ATTENTION SPAN PREDICTION USING HEAD-POSE ESTIMATION WITH DEEP NEURAL NETWORKS

A PROJECT REPORT

Submitted by

K.BUVANESHVARAN 811318104003

A.JHEYARAGHAVAN 811318104006

J.KEVIN ROLAND 811318104007

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

J. J. COLLEGE OF ENGINEERING AND TECHNOLOGY

TIRUCHIRAPPALLI - 620 009

ANNA UNIVERSITY : CHENNAI 600 025

JUNE 2022

ANNA UNIVERSITY : CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this project report **“ATTENTION SPAN PREDICTION USING HEAD-POSE ESTIMATION WITH DEEP NEURAL NETWORKS”** is the bonafide work of **“K.BUVANESHVARAN (811318104003), A.JHEYARAGHAVAN (811318104006) , J.KEVIN ROLAND (811318104007)”** who carried out the project work under my supervision. Certified further that to the best of my knowledge the work reported here in does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE	**SIGNATURE**
Mrs. J.B.PRADEEPA	Drs. M.P.REVATHI
SUPERVISOR	**HEAD OF THE DEPARTMENT**
Dept of Computer Science and Engg.	Dept of Computer Science and Engg.
J. J. College of Engg. and Technology,	J. J. College of Engg. and Technology,
Trichy 620 009.	Trichy 620 009.

Submitted for Semester Project viva-voce examination held on _________

INTERNAL EXAMINER. **EXTERNAL EXAMINER.**

ACKNOWLEDGEMENT

We are extremely thankful to our respected chairman **Dr. S. Ramamoorthiy**, J. J. Group of Institutions our esteemed Vice-Chairman **Er. R. Chenthurselvan** and our beloved Principal **Dr. P. Mathiyalagan**, J. J. College of Engineering and Technology, Trichy for the following us to take up this project.

We express our deep sense of gratitude to **Dr.M.P.Revathi**, Associate Professor and Head of the Department, Computer Science and Engineering, J. J. College of and Technology, Trichy for her technical and moral support.

We are greatly indebted to our coordinator **Dr.P.Chellammal**, Associate Professor, Department of Computer Science and Engineering, J.J. College of Engineering and Technology, Trichy for their valuable help and encouragement during the project work.

We owe our deep sense of gratitude to our Project Coordinator and guide **Mrs.J.B.Pradeepa**, Assistant Professor, Department of Computer Science and Engineering, J. J. College of Engineering and Technology, Trichy for her valuable guidance and encouragement during the project work.

We also express our sincere thanks to all the faculty members, Department of Computer Science and Engineering, J. J. College of Engineering and Technology, Trichy and friends who helped for the successful completion of the project.

Finally, yet importantly, we would like to express our heartful thanks to our beloved parents and all our family members for their blessings.

ABSTRACT

Designing a system for automatic image content recognition is a non-trivial task that has been studied for a variety of applications such as face detection, face recognition. Face recognition is one of numerous presentations of digital image processing. Automatic face detection is a complex problem which is concerned with the automatic identification of an individual. But there are no solutions to detect faces automatically with low resolutions in various applications. We implement the project, computer vision system to predict the screens which are near to their vision or not. Monitors placed too close or too far away may cause problems that may lead to eyestrain. Viewing distances that are too long can cause to lean forward and strain to see small text. This can tiredness the eyes and place stress on the torso because the backrest is no longer provided that support. Viewing distances that are too short may cause eyes to work harder to focus (convergence problems) and may require sitting in awkward postures. For instance, user may tilt their head backward or push chair away from the screen, causing you to type with outstretched arms. But there is no alert system for measuring distance automatically from monitor to eye. So we design the implementation for automatic alert based on distance based on face recognition. The minimum distance from computer to human is 0.38 m(1.2 ft.) and maximum distance from computer to human is 1.02 m(3.3 ft.). It can be achieved by using deep learning. We can use

web camera for capturing human head positions and separate the background from foreground head positions. Then using image processing techniques to detect face and recognize. Finally calculate the distance from monitor to face via web camera. If the distance is minimum to pre-define threshold value means, alert is automatically generated and intimate to users without using any sensors and also extend the approach to design the parent children framework to send alert at the time of seeing unwanted websites.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

AI	Artificial Intelligence
RPA	Robotic Process Automation
NLP	Natural Language Processing
ANN	Artificial Neural Networks
MAEV	Mean Absolute Error Of Vectors
ICP	Iterative Closest Point
HRI	Human-Robot Interaction
DFD	Data Flow Diagram
CNN	Convolutional Neural Network Algorithm
LDA	Linear Discriminative Analysis
RPV	Resting Factor Of Vergence

CHAPTER 1

INTRODUCTION

1.1 ARTIFICIAL INTELLIGENCE

AI (Artificial Intelligence) is the simulation of human intelligence processes by machines, especially computer systems. These processes include learning (the acquisition of information and rules for using the information), reasoning (using rules to reach approximate or definite conclusions) and self-correction. Particular applications of AI include expert systems, speech recognition and machine vision. AI can be categorized in any number of ways, but here are two examples. The first classifies AI systems as either weak AI or strong AI. Weak AI, also known as narrow AI, is an AI system that is designed and trained for a particular task. Virtual personal assistants, such as Apple's Siri, are a form of weak AI. Strong AI, also known as artificial general intelligence, is an AI system with generalized human cognitive abilities so that when presented with an unfamiliar task, it has enough intelligence to find a solution. The Turing Test, developed by mathematician Alan Turing in 1950, is a method used to determine if a computer can actually think like a human, although the method is controversial.

1.2 TECHNOLOGIES IN AI

- **Automation:** What makes a system or process function automatically. For example, robotic process automation (RPA) can be programmed to perform high-volume, repeatable tasks that humans normally performed. RPA is different from IT automation in that it can adapt to changing circumstances.
- **Machine learning:** The science of getting a computer to act without programming. Deep learning is a subset of machine learning that, in very

simple terms, can be thought of as the automation of predictive analytics. There are three types of machine learning algorithms:

- Supervised learning: Data sets are labelled so that patterns can be detected and used to label new data sets.
- Unsupervised learning: Data sets aren't labelled and are sorted according to similarities or differences.
- Reinforcement learning: Data sets aren't labelled but, after performing an action or several actions, the AI system is given feedback.

- **Machine vision:** The science of allowing computers to see. This technology captures and analyzes visual information using a camera, analog-to-digital conversion and digital signal processing. It is often compared to human eyesight, but machine vision isn't bound by biology and can be programmed to see through walls, for example. It is used in a range of applications from signature identification to medical image analysis. Computer vision, which is focused on machine-based image processing, is often conflated with machine vision.
- **Natural language processing (NLP):** The processing of human and not computer language by a computer program. One of the older and best known examples of NLP is spam detection, which looks at the subject line and the text of an email and decides if it's junk. Current approaches to NLP are based on machine learning. NLP tasks include text translation, sentiment analysis and speech recognition.
- **Robotics:** A field of engineering focused on the design and manufacturing of robots. Robots are often used to perform tasks that are difficult for humans to perform or perform consistently. They are used in assembly lines for car

production or by NASA to move large objects in space. Researchers are also using machine learning to build robots that can interact in social settings.

- **Self-driving cars:** These use a combination of computer vision, image recognition and deep learning to build automated skill at piloting a vehicle while staying in a given lane and avoiding unexpected obstructions, such as pedestrians.

1.3 APPLICATIONS OF AI

Artificial intelligence has made its way into a number of areas. Here are six examples.

- **AI in healthcare**. The biggest bets are on improving patient outcomes and reducing costs. Companies are applying machine learning to make better and faster diagnoses than humans. One of the best known healthcare technologies is IBM Watson. It understands natural language and is capable of responding to questions asked of it. The system mines patient data and other available data sources to form a hypothesis, which it then presents with a confidence scoring schema. Other AI applications include chatbots, a computer program used online to answer questions and assist customers, to help schedule follow-up appointments or aid patients through the billing process, and virtual health assistants that provide basic medical feedback.
- **AI in business**. Robotic process automation is being applied to highly repetitive tasks normally performed by humans. Machine learning algorithms are being integrated into analytics and CRM platforms to uncover information on how to better serve customers. Chatbots have been incorporated into

websites to provide immediate service to customers. Automation of job positions has also become a talking point among academics and IT analysts.

- **AI in education**. AI can automate grading, giving educators more time. AI can assess students and adapt to their needs, helping them work at their own pace. AI tutors can provide additional support to students, ensuring they stay on track. AI could change where and how students learn, perhaps even replacing some teachers.
- **AI in finance**. AI in personal finance applications, such as Mint or Turbo Tax, is disrupting financial institutions. Applications such as these collect personal data and provide financial advice. Other programs, such as IBM Watson, have been applied to the process of buying a home. Today, software performs much of the trading on Wall Street.
- **AI in law**. The discovery process, sifting through of documents, in law is often overwhelming for humans. Automating this process is a more efficient use of time. Start-ups are also building question-and-answer computer assistants that can sift programmed-to-answer questions by examining the taxonomy and ontology associated with a database.
- **AI in manufacturing**. This is an area that has been at the forefront of incorporating robots into the workflow. Industrial robots used to perform single tasks and were separated from human workers, but as the technology advanced that changed.

1.4 DISADVANTAGES OF AI

While AI tools present a range of new functionality for businesses, artificial intelligence also raises some ethical questions. Deep learning algorithms, which underpin many of the most advanced AI tools, only know what's in the data used during training. Most available data sets for training likely contain traces of human bias. This in turn can make the AI tools biased in their function. This has been seen in the Microsoft Chabot Tay, which learned a misogynistic and anti-Semitic vocabulary from Twitter users, and the Google Photo image classification tool that classified a group of African Americans as gorillas. The application of AI in the realm of self-driving cars also raises ethical concerns. When an autonomous vehicle is involved in an accident, liability is unclear. Autonomous vehicles may also be put in a position where an accident is unavoidable, forcing it to make ethical decisions about how to minimize damage. Another major concern is the potential for abuse of AI tools. Hackers are starting to use sophisticated machine learning tools to gain access to sensitive systems, complicating the issue of security beyond its current state. Deep learning-based video and audio generation tools also present bad actors with the tools necessary to create so-called deep fakes, convincingly fabricated videos of public figures saying or doing things that never took place.

1.5 AI IN IMAGE PROCESSING

In imaging science, image processing is processing of images using mathematical operations by using any form of signal processing for which the input is an image, a series of images, or a video, such as a photograph or video frame; the output of image processing may be either an image or a set of characteristics or parameters related to the image. Most image-processing techniques involve treating the image as a two-dimensional signal and applying standard signal-processing techniques to it. Images are also processed as three-dimensional signals with the third-dimension being time or the z-axis. Image processing usually refers to digital image processing, but optical and analog image processing also are possible. This article is about general techniques that apply to all of them. The acquisition of images (producing the input image in the first place) is referred to as imaging. Closely related to image processing are computer graphics and computer vision.

1.6 DEEP LEARNING

Deep learning is a type of machine learning and artificial intelligence (AI) that imitates the way humans gain certain types of knowledge. Deep learning is an important element of data science, which includes statistics and predictive modelling.

- Deep learning is an AI function that mimics the workings of the human brain in processing data for use in detecting objects, recognizing speech, translating languages, and making decisions.
- Deep learning AI is able to learn without human supervision, drawing from data that is both unstructured and unlabeled.
- Deep learning, a form of machine learning, can be used to help detect fraud or money laundering, among other functions.

Deep learning has evolved hand-in-hand with the digital era, which has brought about an explosion of data in all forms and from every region of the world. This data, known simply as big data, is drawn from sources like social media, internet search engines, e-commerce platforms, and online cinemas, among others. This enormous amount of data is readily accessible and can be shared through fintech applications like cloud computing. However, the data, which normally is unstructured, is so vast that it could take decades for humans to comprehend it and extract relevant information. Companies realize the incredible potential that can result from unraveling this wealth of information and are increasingly adapting to AI systems for automated support. One of the most common AI techniques used for processing big data is machine learning, a self-adaptive algorithm that gets increasingly better analysis and patterns with experience or with newly added data. If a digital payments company wanted to detect the occurrence or potential for fraud in its system, it could employ machine learning tools for this purpose. The computational algorithm built into a computer model will process all transactions happening on the digital platform, find patterns in the data set, and point out any anomaly detected by the pattern. Deep learning, a subset of machine learning, utilizes a hierarchical level of artificial neural networks to carry out the process of machine learning. The artificial neural networks are built like the human brain, with neuron nodes connected together like a web. While traditional programs build analysis with data in a linear way, the hierarchical function of deep learning systems enables machines to process data with a nonlinear approach. A traditional approach to detecting fraud or money laundering might rely on the amount of transaction that ensues, while a deep learning nonlinear technique would include time, geographic location, IP address, type of retailer, and any other feature that is likely to point to fraudulent activity. The first layer of the neural network processes a raw data input like the amount of the transaction and passes it on to the next layer as output. The second layer processes the previous layer's information by including additional information like the user's IP address and passes on its result. The next layer takes the second layer's information and includes raw data like geographic location and makes the machine's pattern even better. This continues across all levels of the neuron network. Artificial neural networks (ANNs) were inspired by information processing and distributed communication nodes in biological systems. ANNs have various differences from biological brains. Specifically, neural networks tend to be static and symbolic, while the biological brain of most living organisms is dynamic (plastic) and analogue. The adjective "deep" in deep learning refers to the use of multiple layers in the network. Early work showed that a linear perception cannot be a universal classifier, and then that a network with a non polynomial activation function with one hidden layer of unbounded width can on the other hand so be. Deep learning is a modern variation which is concerned with an unbounded number of layers of bounded size, which permits practical application and optimized implementation, while retaining theoretical universality under mild conditions. In deep learning the layers are also permitted to be heterogeneous and to deviate widely from biologically informed connectionist models, for the sake of efficiency, trainability and understandability, whence the "structured" part. Deep Learning is a machine learning technique that constructs artificial neural networks to mimic the structure and function of the human brain. In practice, deep learning, also known as deep structured learning or hierarchical learning, uses a large number hidden layers, typically more than 6 but often much higher of nonlinear processing to extract features from data and transform the data into different levels of abstraction (representations).

