



IMAGE IDENTIFICATION BASED MONITORING ALGORITHM

B.E. [ECE- 3], VII – SEMESTER

PROJECT SEMINAR (subject code : 16ECC38)

under the guidance of **smt D. Nagadevi**

160117735158 : Nitesh Aloney

160117735333 : Kesar Nand Kishore

160117735336 : Manakan Karthik

INTRODUCTION

- **Why this Project ..?** Imagine a situation where you just want to record the images of the unknown persons instead of recording their movement, this algorithm will not only increase data processing speed but on following techniques like proper naming convention, data accessing can also be improved.
- In Short, This project (algorithm) is inspired by the idea of having a monitoring algorithm which is based on IMAGE IDENTIFICATION techniques unlike regular Video Monitoring Systems (basically CCTVs).
- This algorithm will be Intelligent enough to not only detect the persons but also recognize and identify them.
- This can be achieved using image processing libraries like open cv, PIL and other python libraries which are mentioned in technology stack slide futher in this presentation.

CONTENTS

- **AIM**
- **METHODOLOGY**
- **TECHNOLOGY STACK**
- **RESULTS**
- **APPLICATIONS**
- **CONCLUSION**

- The sole idea of this project to successfully develop an monitoring algorithm which is based on IMAGE IDENTIFICATION techniques unlike regular Video based Monitoring Systems.
- This algorithm recognizes the persons in the camera frame if they are known to the dataset but captures the frame and saves it if the detected person's features are not recognized by algorithm.
- One of the naming convention of capturing images can be 'DATE/Unknown_at_<TIMESTAMP>.jpg'
- This algorithm should be able to communicate with the databases for the known user information and possibly with the cloud for the captured images when required by the users.

- **This Algorithm will mainly consists of four components :**
 1. Dataset Generator.
 2. Trainer file.
 3. Recognizer.
 4. Driver Code (user can interact with algorithm).
- **DATASET Generator:** This is the part of algorithm where we generate enough pictures of the user to train the algorithm. For example `user = 'Robert Plant'` then when we call dataset generator it starts capturing the images of the user and provide `ID` to the user and stores the user information into the database after generating the training data. These all captured images are stored in the dataset folder which is a common folder for all the training data
- **TRAINER:** This is the part of the algo where we use the images in the dataset folder and train the our algorithm so that it is fit for recognizing the users which are familiar to our database. Every time we call the dataset generator we need to call the trainer so that our algorithm stays updated about the new users in the database.

METHODOLOGY

- **RECOGNIZER:** This is the part of algorithm where we the algorithm is completely trained and intelligent enough to recognize the persons and differentiate them with the strangers. Recognizer labels the recognized faces with their username but on detecting an unknown face captures them continuously and saves the frame with the time stamp into the specified folder and also pushes the record into the unknown_log file.
- **DRIVER CODE:** This is the part of algorithm is the communication media between the users and databases, the three components of the algorithm (Recognizer, Trainer, Generate Dataset). When user wants to add a new user, or if he wants to review the log, remove an existing user, etc.... These all can be done by Driver Code. This Driver code will be running all the time so that the operations of the algorithm never stop.

TECHNOLOGY STACK

TECHNOLOGY	USAGE
Python 3.x Env (PyCharm / Jupyter Lab)	To Program our algorithm. We write our project code using this Python Environment.
ML libs (cv2, PIL)	For Image Processing Operations like converting images from one to other form
Python libs (pandas, numpy, Matplotlib, sql)	For matrix operations, database queries, Data Visualizations, Database Management as well.
Databases (.CSV, SQL)	.CSV files (less number of users), MySQL / MongoDB for storing the user information on large scale.
Haarcascade Classifiers	These are the Open source algorithms which we'll use for the Frontal face detection.


APPLICATIONS

- ❑ This Projection can be extended to a lot of Domains like security, monitoring.
- ❑ This can be very useful in Ticket Counters (in railways etc) as we can capture the people who are booking a ticket for the future references.
- ❑ This can serve as an alternative to CCTVs where Data Processing, Data Accessing are improved but compromising the level of security the cost of installing this algorithm is not even the half of that of installing regular CCTVs.
- ❑ This can be Integrated with the SMTP Protocol when user wants the records in mails.
- ❑ Consider the usage of this project in an examination center where proxies can be handled in a great way.

RESULTS

EXPECTED ALGORITHM FLOW AND TWO BASE CASES

CASE 1:

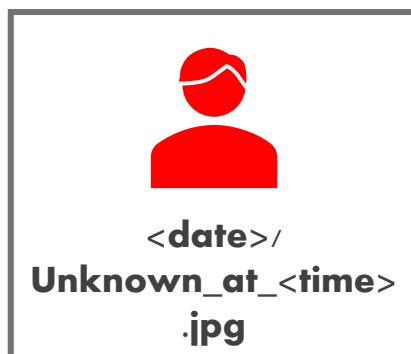
 algorithm
+
Robert Plant



Recognizes



CASE 2:



Tries to match
with the
dataset but fails



algorithm
+
Unknown_User

CONCLUSION

- **This Algorithm is also capable of establishing itself as a market product after embedding into a CCTV's and other similar products.**
- **Thus this algorithm is capable of serving as an alternative to video monitoring systems with better data processing and accessing techniques and it is also cost effective.**
- **This algorithm is also versatile enough to be integrated with other open source projects**

THANK YOU

