects. The first option is the grayscale image.

scaleFactor: Since some faces may be closer to the camera, they would appear bigger than those faces in the back. The scale factor compensates for this.

minNeighbors: The detection algorithm uses a moving window to detect objects. This parameter defines how many objects are detected near the current one before it declares the face found.

There are other parameters as well and you can review the full details of this function here. These parameters need to be tuned according to your data.

For example let's try our Haar face detector on another test image.

**#load another image**

test2 = cv2.imread('data/test4.jpg')

**#call our function to detect faces**

faces\_detected\_img = detect\_faces(haar\_face\_cascade, test2)

**#conver image to RGB and show image**

plt.imshow(convertToRGB(faces\_detected\_img))



**Haar and LBP Results Analysis**

We will run both Haar and LBP on test images to see accuracy and time delay of each.

**#load cascade classifier training file for haarcascade**

haar\_face\_cascade = cv2.CascadeClassifier('data/haarcascade\_frontalface\_alt.xml')

**#load cascade classifier training file for lbpcascade**

lbp\_face\_cascade = cv2.CascadeClassifier('data/lbpcascade\_frontalface.xml')

**#load test image1**

test1 = cv2.imread('data/test5.jpg')

**#load test image2**

test2 = cv2.imread('data/test6.jpg')

**Test-1**

**#------------HAAR-----------**

**#note time before detection**

t1 = time.time()

**#call our function to detect faces**

haar\_detected\_img = detect\_faces(haar\_face\_cascade, test1)

**#note time after detection**

t2 = time.time()

**#calculate time difference**

dt1 = t2 - t1

**#print the time differene**

**#------------LBP-----------**

**#note time before detection**

t1 = time.time()

lbp\_detected\_img = detect\_faces(lbp\_face\_cascade, test1)

**#note time after detection**

t2 = time.time()

**#calculate time difference**

dt2 = t2 - t1

**#print the time differene**

**#----------Let's do some fancy drawing-------------**

**#create a figure of 2 plots (one for Haar and one for LBP)**

f, (ax1, ax2) = plt.subplots(1, 2, figsize=(10, 5))

**#show Haar image**

ax1.set\_title('Haar Detection time: ' + str(round(dt1, 3)) + ' secs')

ax1.imshow(convertToRGB(haar\_detected\_img))

**#show LBP image**

ax2.set\_title('LBP Detection time: ' + str(round(dt2, 3)) + ' secs')

ax2.imshow(convertToRGB(lbp\_detected\_img))

**#show images**

**# plt.imshow(faces\_detected\_img)**



**Accuracy: Both Haar and LBP detected faces successfully.**

**Speed: LBP was faster than Haar**

As you can see LBP is significantly faster than Haar and not that much behind in accuracy so depending on the needs of your application you can use any of the above-mentioned face detection algorithms. Try the code and have fun detecting different faces and analyzing the result. And don't forget to thank OpenCV for giving the implementation of the above-mentioned algorithms.

Face detection has rich real-time applications that include facial recognition, emotions detection (smile detection), facial features detection (like eyes), face tracking etc. You can also explore more exciting machine learning and computer vision algorithms available in OpenCV library.

**Conclusion**

My internship in artificial intelligence provided me with a valuable learning experience and exposure to the latest advancements in AI technology. During the internship, I was able to apply my theoretical knowledge to real-world problems and gain practical experience in AI programming, data analysis, and machine learning algorithms.

Through hands-on training, I learned about various AI frameworks and tools, including TensorFlow, Keras, and PyTorch, and how to use them to develop intelligent applications. I also worked on several projects that involved natural language processing, computer vision, and predictive modeling, which allowed me to develop a deeper understanding of how AI is used in various industries.

Overall, this internship has provided me with a strong foundation in artificial intelligence, and I am confident that the skills and knowledge I have gained will be beneficial in my future career as an AI professional. I am grateful for the opportunity to work with experienced AI professionals and learn from their expertise, and I look forward to applying the knowledge and skills I have gained to solve real-world problems and contribute to the advancement of AI technology.

# LIST OF REFERENCE

1. **Kaggle (for data set):** [image-caption-generator | Kaggle](https://www.kaggle.com/code/wikiabhi/image-caption-generator)
2. **OpenCV:** [Home - OpenCV](https://opencv.org/)
3. **PySpark Documentation:** [PySpark Overview — PySpark 3.4.0 documentation](https://spark.apache.org/docs/latest/api/python/index.html) [(apache.org)](https://spark.apache.org/docs/latest/api/python/index.html)
4. **(AIChatbot)**<https://www.google.com/search?q=chat+gpt&rlz=1C1ONGR_enIN1033IN1033&oq=&aqs=chrome.0.35i39i362l8.818434307j0j15&sourceid=chrome&ie=UTF-8>
5. **(Kiran Naik)**<https://www.youtube.com/watch?v=YFNKnUhm_-s&list=PLZoTAELRMXVPGU70ZGsckrMdr0FteeRUi&index=2>
6. **(BOOK)**[2-Aurélien-Géron-Hands-On-Machine-Learning-with-Scikit-Learn-Keras-and-Tensorflow\_-Concepts-Tools-and-Techniques-to-Build-Intelligent-Systems-O’Reilly-Media-2019.pdf](file:///C:\Users\sbsan\Downloads\2-Aur%C3%A9lien-G%C3%A9ron-Hands-On-Machine-Learning-with-Scikit-Learn-Keras-and-Tensorflow_-Concepts-Tools-and-Techniques-to-Build-Intelligent-Systems-O%E2%80%99Reilly-Media-2019.pdf)