As an example, assume the input data is a matrix of pixels. The first layer typically abstracts the pixels and recognizes the edges of features in the image. The next layer might build simple features from the edges such as leaves and branches. The next layer could then recognize a tree and so on. The data passing from one layer to the next is considered a transformation, turning the output of one layer into the input for the next. Each layer corresponds with a different level of abstraction and the machine can learn which features of the data to place in which layer/level on its own. Deep learning is differentiated from traditional “shallow learning” because it learns much deeper levels of hierarchical abstraction and representations. This learning technique is a groundbreaking tool for processing large quantities of data, since the performance of the machine improves as it analyzes more data. As the amount of data increases, the machine becomes more adept at recognizing even hidden patterns among the data. Because the machine is also learning from the processed data, it is able to perform feature extraction and abstraction automatically from the raw data with little to no human input.

1.7 DEEP LEARNING IN IMAGE PROCESSING

In imaging science, image processing is processing of images using mathematical operations by using any form of signal processing for which the input is an image, a series of images, or a video, such as a photograph or video frame; the output of image processing may be either an image or a set of characteristics or parameters related to the image. Most image-processing techniques involve treating the image as a two-dimensional signal and applying standard signal-processing techniques to it. Images are also processed as three-dimensional signals with the third-dimension being time or the z-axis. Image processing usually refers to digital image processing, but optical and analog image processing also are possible. This article is about general techniques that apply to all of them. The acquisition of images (producing the input image in the first place) is referred to as imaging. Closely related to image processing are computer graphics and computer vision. In computer graphics, images are manually made from physical models of objects, environments, and lighting, instead of being acquired (via imaging devices such as cameras) from natural scenes, as in most animated movies. Computer vision, on the other hand, is often considered high-level image processing out of which a machine/computer/software intends to decipher the physical contents of an image or a sequence of images (e.g., videos or 3D full-body magnetic resonance scans). In modern sciences and technologies, images also gain much broader scopes due to the ever growing importance of scientific visualization (of often large-scale complex scientific/experimental data). Examples include microarray data in genetic research, or real-time multi-asset portfolio trading in finance. Image analysis is the extraction of meaningful information from images; mainly from digital images by means of digital image processing techniques. Image analysis tasks can be as simple as reading bar coded tags or as sophisticated as identifying a person from their face.

Computers are indispensable for the analysis of large amounts of data, for tasks that require complex computation, or for the extraction of quantitative information. On the other hand, the human visual cortex is an excellent image analysis apparatus, especially for extracting higher-level information, and for many applications including medicine, security, and remote sensing human analysts still cannot be replaced by computers. For this reason, many important image analysis tools such as edge detectors and neural networks are inspired by human visual perception models. Image editing encompasses the processes of altering images, whether they are digital photographs, traditional photochemical photographs, or illustrations. Traditional analog image editing is known as photo retouching, using tools such as an airbrush to modify photographs, or editing illustrations with any traditional art medium. Graphic software programs, which can be broadly grouped into vector graphics editors, raster graphics editors, and 3D modelers, are the primary tools with which a user may manipulate, enhance, and transform images.

CHAPTER 2

LITERATURE SURVEY

2.1 FINE-GRAINED HEAD POSE ESTIMATION WITHOUT KEY POINTS

AUTHOR : NATANIEL RUIZ EUNJI CHONG

ABSTRACT

The author Nataniel Ruiz Eunji Chong (2018) Proposed new Technique for “Fine-Grained Head Pose Estimation Without Key Points”. In this work we show that a multi-loss deep network can directly, accurately and robustly predict head rotation from image intensities. We show that such a network outperforms landmark-to-pose methods using state-of-the-art landmark detection methods. Landmark-to-pose methods are studied in this work to show their dependence on extraneous factors such as head model and landmark detection accuracy. We also show that our proposed method generalizes across datasets and that it outperforms networks that regress head pose as a sub-goal in detecting landmarks. We show that landmark-to-pose is fragile in cases of very low resolution and that, if the training data is a`ppropriately augmented; our method shows robustness to these situations. Synthetic data generation for extreme poses seems to be a way to improve performance for the proposed method as are studies into more intricate network architectures that might take into account full body pose for example.

2.1.1 TECHNOLOGY

- Multi-loss Convolutional neural network algorithm.

ADVANTAGES

- Accurately and robustly predict head rotation from image intensities.

DISADVANTAGES

- Error rate is high.

2.2 ROBUST MODEL-BASED 3D HEAD POSE ESTIMATION

AUTHOR: GREGORY P. MEYER,SHALINI GUPTA, IURI FROSIO

ABSTRACT

The author Gregory P. Meyer et.al (2016) Proposed new technique for "Robust Model-Based 3d Head Pose Estimation" It requires no initialization, handles extreme rotations and partial occlusions, and efficiently registers facial surfaces. Numerous factors contribute to the success of our algorithm: the overlap term (Ec) in the cost function, the combined PSO and ICP algorithm, dynamically adapting the weights of the face model, and the adoption of a morph able face model. While these concepts have each been introduced individually in previous studies, the contribution of our work lies in combining these disparate ideas in an effective manner to significantly improve the accuracy of 3D head pose estimation. Our work also presents for the first time a systematic quantitative assessment of the contribution of each of these various factors in improving the accuracy of head pose estimation. Building upon the work of Quintal. we also provide deeper insights into the working of the combined PSO and ICP optimization for 3D surface registration. To our knowledge, ours is also the first work to provide a comprehensive review and in-depth comparison of the existing state-of-the-art techniques for 3D head pose estimation on a common benchmark dataset.

2.2.1 TECHNOLOGY

- The iterative closest point (ICP).

ADVANTAGES

- Best-in-class accuracy on benchmark datasets.

DISADVANTAGES

- Computational complexity is high.

2.3 A VECTOR-BASED REPRESENTATION TO ENHANCE HEAD POSE ESTIMATION

AUTHOR: ZHIWEN CAO, ZONGCHENG CHU, DONGFANG LIU

ABSTRACT

The author Zhiwen Cao et.al (2020) Proposed new technique for “A Vector-Based Representation To Enhance Head Pose Estimation” In this paper, we put forward a new vector-based annotation and a new metric MAEV. They can solve the discontinuity issues caused by Euler angles. By the combination of new vector representation and our TriNet, we achieve state of-the-art performance on the task of head pose estimation. Then show that MAE may not reflect the actual behaviour especially for the cases of profile views. To solve two problems, we propose a new annotation method which uses three vectors to describe head poses and a new measurement Mean Absolute Error of Vectors (MAEV) to assess the performance. We also train a new neural network to predict the three vectors with the constraints of orthogonally. Our proposed method achieves state-of-the-art results on both AFLW2000 and BIWI datasets. Experiments show our vector-based annotation method can effectively reduce prediction errors for large pose angles.

2.3.1 TECHNOLOGY

- Vector Based Annotation method.

ADVANTAGES

- Can achieve better results in head detection.

DISADVANTAGES

- Low level accuracy.

2.4 FSA-NET: LEARNING FINE-GRAINED STRUCTURE AGGREGATION FOR HEAD POSE ESTIMATION FROM A SINGLE IMAGE

AUTHOR: TSUN-YI YANG, YI-TING CHEN1 YEN-YU LIN

ABSTRACT

The author Tsun-Yi Yang et.al (2019) Proposed new technique for “Fsa-Net: Learning Fine-Grained Structure Aggregation For Head Pose Estimation From A Single Image” In this paper, we propose a new way to acquire more meaningful aggregated features with the fine-grained spatial structures. By defining learnable and non-learnable scoring functions of the pixel-level features, we are able to learn complementary model variants. Experiments show that the ensemble of these variants outperforms the state-of-the art methods (both landmark-based and landmark-free ones) while its model size is around 100× smaller than those of previous methods. Furthermore, its estimation on the yaw angle is even more accurate than those methods with multi- modality information such as the RGB-D or RGB-Time recurrent model. We show that it is possible to improve regression

results by learning meaningful intermediate features. Although we only demonstrate on the pose estimation problem, we believe that the idea can be extended to other regression problems as well.

2.4.1 TECHNOLOGY

- RGB-D or RGB-Time recurrent model.

ADVANTAGES

- Detect the multiple faces in images.

DISADVANTAGES

- Need more training data.

2.5 FACE-FROM-DEPTH FOR HEAD POSE ESTIMATION ON DEPTH IMAGES

AUTHOR: GUIDO BORGHI

ABSTRACT

The author Guido Borghi (2018) Proposed new technique for “Face-From-Depth For Head Pose Estimation On Depth Images” In this paper, we propose a robust and fast solution for head and shoulder pose estimation, especially devoted to drivers in cars, but that can be easily generalized to any application where depth images are available. The presented framework provides impressive results, reaching In this paper, we propose a robust and fast solution for head and shoulder pose estimation, especially devoted to drivers in cars, but that can be easily generalized to any application where depth images are available. The presented framework provides impressive results, reaching an accuracy higher than 73% on the new Pandora dataset and a low average error on the Biwi dataset, thus overcoming all state-of-art related works. The complete framework proposed in this paper merges together several modern aspects of computer vision. Among the others, the detection, localization, and pose estimation of the head and the shoulders on depth images have been included. In the following, we describe the state of the art of each mentioned topic, including the Domain Translation research area related to the Face-from-Depth module.

2.5.1 TECHNOLOGY

- Conditional GAN approach.

ADVANTAGES

- Shown real-time and impressive results with two public datasets.

DISADVANTAGES

- Need large number of training datasets.

2.6 HEAD POSE ESTIMATION VIA PROBABILISTIC HIGH-DIMENSIONAL REGRESSION

AUTHOR: VINCENT DROUARD, SILÈYE BA, GEORGIOS EVANGELIDIS, 2015

ABSTRACT

The author Vincent Drouard (2015) Proposed new technique for “Head Pose Estimation Via Probabilistic High-Dimensional Regression” In is paper Head pose is an important visual cue in many scenarios such as social event analysis human-robot interaction (HRI) and driver-assistance systems to name a few. For example, in social event analysis, 3D head-pose information drastically helps to determine the interaction between people and to extract the visual focus of attention .The pose is typically expressed by three angles that describe the egocentric orientation of the head. Its estimation becomes challenging when several people are present in an image, so that their faces have a small support area, typically lower than pixels. Even if the face position in the image is known, one has to extract the pose angles from low resolution data. The detection of local features, e.g., facial landmarks, is problematic in this case and one can only use global visual information.

2.6.1 TECHNOLOGY

- Probabilistic High-Dimensional Regression algorithm.

ADVANTAGES

- Facial landmarks are detected efficiently.

DISADVANTAGES

- Time complexity is high.

2.7 DEEP HEAD POSE ESTIMATION USING SYNTHETIC IMAGES AND PARTIAL ADVERSARIAL DOMAIN ADAPTION FOR CONTINUOUS LABEL SPACES

AUTHOR: FELIX KUHNKE, JORN OSTERMANN

ABSTRACT

The author Felix Kuhnke (2021) Proposed new technique for “Deep Head Pose Estimation Using Synthetic Images And Partial Adversarial Domain

Adaption For Continuous Label Spaces” We proposed a novel unsupervised domain adaptation technique to improve deep head pose estimation performance. We extended recent works on partial domain adaptation to the previously neglected regression tasks where labels are not discrete classes but reside in a continuous label space. Using a balanced resampling of source data and partial adversarial domain adaptation, we lowered the head pose estimation error by nearly 10%. Our approach can be applied to other regression tasks such as hand or body pose estimation to improve results when training on data from another domain (e.g., synthetic data). With our results for partial domain adaption, a promising research direction was established. We will try to extend our work in further studies. In this regard, we are looking forward to others proposing solutions using the novel domain adaptation benchmark1 introduced in this paper.

2.7.1 TECHNOLOGY

- Partial Adversarial Domain Adaptation.

ADVANTAGES

- Detect the face from landmark datasets.

DISADVANTAGES

- Computational steps are high.

CHAPTER 3

PROBLEM DEFINITION

3.1 EXISTING SYSTEM

Nowadays all peoples are used computers, laptops in many fields. These are very useful to do work easily and quickly. But they are many drawbacks using these Personal computers. The laptops which were invented to be itinerant and portable have now replaced the desktops completely. Only operate the laptops limited time only. Because it cause pain in spiral cord, headache and so on. When we use a laptop for prolonged periods, as the distance between the screen of the laptop and the keyboard is very little, the constant staring at the flashing screen can spell sorry-sore to the eye. The reddening of the eye, itching and blurring are some of the common problems related to the eyes. In existing system, provide inbuilt monitors to save our vision problems and also LED technology used with backlighting. In existing several systems designed for the purpose of finding people or faces in images have already been proposed by numerous research groups. The process for detection of faces in this project was based on a two-step approach. First, the image is filtered so that only regions likely to contain human skin are marked. This filter was designed using basic mathematical and image processing functions using sensors. The second stage involves taking the marked skin regions and removing the darkest and brightest regions from the map. The removed regions have been shown through empirical tests to correspond to those regions in faces which are usually the eyes and eyebrows, nostrils, and mouth. The binary skin map and the original image together are used to detect faces in the image. The technique relies on thresh holding the skin regions properly so that holes in face regions will appear at the eyebrows, eyes, mouth, or nose. Theoretically, all other regions of skin will have little or no features and no holes will be created except for at the desired facial features. But there is no advanced sensor to sense whether the person is near to the system or not.

DISADVANTAGES

- There is no sensor for analyzing vision measurements.
- Need additional hardware system.
- Distance can’t be measured properly.
- Only the brightness levels of displays could be measured.

3.2 PROPOSED SYSTEM

In the years since the introduction of the personal computer and the realization that it was the cause of workplace health problems, many guidelines have been published concerning the best viewing angles and distances. The distances allowed are too close and the angles too high. The proven relationship between viewing angle and viewing distance is mostly ignored. Computer work takes place at near distances. In this project we can implement the system, for assessing vision system for measuring distances based on their inbuilt web cameras. We can capture face images and separate foreground from background. Face detection is a computer technology being used in a variety of applications that identifies human faces in digital images. Face detection also refers to the psychological process by which humans locate and attend to faces in a visual scene. Face detection can be regarded as a specific case of object-class detection. HAAR Cascade algorithm focus on the detection of frontal human faces. Any facial feature changes in the database will invalidate the matching process. Firstly, the possible human eye regions are detected by testing all the valley regions in the gray-level image. Then algorithm is used to generate all the possible face regions which include the eyebrows, the iris, the nostril and the mouth corners. Each possible face candidates is normalized to reduce lightning effect caused due to uneven illumination and the shirring effect due to head movement. The fitness value of each candidate is measured based on its projection on the eigen-faces. After a number of iterations, all the face candidates with a high fitness value are selected for further verification. At this stage, the face symmetry is measured and the existence of the different facial features is verified for each face candidate. And draw the bounding box and also calculate distance measurement from web cameras. Then extend the framework to implement parental control to control the children to know about constant seeing and access of unwanted websites.

ADVANTAGES

- Provide self-assessment system for vision applications.
- There is no need to use sensors.
- Provide cost effective systems.
- Implemented in real time scenarios.

CHAPTER 4

SYSTEM DESIGN

4.1 SYSTEM ARCHITECTURE

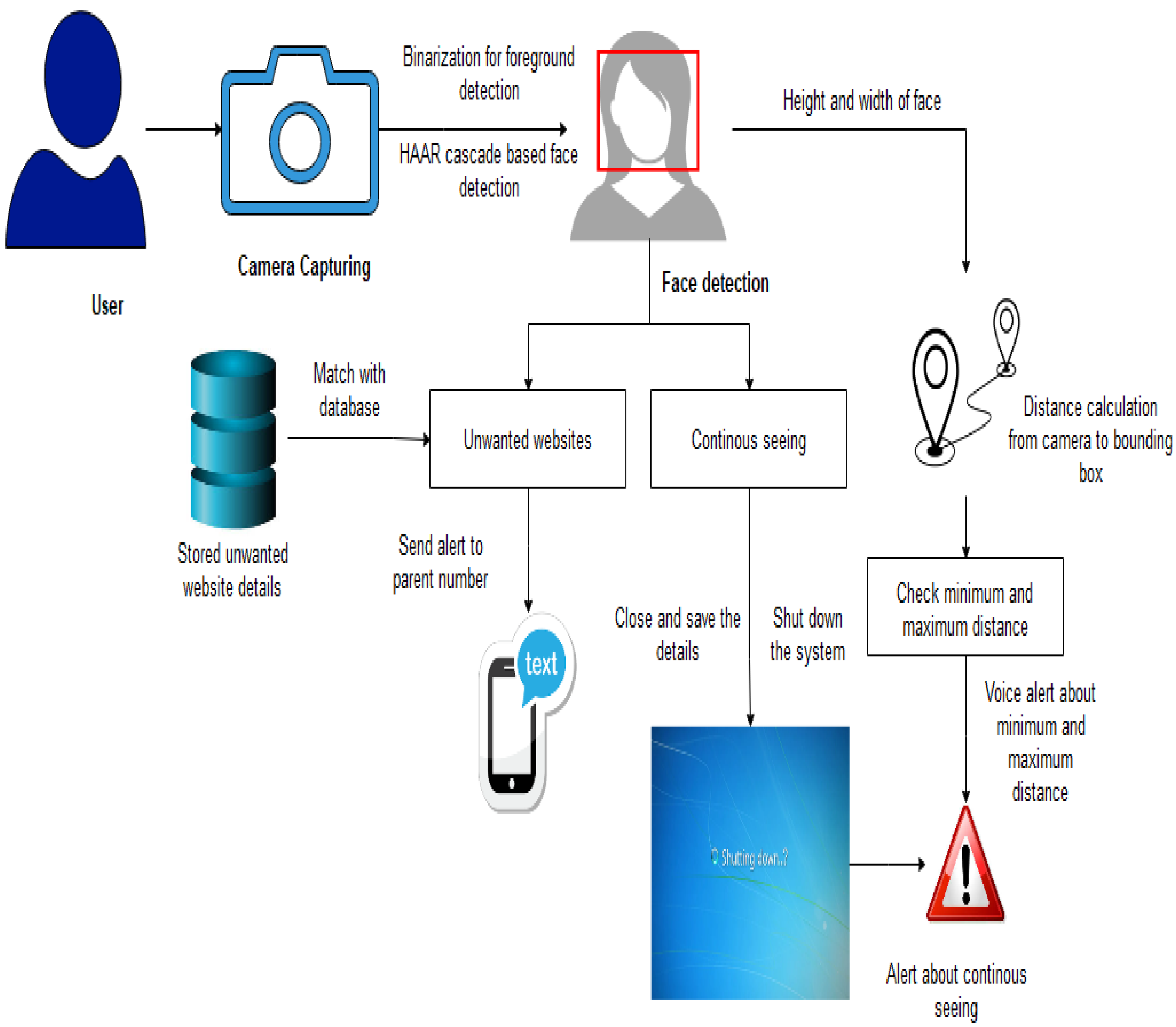
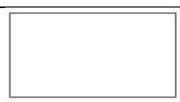
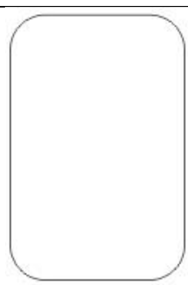


Fig 4.1 System Architecture

4.2 DATA FLOW DIAGRAM

A two-dimensional diagram explains how data is processed and transferred in a system. The graphical depiction identifies each source of data and how it interacts with other data sources to reach a common output. Individuals seeking to draft a data flow diagram must identify external inputs and outputs, determine how the inputs and outputs relate to each other, and explain with graphics how these connections relate and what they result in. This type of diagram helps business development and design teams visualize how data is processed and identify or improve certain aspects.

Data flow Symbols:

Symbol	Description
	An **entity**. A source of data or a destination for data.
	A **process** or task that is performed by the system.
	A **data store**, a place where data is held between processes.
	A **data flow**.

4.3 MODULES DESCRIPTION

- Image acquisition
- Foreground subtraction
- Face detection algorithm
- Face Recognition
- Distance measurement
- Alert system
- Parental Control

4.3.1 IMAGE ACQUISITION:

The first stage of any vision system is the photo acquisition stage. After the photo has been acquired, numerous methods of processing can be carried out to the photo to perform the numerous extraordinary imaginative and prescient tasks required nowadays. However, if the photograph has now not been obtained satisfactorily then the intended obligations won't be possible, in spite of the aid of a few shape of picture enhancement. The fundamental -dimensional picture is a monochrome (greyscale) picture which has been digitized. Describe image as a -dimensional mild depth characteristic f(x,y) where x and y are spatial coordinates and the fee of at any point (x, y) is proportional to the brightness or grey value of the image at that factor. A digitized image is one where

- spatial and gray scale values have been made discrete.
- intensity restrained across a frequently spaced grid in x and y directions.

- intensities sampled to 8 bits (256 values).

In this module, we can capture the face images through web cameras in real time. Capture image can by any type and any size.

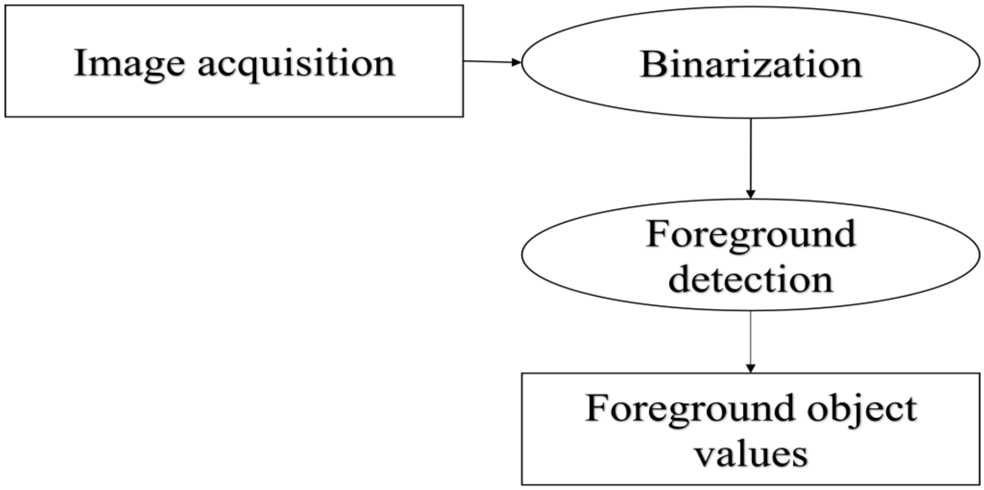


Fig 4.3.1 Level 0 (Image Acquisition)

4.3.2 FOREGROUND SUBTRACTION

Background subtraction, also recognized as foreground recognition, is a technique in the fields of picture processing and pc vision in which an picture's foreground is extracted for in addition processing (item recognition and so forth.). Generally an image's areas of interest are items (humans, cars, text and so forth.) in its foreground. After the stage of photograph preprocessing (which can also include photo denoising, post processing like morphology and many others.) item localization is needed which may also employ this technique. Background subtraction is a widely used technique for detecting transferring objects in motion pictures from static cameras. The reason within the approach is that of detecting the moving items from the difference between the cutting-edge frame and a reference body, frequently called "heritage photograph", or "history model". Background subtraction is usually finished if the photograph in query is part of a video flow. In this module we can implement binarization algorithm to separate the foreground from back ground. File picture binarization is performed within the preprocessing stage for record analysis and it ambitions to section the foreground data from captured image.

4.3.3 FACE DETECTION ALGORITHM:

Automatic face detection is a complicated trouble in photo processing. The assignment given to us turned into to broaden a set of rules able to finding every face in a coloration image of the magnificence. Face detection can be seemed as a specific case of item-elegance detection. In object-magnificence detection, the project is to find the places and sizes of all objects in a photo that belong to a given class. Face-detection algorithms cognizance on the detection of frontal human faces. It is similar to photo detection wherein the image of someone is matched bit by bit. Image fits with the photograph stores in database. Any facial function adjustments in the database will invalidate the matching technique. Firstly, the viable human eye areas are detected by checking out all of the valley regions in the grey-stage photograph. Then the super resolution algorithm is used to generate all the possible face regions which include the eyebrows, the iris, the nostril and the mouth corners. Each viable face applicants is normalized to reduce lightning impact triggered because of uneven illumination and the shirring effect because of head movement. The fitness fee of each candidate is measured primarily based on its projection on the eigen-faces. After some of iterations, all the face applicants with an excessive health cost are decided on for similarly verification. At this stage, the face symmetry is measured and the existence of the one-of-a-kind facial capabilities is tested for each face candidate.

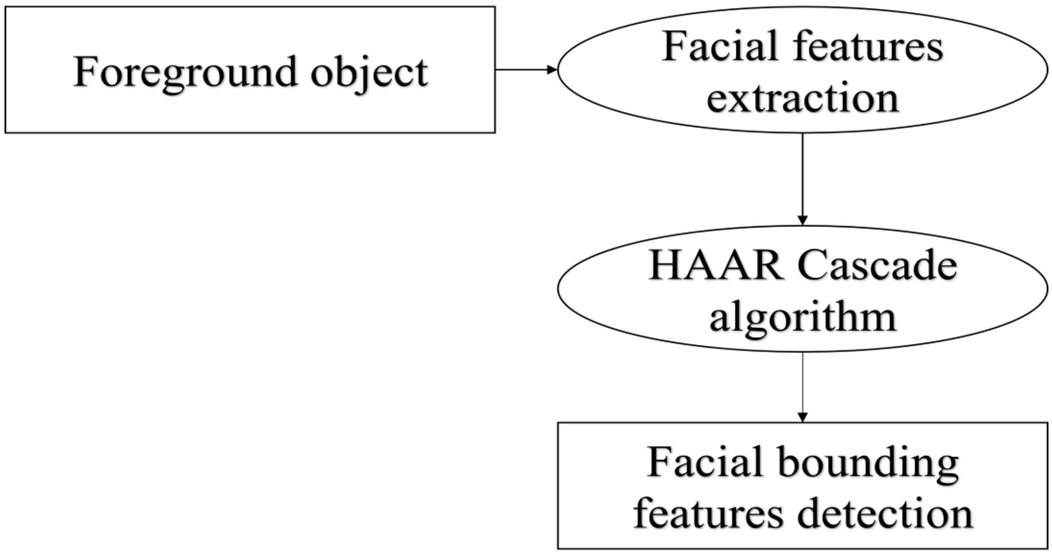


Fig 4.3.3 Level 1 (Face Detection Algorithm)

4.3.4 FACE RECOGNITION:

A user's face is once again acquired and system uses this to either identify who the user is, or verify the claimed identity of the user. While identification includes associating the acquired biometric evidence against templates corresponding to all users in the database, verification involves comparison with only those templates corresponding to claimed identity. The recognition/verification section contains of several modules which might be photograph acquisition, face detection, and face reputation/verification. The input to the face recognition/verification module is the face photo, which is derived from two resources from the digital camera or from the database. Each photograph is reprocessed to get the geometric and photo metric normalized form of the face photograph. During feature extraction, the normalized photo is represented as feature vectors. The end result of the classification for the recognition purpose is determined by using matching the patron index with the purchaser identification inside the database. The reason of the function extraction is to extract the characteristic vectors facts which represent the face.

4.3.5 DISTANCE MEASUREMENT:

In geometry, the minimum or smallest bounding or enclosing box for a face point set (*S*) in *N* dimensions is the box with the smallest within which all the points lie. When other kinds of measure are used, the minutest box is regularly called accordingly, e.g., "minimum-perimeter bounding box". The minimum bounding box of a point set is the same as the minimum bounding box of its convex hull, a fact which may be used heuristically to speed up computation. After bounding box computation, we can calculate viewing distance. The monitor should be at a comfortable horizontal distance for viewing, which usually is around an arm’s length. At this distance you should be able to see the viewing area of the monitor without making head movements. If text looks too small then either use a larger font or magnify the screen image in the software rather than sitting closer to the monitor. Calculate line measurements from web camera through bounding box. Real time implementation can be used for every face images. Bounding box values may be varying for every person.

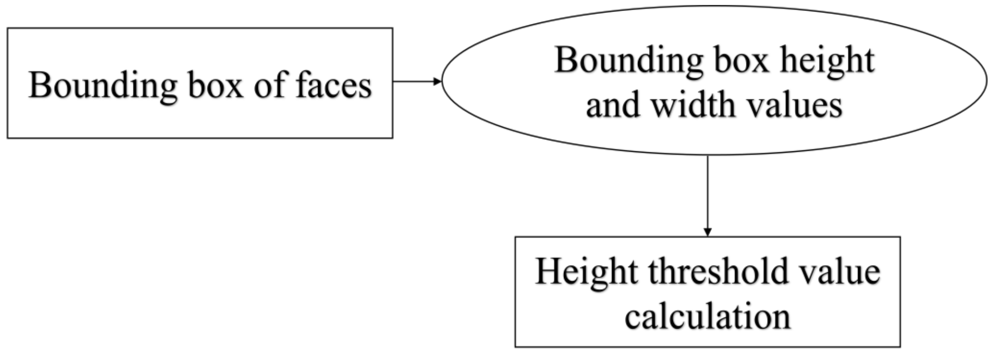


Fig 4.3.5 Level 2 (Distance Measurements)

4.3.6 ALERT SYSTEM:

Alert messaging (or alert notification) is machine-to-person communication that is important or time sensitive. An alert may be a calendar reminder or a notification of a new message. Alert messaging emerged from the study of personal information management (PIM), the science of discovering how people perform certain tasks to acquire, organize, maintain, retrieve and use information relevant to them. In this module implement voice based alert system. We can be set the threshold values for distances. If the distance is less than 1.2 feet means, set voice alert is “Near to the system”. If the distance is greater than 3.3 feet means, set voice alert is “Far from the system”. And also provide voice alert system for constant seeing the system.

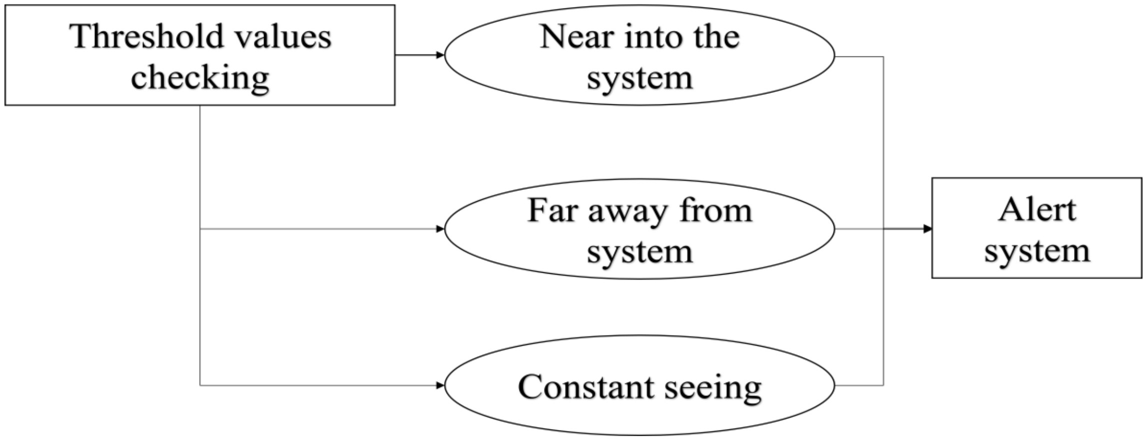


Fig 4.3.6 Level 3 (Alert System)

4.3.7 PARENTAL CONTROL:

In this module, provide parental control about children. The system can monitor the constant seeing based on predefined time settings. After reach the threshold time, alert can be send to stored parents and also mail. Then store the unwanted websites in database for analyzing children can be access the system or not.

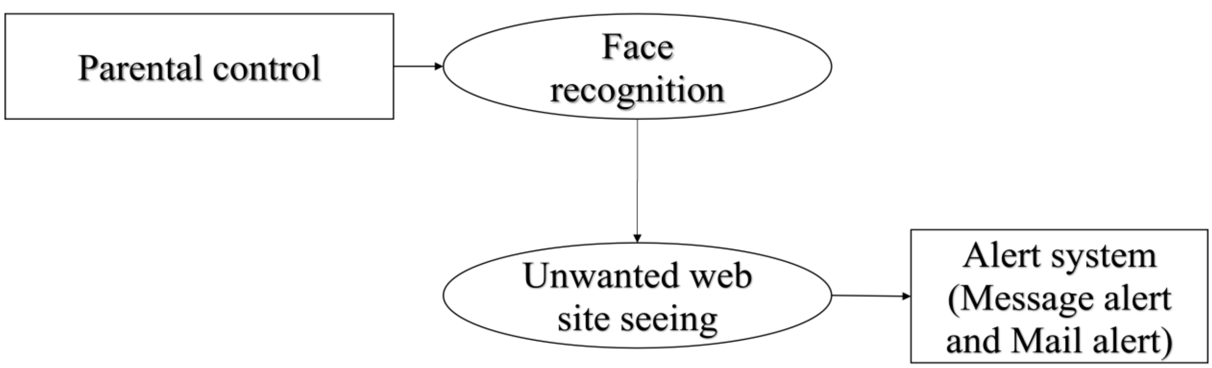


Fig Level 4 (Parental Control)

4.4 ALGORITHM DETAILS

HAAR Cascade algorithm

Haar Cascade classifiers are an effective way for object detection. The algorithm can be explained in four stages:

- Calculating Haar Features.
- Creating Integral Images.
- Using inter and intra class variants.
- Implementing Cascading Classifiers.

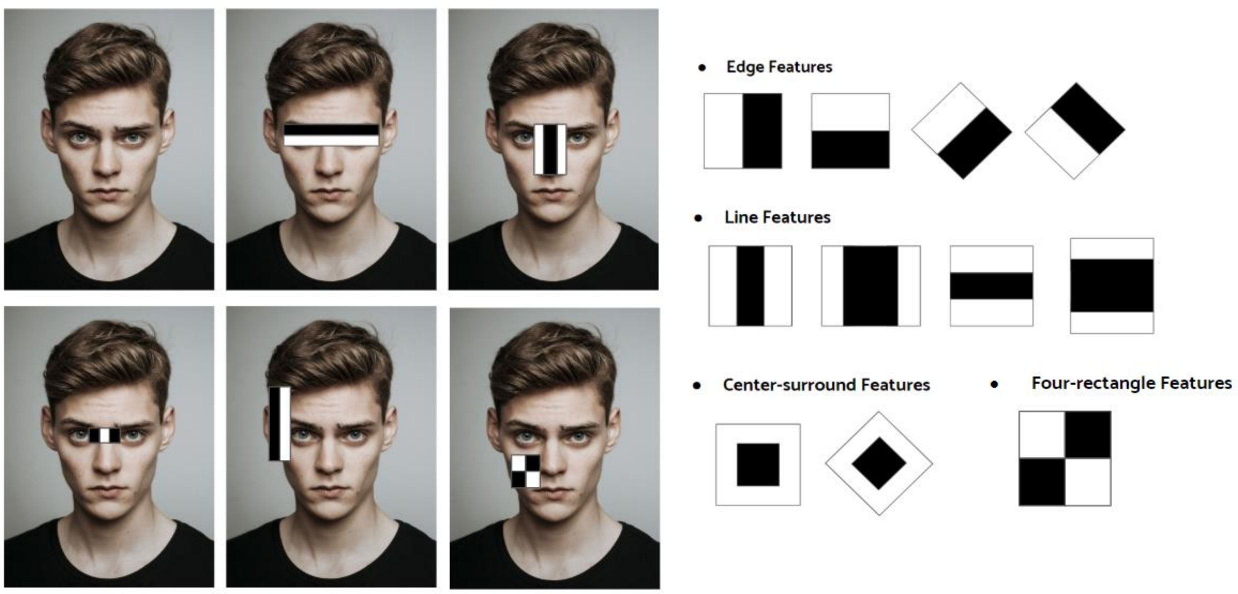


Fig 4.4 HAAR Cascade Algorithm

Linear Discriminative Analysis

LDA is based on two assumptions of linearity. It assumes that the face subspace is linear and there exists a linear separation between classes. Algorithm steps as follows:

Calculate within-class scatter matrix Sw

$$S_w = \sum \sum (x_i^j - \mu_j)(x_i^j - \mu_j)^T$$

where

x_i^j :ith sample of class j C: Number of classes, N_j : Number of samples in class j

Calculate between class scatter matrixSb

$$S_b = \sum (\mu_j - \mu)\,(\mu_j - \mu)^T$$

Where μ represents the mean of all classes. Calculate the eigenvectors of the projection matrix

$$W = eig(S_w^{-1} S_b)$$

Compare projection matrices of the test image and training images and the result is the training image closest to the test image.

Convolutional neural network

Eye states are classified using Convolutional neural network algorithm. CNNs represent feed-forward neural networks which encompass diverse combos of the convolutional layers, max pooling layers, and completely related layers and Take advantage of spatially neighbourhood correlation by way of way of imposing a nearby connectivity pattern among neurons of adjacent layers. Convolutional layers alternate with max pooling layers mimicking the individual of complex and clean cells in mammalian seen cortex. A CNN includes one or extra pairs of convolution and max pooling layers and ultimately ends with completely related neural networks.

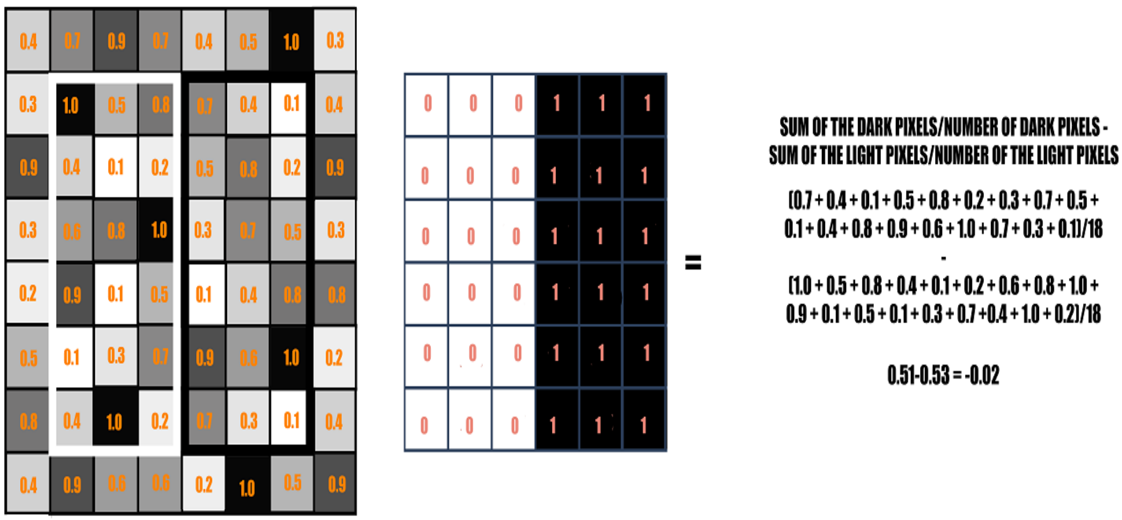


Fig 4.4 Convolutional Neural Network

CHAPTER 5

SYSTEM REQUIREMENTS

5.1 HARDWARE REQUIREMENTS

- Processor : Intel core processor 2.6.0 GHZ
- RAM : 1GB
- Hard disk : 160 GB
- Compact Disk : 650 Mb
- Keyboard : Standard keyboard
- Monitor : 15 inch color monitor

5.2 SOFTWARE REQUIREMENTS

- Operating system : Windows OS
- Front End : .NET (C#)
- Back End : SQL SERVER
- IDE : VISUAL STUDIO

CHAPTER 6

CONCLUSION AND FUTURE ENHANCEMENT

6.1 CONCLUSION

Convergence is when the eyes turn inward towards the nostril while we view close gadgets. Convergence permits the image of the gadgets to be projected to the identical relative vicinity on each retina. Without accurate convergence, we see double photos. The closer the gadgets, the greater the stress on the muscles that converge the eyes. The visual machine also has a resting factor of vergence (RPV). It is similar to the resting point of accommodation, but it's the gap at which the eyes are set to converge while there may be no object to converge on. It's additionally known as darkish vergence. It is difficult to set a specific limit for a minimum viewing distance. If sustained viewing closer than the resting point of vergence contributes to eyestrain, perhaps we must say that eye-display distance should now not be closer than the resting factor of vergence. In this assignment we may be implemented the gadget to using photo processing techniques to detect the faces from digital camera capturing. Then successfully music the faces and to provide bounding boxes on face pictures. Finally set the gap limits to discover whether or not the individual is close to the device or not. And also calculated the person regular seeing conditions and undesirable internet site get entry to. This device can be beneficial to all aged peoples in various packages such as gaming applications, venture works and so on.

6.2 FUTURE ENHANCEMENT

In future we can extend the system to implement various face detection algorithms to improve the accuracy of the system and implement in different scenarios. We can also implemented in various types monitors.

APPENDICES

A.1 SAMPLE SOURCE CODE

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.Data.SqlClient;

namespace EyeiresBased_AttendanceSystem
{
public partial class AdminHome : Form
{
SqlConnection con = new SqlConnection(@"Data
Source=.\SQLEXPRESS;AttachDbFilename=C:\Users\lenovo\Desktop\project\Sel
fvision\Selfvision\selefdb.mdf;Integrated Security=True;User Instance=True");
SqlCommand cmd;
public AdminHome()
    {
        InitializeComponent();
    }
    private void button1_Click(object sender, EventArgs e)
    {
        cmd = new SqlCommand("insert into urltb values('" + comboBox1.Text +
         "')", con);
        con.Open();
        cmd.ExecuteNonQuery();
        con.Close();
        cmd = new SqlCommand("select * from urltb", con);
        SqlDataAdapter da = new SqlDataAdapter(cmd);
        DataTable dt = new DataTable();
        da.Fill(dt);
        dataGridView1.DataSource = dt;
        dataGridView1.Refresh();
    }
```

```
        private void comboBox1_Enter(object sender, EventArgs e)
        {
            con.Open();
            cmd = new SqlCommand("select * from urltb ", con);
            SqlDataReader  dr = cmd.ExecuteReader();
            while (dr.Read())
           {
              comboBox1.Items.Add(dr["url"]);
            }
        con.Close();
        }
        private void button2_Click(object sender, EventArgs e)
        {
          cmd = new SqlCommand("Delete from urltb where url='" +
          comboBox1.Text + "'", con);
            con.Open();
            cmd.ExecuteNonQuery();
            con.Close();
        }
        private void AdminHome_Load(object sender, EventArgs e)
        {
            cmd = new SqlCommand("select * from urltb", con);
            SqlDataAdapter da = new SqlDataAdapter(cmd);
            DataTable dt = new DataTable();
            da.Fill(dt);
            dataGridView1.DataSource = dt;
            dataGridView1.Refresh();
        }
    }
}

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;

namespace EyeiresBased_AttendanceSystem
```

```
{
    public partial class Home : Form
    {
        public Home()
        {
            InitializeComponent();
        }
        private void button2_Click(object sender, EventArgs e)
        {
            Login l = new Login();
            l.Show();
            this.Close();
        }
        private void button1_Click(object sender, EventArgs e)
        {
            NewUser nn = new NewUser();
            nn.Show();
            this.Close();
        }
        private void button3_Click(object sender, EventArgs e)
        {
            AdminHome ah = new AdminHome();
            ah.Show();
        }
    }
}
```

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using Emgu.CV.UI;
using Emgu.CV;
using Emgu.CV.Structure;
using Emgu.CV.CvEnum;
using System.IO;
```

```
using System.Diagnostics;
using AForge;
using AForge.Imaging;
using points = System.Drawing.Point;
using AForge.Imaging.Filters;
using System.Data.SqlClient;

namespace EyeiresBased_AttendanceSystem
{
   public partial class InputName : Form
   {
SqlConnectioncon=newSqlConnection(@"DataSource=.\SQLEXPRESS;AttachDb
Filename=C:\Users\lenovo\Desktop\project\Selfvision\Selfvision\selefdb.mdf;Inte
grated Security=True;User Instance=True");
      SqlCommand cmd;
      public string userName;
      public InputName()
      {
         InitializeComponent();
      }
      private void button1_Click(object sender, EventArgs e)
      {
         this.Close();
      }
      private void textBox1_TextChanged(object sender, EventArgs e)
      {
      }
      private void InputName_Load(object sender, EventArgs e)
      {
      }
      private void pictureBox1_Click(object sender, EventArgs e)
      {
      }
      private void label3_Click(object sender, EventArgs e)
      {
      }
      private void textBox1_TextChanged_1(object sender, EventArgs e)
      {
         userName = textBox1.Text;
      }
```

```
    }
}

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.Data;
using System.Data.SqlClient;
using Emgu.CV.UI;
using Emgu.CV;
using Emgu.CV.Structure;
using Emgu.CV.CvEnum;
using System.IO;
using System.Diagnostics;
using AForge;
using AForge.Imaging;
using points = System.Drawing.Point;
using AForge.Imaging.Filters;
using System.Data.SqlClient;

namespace EyeiresBased_AttendanceSystem
{
    public partial class InputName1 : Form
    {
        SqlConnectioncon=newSqlConnection(@"DataSource=.\SQLEXPRESS;Att
        achDbFilename=C:\Users\lenovo\Desktop\project\Selfvision\Selfvision\sele
        fdb.mdf;Integrated Security=True;User Instance=True");
        SqlCommand cmd;
        public string userName;
        public Bitmap bmp;
        string path;
        Random rnd;
        int count = 0;
        Image<Bgr, Byte> currentFrame, faceframe, eyeframe;
```

```
public HaarCascade Face = new HaarCascade(Application.StartupPath +
"/Cascades/haarcascade_frontalface_default.xml");
public HaarCascade eye = new HaarCascade(Application.StartupPath +
"/Cascades/haarcascade_eye.xml");
Capture grabber;
Image<Gray, byte> result = null;
Image<Gray, byte> gray_frame = null;
MCvFont font = new MCvFont(FONT.CV_FONT_HERSHEY_COMPLEX,
0.5, 0.5);
public InputName1()
{
   InitializeComponent();
}
private void button1_Click(object sender, EventArgs e)
{
}
private void textBox1_TextChanged(object sender, EventArgs e)
{
}
private void textBox1_TextChanged_1(object sender, EventArgs e)
{
}
private void button1_Click_1(object sender, EventArgs e)
{
   string gender;
   if (radioButton1.Checked == true)
   {
      gender = radioButton1.Text;
   }
   else
   {
      gender = radioButton2.Text;
   }
   cmd = new SqlCommand("insert into regtb values('" + label9.Text + "','"
  +gender + "','" + textBox2.Text + "','" + textBox3.Text + "','" +
  textBox4.Text + "')", con);
   con.Open();
   cmd.ExecuteNonQuery();
   con.Close();
   MessageBox.Show("Record Saved!");
```

```
        Home h = new Home();
        h.Show();
        this.Close();
      }
      private void button3_Click(object sender, EventArgs e)
      {
         grabber = new Capture();
         Application.Idle += new EventHandler(FrameGrabber_Standard);
      }
      void FrameGrabber_Standard(object sender, EventArgs e)
      {
         string kkk = "";
         try
         {
            currentFrame = grabber.QueryFrame();
            pictureBox1.Image = currentFrame.ToBitmap();
            gray_frame = currentFrame.Convert<Gray, Byte>();
MCvAvgComp[][]facesDetected=gray_frame.DetectHaarCascade(Face,1.2,10,Em
gu.CV.CvEnum.HAAR_DETECTION_TYPE.DO_CANNY_PRUNING, new
Size(50, 50));
            foreach (MCvAvgComp face_found in facesDetected[0])
               {
               faceframe=currentFrame.Copy(face_found.rect).Resize(100,100,Emg
               u.CV.CvEnum.INTER.CV_INTER_CUBIC);
               currentFrame.Draw(face_found.rect, new Bgr(Color.Violet), 2);
               pictureBox1.Image = faceframe.ToBitmap();
            }
MCvAvgComp[][]eyeDetected=gray_frame.DetectHaarCascade(eye,1.2,10,Emgu.
CV.CvEnum.HAAR_DETECTION_TYPE.DO_CANNY_PRUNING,newSize(50,
50));

            foreach (MCvAvgComp eye_found in eyeDetected[0])
            {
eyeframe=currentFrame.Copy(eye_found.rect).Resize(100,100,Emgu.CV.CvEnum
.INTER.CV_INTER_CUBIC);
               currentFrame.Draw(eye_found.rect, new Bgr(Color.Violet), 2);
               pictureBox2.Image = eyeframe.ToBitmap();
            }
         }
         catch (Exception ex)
```

```
            {
            }
        }
        private void button2_Click(object sender, EventArgs e)
        {
            textBox2.Text = "";
            textBox3.Text = "";
            textBox4.Text = "";
        }
        private void button4_Click(object sender, EventArgs e)
        {
            grabber.Dispose();
            grabber.Stop();
            pictureBox2.SizeMode = PictureBoxSizeMode.StretchImage;
        }
        List<points> points1 = new List<points>();
        static int[] BallX, BallY;
        private void button5_Click(object sender, EventArgs e)
        {
            Bitmap image = new Bitmap(pictureBox2.Image);
            Bitmap grayImage;
            Bitmap bmp = new Bitmap(pictureBox2.Image);
            Grayscale filter = new Grayscale(0.2125, 0.7154, 0.0721);
            grayImage = filter.Apply(bmp);
            ResizeNearestNeighbor rb = new
    ResizeNearestNeighbor(pictureBox1.Width, pictureBox1.Height);
            image = rb.Apply(image);
            MessageBox.Show(image.Height.ToString());
            MessageBox.Show(image.Width.ToString());
            Graphics graphics = Graphics.FromImage(image);
            SolidBrush brush = new SolidBrush(Color.LightYellow);
            Pen pen = new Pen(brush);
            MoravecCornersDetector mcd = new MoravecCornersDetector();
            List<IntPoint> corners = mcd.ProcessImage(image);
            BallX = new int[Convert.ToInt32(corners.Count)];
            BallY = new int[Convert.ToInt32(corners.Count)];
            int vall = 0;
            foreach (IntPoint corner in corners)
            {
                graphics.DrawRectangle(pen, corner.X - 1, corner.Y - 1, 1, 1);
```

```
                BallX[vall] = corner.X;
                BallY[vall] = corner.Y;
                vall += 1;
            }
        }
        private void InputName1_Load(object sender, EventArgs e)
        {
            label9.Text = userName;
        }
    }
}

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.Runtime.InteropServices;
using Luxand;
using System.Data.SqlClient;
using System.IO;
using Emgu.CV.UI;
using Emgu.CV;
using Emgu.CV.Structure;
using Emgu.CV.CvEnum;
using System.Diagnostics;
using AForge;
using AForge.Imaging;
using points = System.Drawing.Point;
using AForge.Imaging.Filters;
using Image = System.Drawing.Image;
using System.Web;
using System.Net;
using System.Speech.Synthesis;
using System.Management;
using System.Net.Mail;
using System.Drawing.Imaging;
```

```
namespace EyeiresBased_AttendanceSystem
{
    public partial class Login : Form
    {
        SqlConnection con = new SqlConnection(@"Data Source=.\SQLEXPRESS;AttachDbFilename=C:\Users\lenovo\Desktop\project\Selfvision\Selfvision\selefdb.mdf;Integrated Security=True;User Instance=True");
        SqlCommand cmd;
        public string usname;
        public Login()
        {
            InitializeComponent();
        }
        Image<Bgr, Byte> currentFrame, faceframe;
        public HaarCascade Face = new HaarCascade(Application.StartupPath + "/Cascades/haarcascade_frontalface_default.xml");
        Capture grabber;
        Image<Gray, byte> result = null;
        Image<Gray, byte> gray_frame = null;
        MCvFont font = new MCvFont(FONT.CV_FONT_HERSHEY_COMPLEX, 0.5, 0.5);
        SpeechSynthesizer speaker = new SpeechSynthesizer();
        string mobile;
        private void Login_Load(object sender, EventArgs e)
        {
        }
        private void button1_Click(object sender, EventArgs e)
        {
            grabber = new Capture();
            Application.Idle += new EventHandler(FrameGrabber_Standard);
        }
        int to;
        int ld, cl;
        void FrameGrabber_Standard(object sender, EventArgs e)
        {
            to = 0;
            ld = 0;
            cl = 0;
            try
            {
```

```
            currentFrame = grabber.QueryFrame();
            pictureBox1.Image = currentFrame.ToBitmap();
            gray_frame = currentFrame.Convert<Gray, Byte>();
            MCvAvgComp[][] facesDetected = gray_frame.DetectHaarCascade(Face, 1.2, 10, Emgu.CV.CvEnum.HAAR_DETECTION_TYPE.DO_CANNY_PRUNING, new Size(50, 50));
            foreach (MCvAvgComp face_found in facesDetected[0])
            {
               to += 1;
               result = currentFrame.Copy(face_found.rect).Convert<Gray, byte>().Resize(100, 100, Emgu.CV.CvEnum.INTER.CV_INTER_CUBIC);
               int ii = Convert.ToInt32(currentFrame.Copy(face_found.rect).Width);
               int i1 = Convert.ToInt32(currentFrame.Copy(face_found.rect).Height);
               int yyy = ii + i1;
               int distance = (10 - yyy / 100) * 6;
               listBox1.Items.Add("Distance:" + (10 - yyy / 100) * 6 + ".cm");
               label5.Text = ("Distance:" + (10 - yyy / 100) * 6 + ".cm");
               faceframe = currentFrame.Copy(face_found.rect).Resize(100, 100, Emgu.CV.CvEnum.INTER.CV_INTER_CUBIC);
               currentFrame.Draw(face_found.rect, new Bgr(Color.Violet), 2);
               pictureBox1.Image = currentFrame.ToBitmap();
               if (distance < 30)
               {
                  s = "you are close to moniter !";
                  label3.Text = "you are close to moniter !";
                  label4.Text = Convert.ToString(Convert.ToInt32(label4.Text) + 1);
                  if (Convert.ToInt32(label4.Text) == 30)
                  {
                     label4.Text = "0";
                     speaker.SpeakAsync("you are closed to the moniter!");
                  }
                  else
                  {
                     label7.Text = "0";
                  }
               }
               else if (distance >= 40 && distance <= 60)
               {
```

```
            label3.Text = "you are in Long distnce from moniter!";
           label7.Text = Convert.ToString(Convert.ToInt32(label7.Text) + 1);
            if (Convert.ToInt32(label7.Text) == 30)
            {
                label7.Text = "0";
               speaker.SpeakAsync("you are in Long distnce from moniter!");
            }
            else
            {
                label4.Text = "0";
            }
        }
        else
        {
            label3.Text = "Normal Distance";
            label4.Text = "0";
            label7.Text = "0";
        }
        if (distance == 0)
        {
            listBox1.Items.Clear();
        }
        if (listBox1.Items.Count >= 100)
        {
            MessageBox.Show("you are in Long time Use");
            speaker.SpeakAsync("you are in Long time Use");
            listBox1.Items.Clear();
        }
    }
    if (to == 0)
    {
        listBox1.Items.Clear();
    }
}
catch
{
}
}
private void pictureBox1_MouseLeave(object sender, EventArgs e)
{
```

```
        }

        private void pictureBox1_MouseMove(object sender, 
System.Windows.Forms.MouseEventArgs e)
        {

        }
        private void pictureBox1_MouseUp(object sender, 
System.Windows.Forms.MouseEventArgs e)
        {
        }
        private void pictureBox1_MouseHover(object sender, EventArgs e)
        {
        }
        private void pictureBox1_MouseLeave_1(object sender, EventArgs e)
        {
        }
        private void pictureBox1_MouseMove_1(object sender, MouseEventArgs e)
        {
        }
        private void pictureBox1_MouseUp_1(object sender, MouseEventArgs e)
        {
        }
        string email;
        private void Login_Load_1(object sender, EventArgs e)
        {
            label6.Text = usname;
            cmd = new SqlCommand("select * from regtb  where name='" + usname + 
"' ", con);
            con.Open();
            SqlDataReader dr = cmd.ExecuteReader();
            if (dr.Read())
            {
                mobile = dr["Mobileno"].ToString();
                email = dr["Emailid"].ToString();
            }
            con.Close();
            cmd = new SqlCommand("select * from Urltb ", con);
            con.Open();
```

```
            SqlDataReader dr1 = cmd.ExecuteReader();
            while (dr1.Read())
            {
                comboBox1.Items.Add(dr1["Url"].ToString());
            }
            con.Close();
        }
        public void sendmessage(string targetno, string message)
        {
            String query = "http://smsserver9.creativepoint.in/api.php?username=fantasy&password=596692&to=" + targetno + "&from=FSSMSS&message=Dear user  your msg is " + message + " Sent By FSMSG FSSMSS&PEID=1501563800000030506&templateid=1507162882948811640";
            WebClient client = new WebClient();
            Stream sin = client.OpenRead(query);
            MessageBox.Show(targetno + "Message Send!");
        }
        private void button2_Click(object sender, EventArgs e)
        {
            grabber.Stop();
            grabber.Dispose();
        }
        private void button3_Click(object sender, EventArgs e)
        {
        }

        private void button4_Click(object sender, EventArgs e)
        {
        }
        private void button3_Click_1(object sender, EventArgs e)
        {
            con.Open();
            cmd = new SqlCommand("select * from Urltb  where  url like '%" + comboBox1.Text + "%' ", con);
            SqlDataReader dr = cmd.ExecuteReader();
            if (dr.Read())
            {
                sendmessage(mobile, "your Son or  Daughter : " + usname + " Use unwanted Website");
```

```
                MessageBox.Show("Unwanted Website");
                System.Diagnostics.Process.Start(comboBox1.Text);
                string to = email;
                string from = "sampletest685@gmail.com";
                string password = "mailtest4";
                using (MailMessage mm = new MailMessage(from, to))
                {
                    mm.Subject = "Alert";
                    mm.Body = "";
                    Image image = pictureBox1.Image;
                    System.IO.MemoryStream stream = new System.IO.MemoryStream();
                    image.Save(stream, ImageFormat.Jpeg);
                    stream.Position = 0;
                    mm.Attachments.Add(new Attachment(stream, "Screenshot.jpg"));
                    mm.IsBodyHtml = false;
                    SmtpClient smtp = new SmtpClient();
                    smtp.Host = "smtp.gmail.com";
                    smtp.EnableSsl = true;
                    NetworkCredential NetworkCred = new NetworkCredential(from,
password);
                    smtp.UseDefaultCredentials = true;
                    smtp.Credentials = NetworkCred;
                    smtp.Port = 587;
                    smtp.Send(mm);
                    MessageBox.Show("Mail Send!");
                }
            }
            else
            {
                MessageBox.Show("Normal Website");
                System.Diagnostics.Process.Start(comboBox1.Text);
            }
            con.Close();
        }
    }
}
```

```
using System;
using System.Collections.Generic;
```

```
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using System.Runtime.InteropServices;
using Luxand;
using System.Data.SqlClient;
using System.IO;
namespace EyeiresBased_AttendanceSystem
{
   public partial class NewUser : Form
   {
      SqlConnection con = new SqlConnection(@"Data Source=.\SQLEXPRESS;AttachDbFilename=C:\Users\lenovo\Desktop\project\Selfvision\Selfvision\selefdb.mdf;Integrated Security=True;User Instance=True");
      SqlCommand cmd;
      enum ProgramState { psRemember, psRecognize }
      ProgramState programState = ProgramState.psRecognize;
      string path = Path.GetDirectoryName(Application.ExecutablePath).ToString();
      String cameraName;
      bool needClose = false;
      string userName;
      String TrackerMemoryFile;
      int mouseX = 0;
      int mouseY = 0;
      [DllImport("gdi32.dll")]
      static extern bool DeleteObject(IntPtr hObject);
      public NewUser()
      {
         InitializeComponent();
      }
      private void NewUser_Load(object sender, EventArgs e)
      {
         TrackerMemoryFile = path + @"\tracker.dat";
         if (FSDK.FSDKE_OK != FSDK.ActivateLibrary("gyYgVWQTSzjiuGB/hH8dKgg0QrrIuhoHdfUCzD9rY+vru3WRZsaezTX6YWj9osdI/cmxY1NSdLkyWuugMPCxUG7/xNLegHLeaUpzVy
```

```
KpDkaWL8tJIUsIL7xv9bhmgifPbAyTDuxF3VGxXmHkv/L/MStf9kdXV/A1vVv
T93QC4vQ="))
            {
              MessageBox.Show("Please run the License Key Wizard (Start - Luxand
- FaceSDK - License Key Wizard)", "Error activating FaceSDK",
MessageBoxButtons.OK, MessageBoxIcon.Error);
              Application.Exit();
            }
            FSDK.InitializeLibrary();
            FSDKCam.InitializeCapturing();
            string[] cameraList;
            int count;
            FSDKCam.GetCameraList(out cameraList, out count);
            if (0 == count)
            {
              MessageBox.Show("Please attach a camera", "Error",
MessageBoxButtons.OK, MessageBoxIcon.Error);
              Application.Exit();
            }
            cameraName = cameraList[0];
            FSDKCam.VideoFormatInfo[] formatList;
            FSDKCam.GetVideoFormatList(ref cameraName, out formatList, out
count);
            int VideoFormat = 0;
            int tracker = 0;
            if (FSDK.FSDKE_OK != FSDK.LoadTrackerMemoryFromFile(ref tracker,
TrackerMemoryFile))
              FSDK.CreateTracker(ref tracker);
        }
        private void button1_Click(object sender, EventArgs e)
        {
            this.button1.Enabled = false;
            int cameraHandle = 0;
            int r = FSDKCam.OpenVideoCamera(ref cameraName, ref cameraHandle);
            if (r != FSDK.FSDKE_OK)
            {
              MessageBox.Show("Error opening the first camera", "Error",
MessageBoxButtons.OK, MessageBoxIcon.Error);
              Application.Exit();
            }
```

```
int tracker = 0;
if (FSDK.FSDKE_OK != FSDK.LoadTrackerMemoryFromFile(ref tracker, TrackerMemoryFile))
    FSDK.CreateTracker(ref tracker);
int err = 0;
FSDK.SetTrackerMultipleParameters(tracker, "HandleArbitraryRotations=false; DetermineFaceRotationAngle=false; InternalResizeWidth=100; FaceDetectionThreshold=5;", ref err);
while (!needClose)
{
    Int32 imageHandle = 0;
    if (FSDK.FSDKE_OK != FSDKCam.GrabFrame(cameraHandle, ref imageHandle))            {
        Application.DoEvents();
        continue;
    }
    FSDK.CImage image = new FSDK.CImage(imageHandle);
    long[] IDs;
    long faceCount = 0;
    FSDK.FeedFrame(tracker, 0, image.ImageHandle, ref faceCount, out IDs, sizeof(long) * 256);
    Array.Resize(ref IDs, (int)faceCount);
    Application.DoEvents();
    Image frameImage = image.ToCLRImage();
    Graphics gr = Graphics.FromImage(frameImage);
    for (int i = 0; i < IDs.Length; ++i)
    {
        FSDK.TFacePosition facePosition = new FSDK.TFacePosition();
        FSDK.GetTrackerFacePosition(tracker, 0, IDs[i], ref facePosition);
        int left = facePosition.xc - (int)(facePosition.w * 0.6);
        int top = facePosition.yc - (int)(facePosition.w * 0.5);
        int w = (int)(facePosition.w * 1.2);
        String name;
        int res = FSDK.GetAllNames(tracker, IDs[i], out name, 65536);
        if (FSDK.FSDKE_OK == res && name.Length > 0)
        {
            StringFormat format = new StringFormat();
            format.Alignment = StringAlignment.Center;
            gr.DrawString("", new System.Drawing.Font("Arial", 16),
```

```
                    new
System.Drawing.SolidBrush(System.Drawing.Color.LightGreen),facePosition.xc,
top + w + 5, format);
                    label1.Text = name;
                }
                else
                {
                    label1.Text = "Unknow User";
                }
                Pen pen = Pens.LightGreen;
                if (mouseX >= left && mouseX <= left + w && mouseY >= top &&
mouseY <= top + w)
                {
                    pen = Pens.Blue;
                    if (ProgramState.psRemember == programState)
                    {
                        if (FSDK.FSDKE_OK == FSDK.LockID(tracker, IDs[i]))
                        {
                            InputName inputName = new InputName();
                            if (DialogResult.OK == inputName.ShowDialog())
                            {
                                userName = inputName.userName;
                                label1.Text = userName;
                                if (userName == null || userName.Length <= 0)
                                {
                                    String s = "";
                                    FSDK.SetName(tracker, IDs[i], "");
                                    FSDK.PurgeID(tracker, IDs[i]);
                                }
                                else
                                {
                                    FSDK.SetName(tracker, IDs[i], userName);
                                }
                                FSDK.UnlockID(tracker, IDs[i]);
                            }
                        }
                    }
                }
                gr.DrawRectangle(pen, left, top, w, w);
            }
```

```
        programState = ProgramState.psRecognize;
        pictureBox1.Image = frameImage;
        GC.Collect();
        FSDK.SaveTrackerMemoryToFile(tracker, TrackerMemoryFile);
    }
    FSDK.SaveTrackerMemoryToFile(tracker, TrackerMemoryFile);
    FSDK.FreeTracker(tracker);
    FSDKCam.CloseVideoCamera(cameraHandle);
    FSDKCam.FinalizeCapturing();
}
private void pictureBox1_MouseUp(object sender, MouseEventArgs e)
{
    programState = ProgramState.psRemember;
}
private void pictureBox1_MouseMove(object sender, MouseEventArgs e)
{
    mouseX = e.X;
    mouseY = e.Y;
}
private void pictureBox1_MouseLeave(object sender, EventArgs e)
{
    mouseX = 0;
    mouseY = 0;
}
private void button2_Click(object sender, EventArgs e)
{
    needClose = true;
}
private void button3_Click(object sender, EventArgs e)
{
    if (label1.Text == "" || label1.Text == "Unknow User")
    {
        MessageBox.Show("Save your Name First!");
    }
    else
    {
        InputName1 iii = new InputName1();
        iii.userName = label1.Text;
        iii.Show();
        this.Close();
```

```
        }
    }
    private void button4_Click(object sender, EventArgs e)
    {
        if (label1.Text == "" || label1.Text == "Unknow User")
        {
            MessageBox.Show("Save your Name First!");
        }
        else
        {
            Login ll = new Login();
            ll.usname = label1.Text;
            ll.Show();
            this.Close();
        }
    }
}
}
```

A.2 SAMPLE SCREENSHOTS

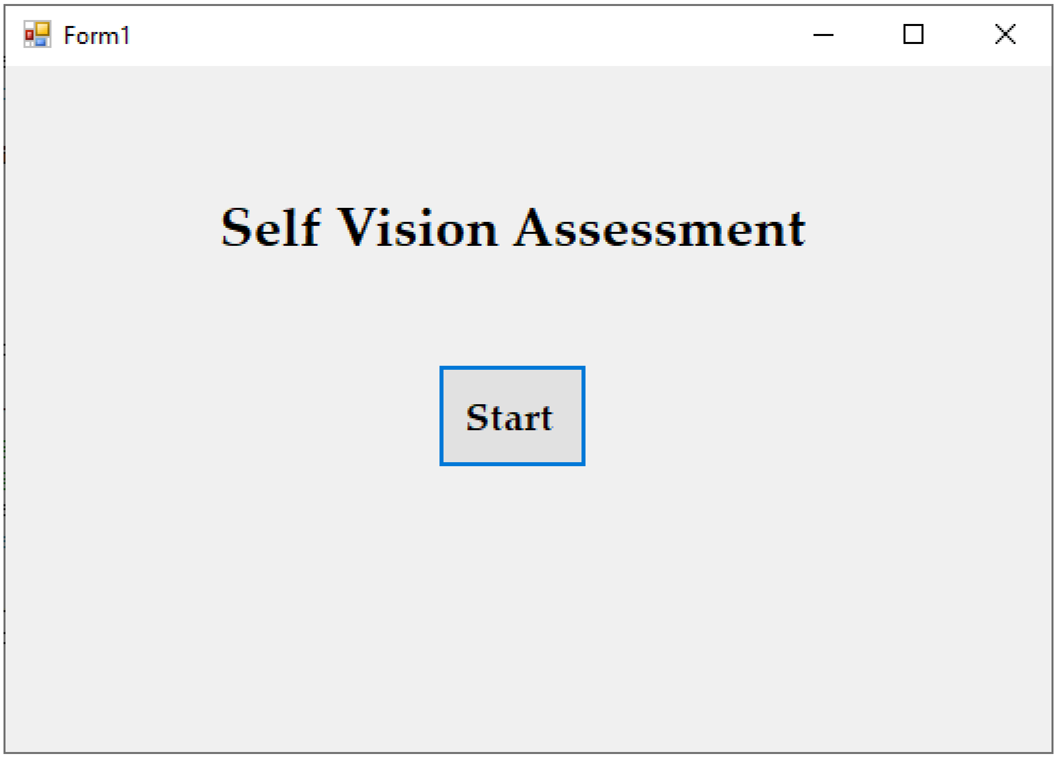


Fig A.2.1 Home

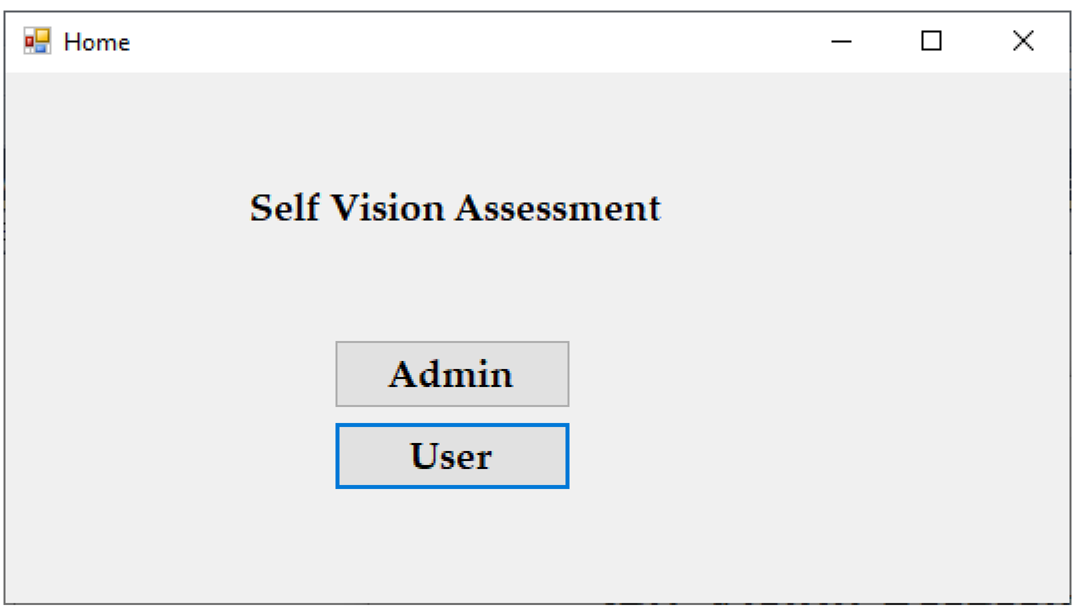


Fig A.2.2 Login

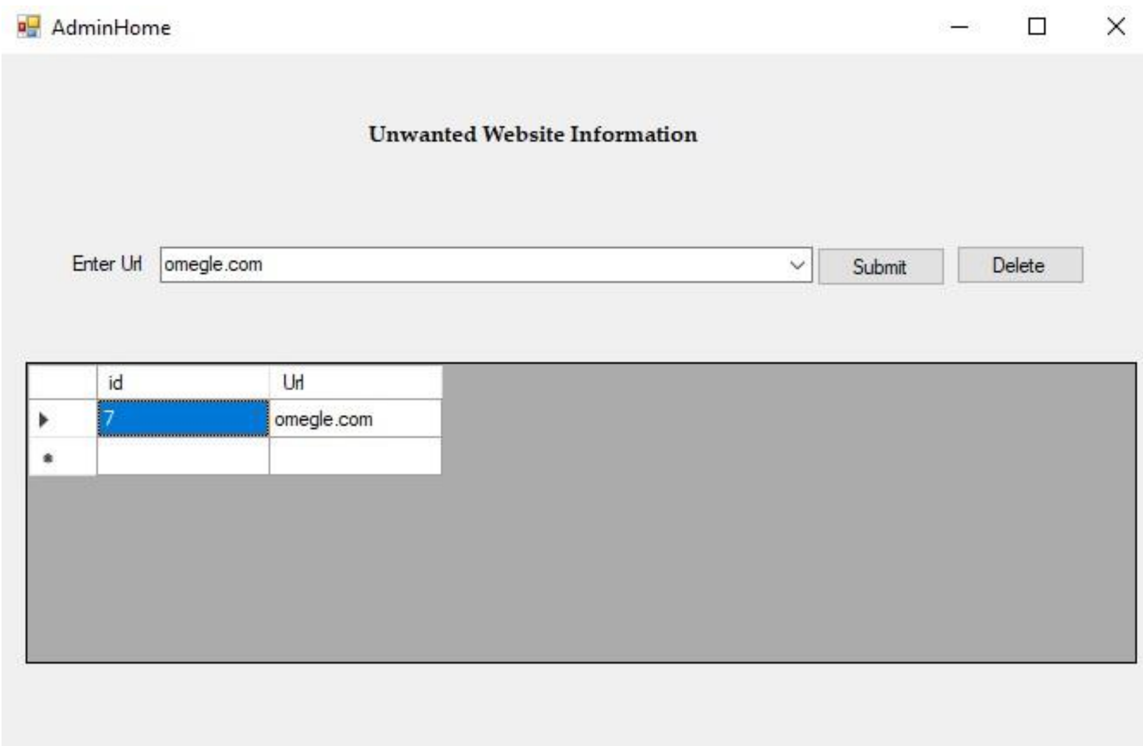


Fig A.2.3 Admin Home Window

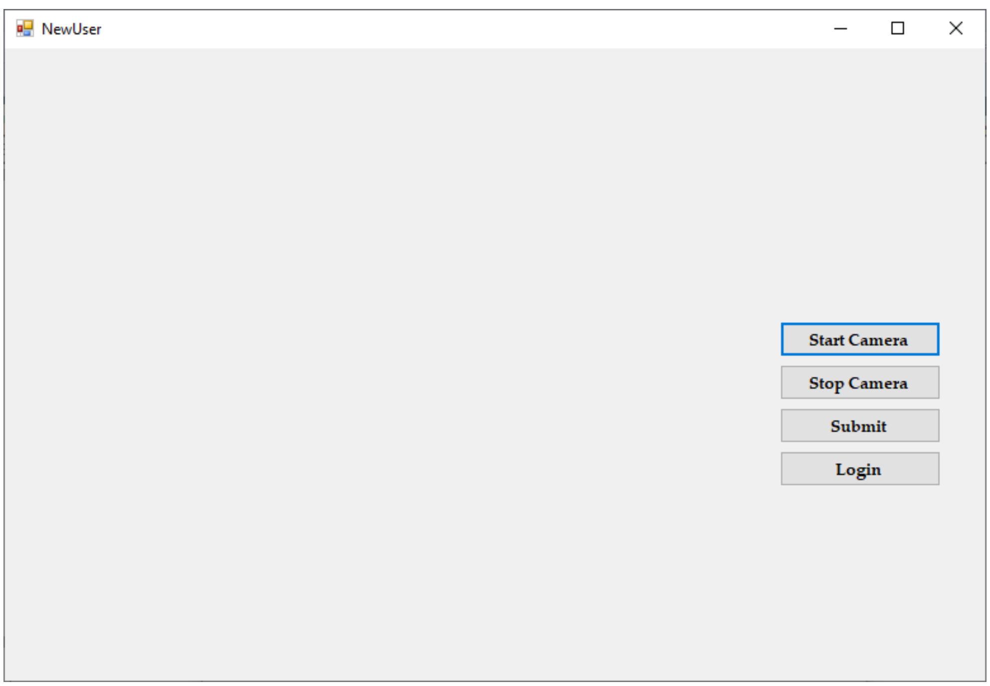


Fig A.2.4 New User Window

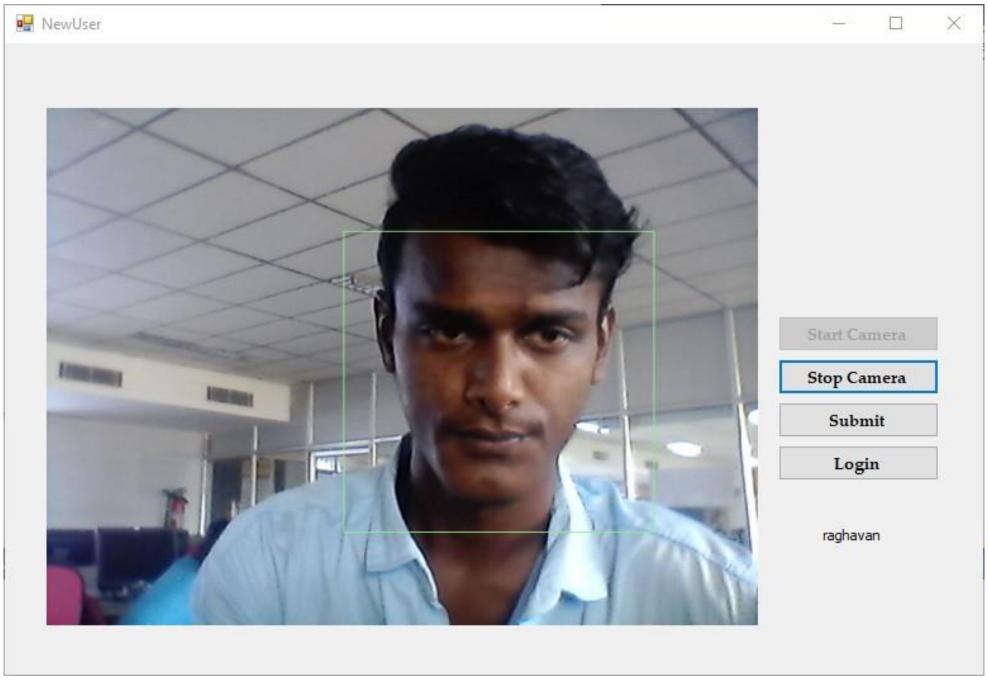


Fig A.2.5 Face Detection

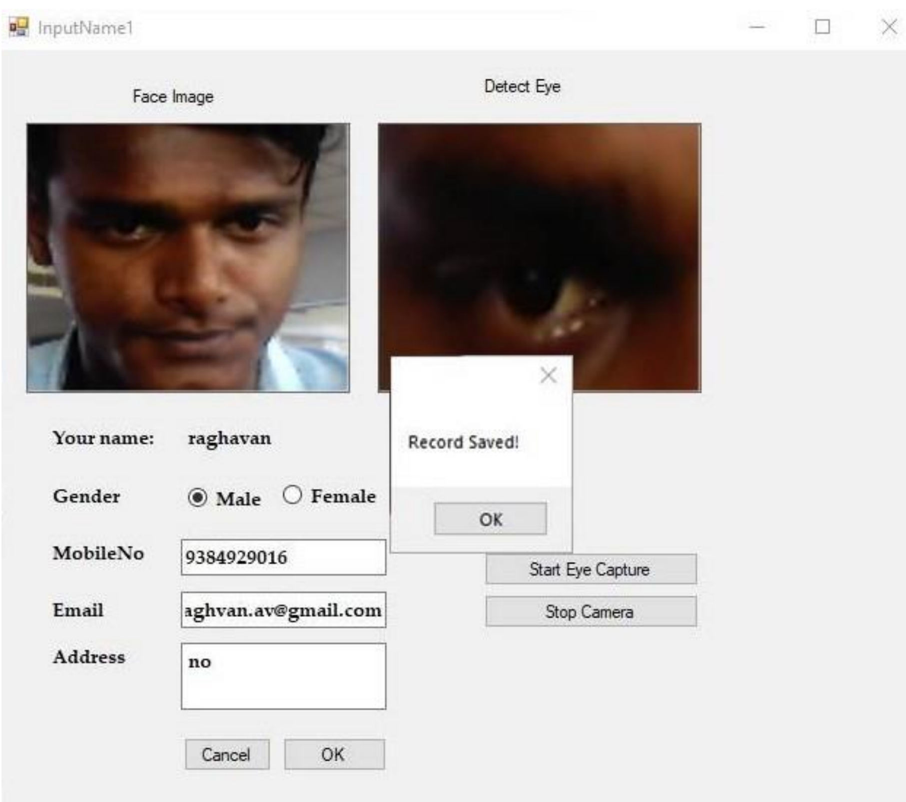


Fig A.2.6 User Information

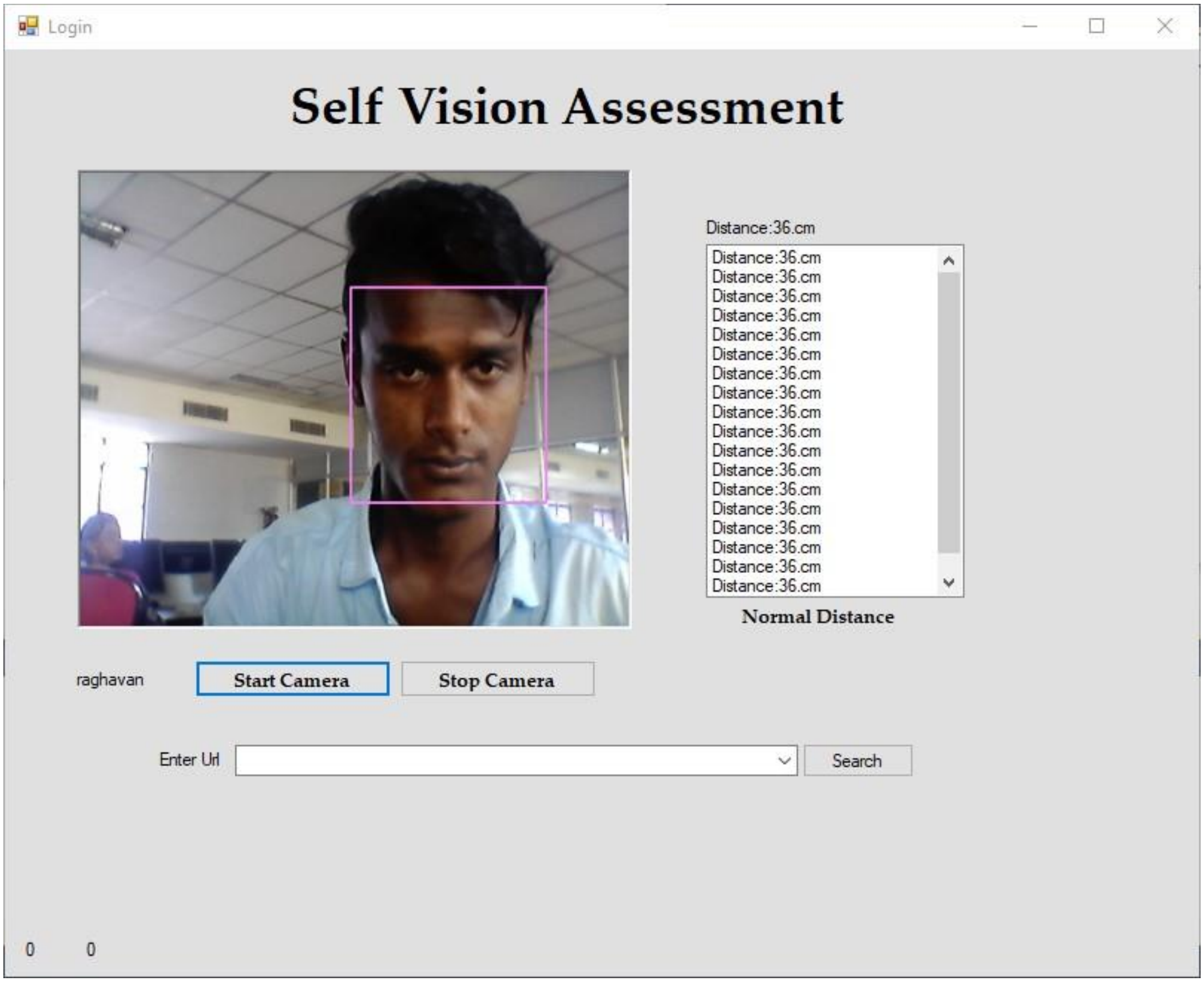


Fig A.2.7 Distance Feedback 1

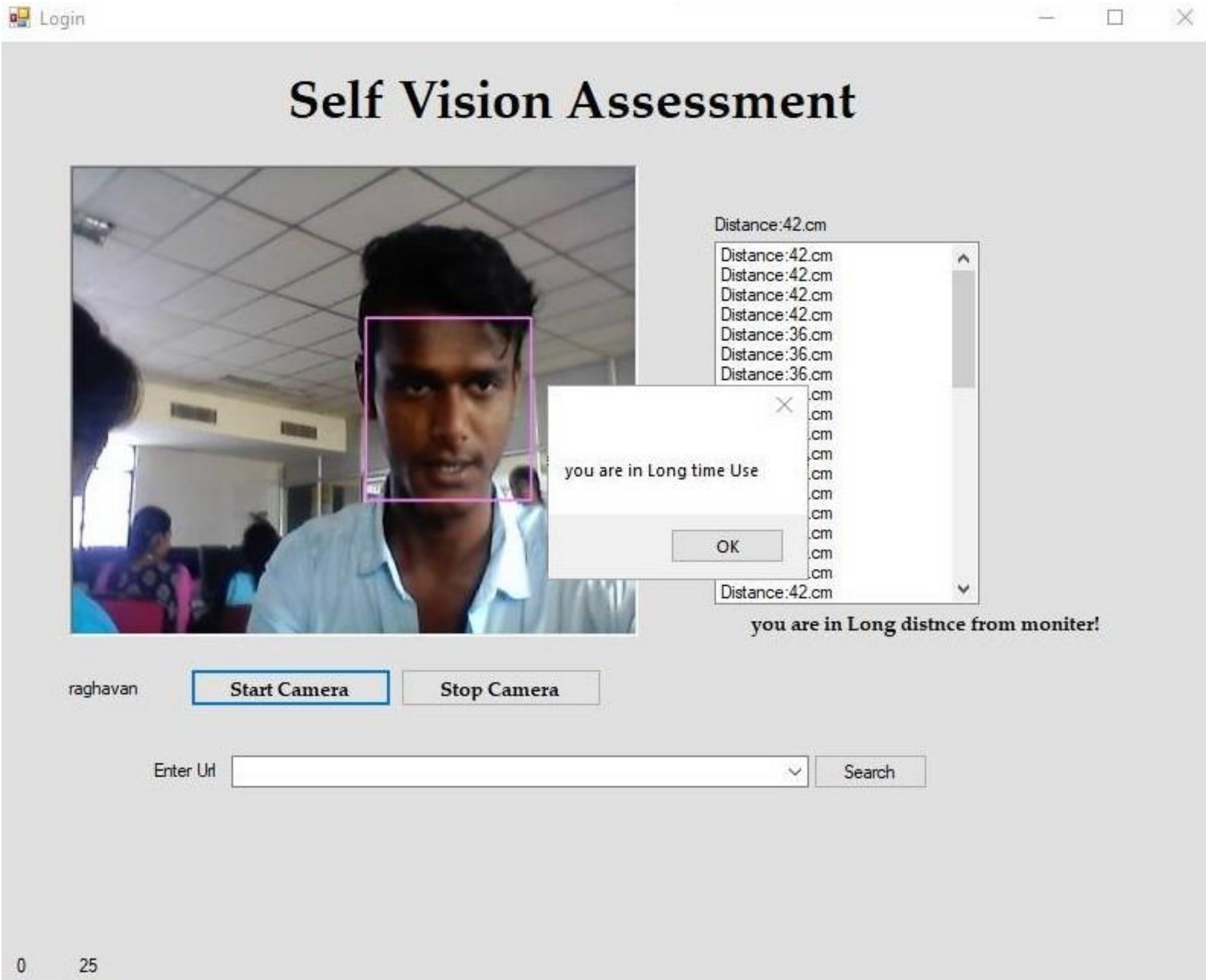


Fig A.2.8 Distance Feedback 2

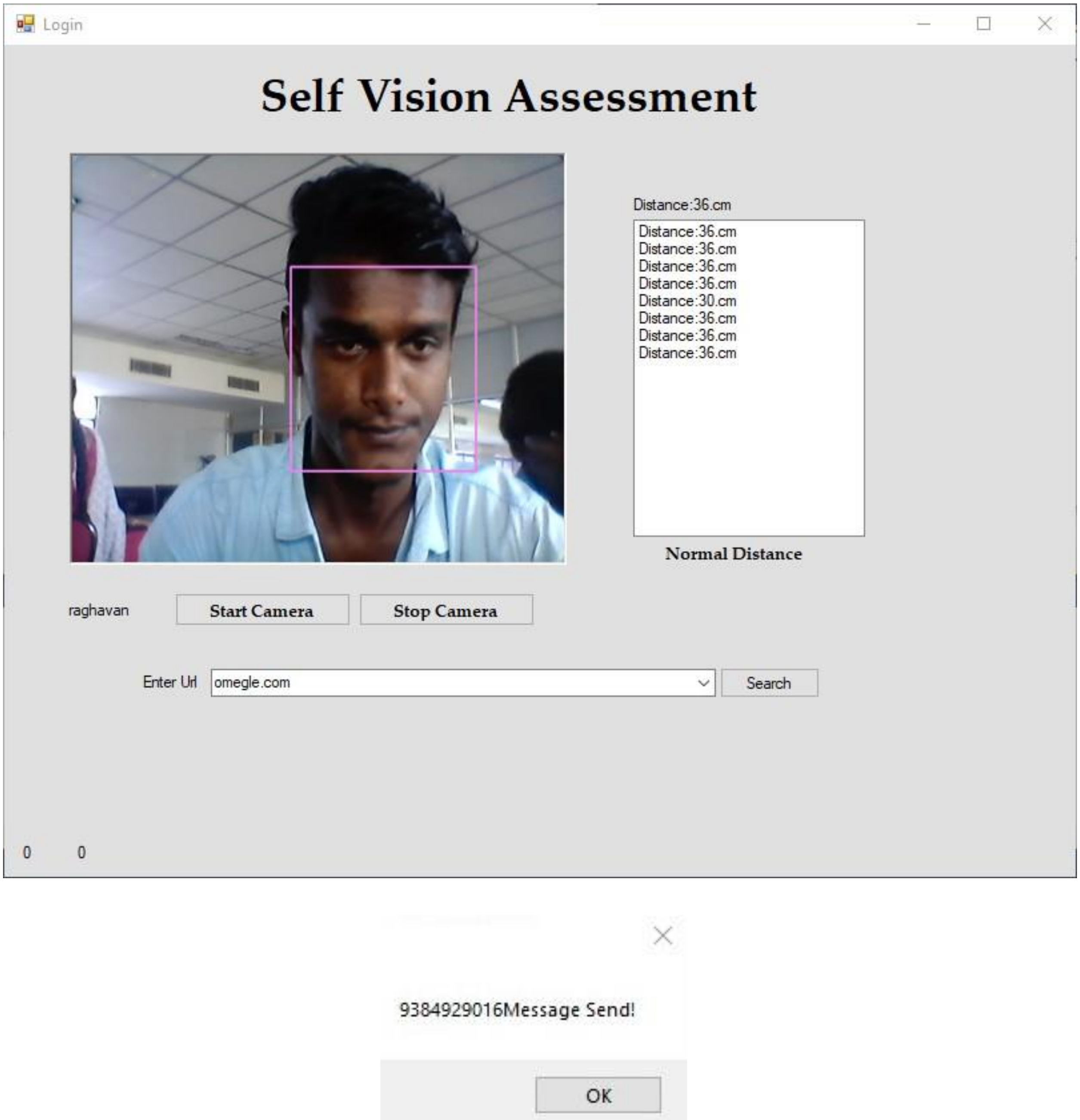


Fig A.2.9 Alert Message 1

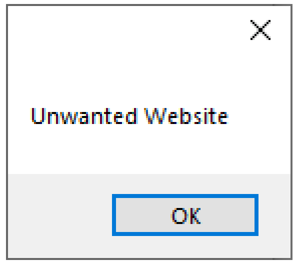


Fig A.2.10 Alert Message 2

Fig A.2.11 Acknowledgment

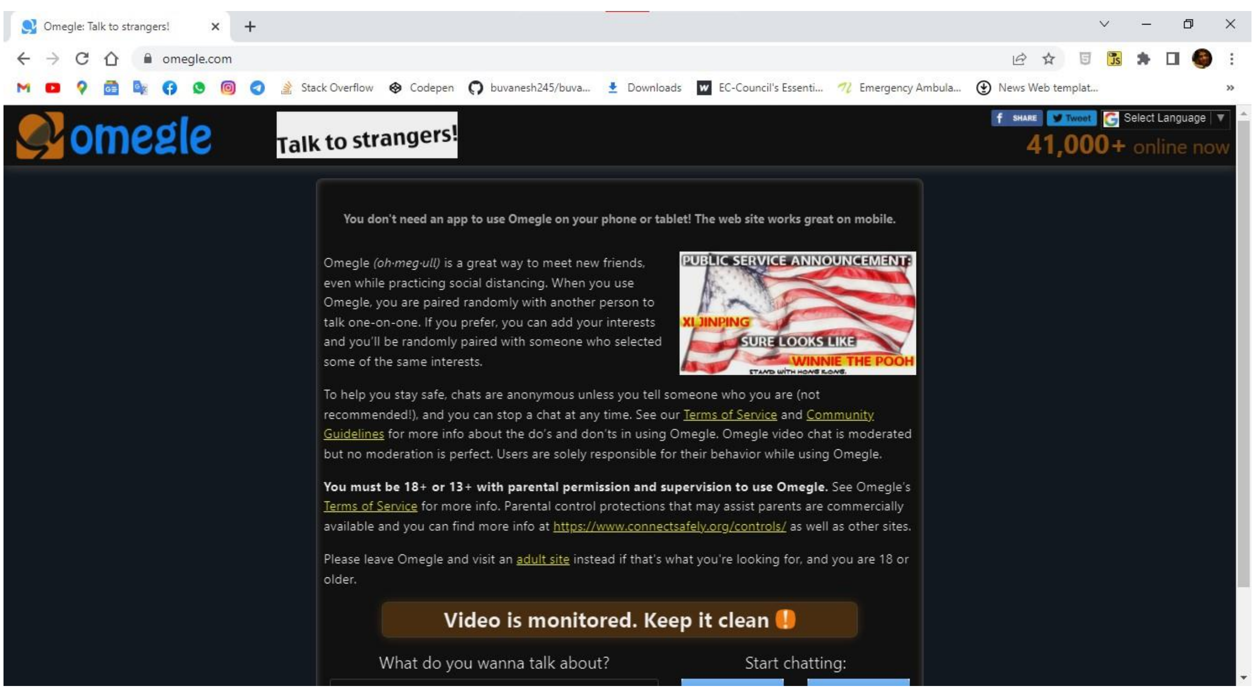


Fig A.2.12 Unwanted Website

REFERENCES

[1] Borghi, Guido, et al. "Face-from-depth for head pose estimation on depth images." IEEE transactions on pattern analysis and machine intelligence 42.3 (2018): 596-609.

[2] Cao, Zhiwen, et al. "A vector-based representation to enhance head pose estimation." Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. 2021.

[3] Diaz-Chito, Katerine, Aura Hernández-Sabaté, and Antonio M. López. "A reduced feature set for driver head pose estimation." Applied Soft Computing 45 (2016): 98-107.

[4] Drouard, Vincent, et al. "Head pose estimation via probabilistic high-dimensional regression." 2015 IEEE international conference on image processing (ICIP). IEEE, 2015.

[5] Kuhnke, Felix, and JornOstermann. "Deep head pose estimation using synthetic images and partial adversarial domain adaption for continuous label spaces." Proceedings of the IEEE/CVF International Conference on Computer Vision. 2019.

[6]Meyer, Gregory P., et al. "Robust model-based 3d head pose estimation." Proceedings of the IEEE international conference on computer vision. 2015.

[7] Ruiz, Nataniel, Eunji Chong, and James M. Rehg. "Fine-grained head pose estimation without keypoints." Proceedings of the IEEE conference on computer vision and pattern recognition workshops. 2018.

[8] Wang, Yujia, et al. "A deep coarse-to-fine network for head pose estimation from synthetic data." Pattern Recognition 94 (2019): 196-206.

[9] Yang, Tsun-Yi, et al. "Fsa-net: Learning fine-grained structure aggregation for head pose estimation from a single image." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019.

[10] Zhang, Hao, et al. "FDN: Feature decoupling network for head pose estimation." Proceedings of the AAAI Conference on Artificial Intelligence. Vol. 34. No. 07. 2020.