ADVANCED DYNAMIC TRACKING

Project Report Submitted

In Partial Fulfillment of the Requirements

For the Degree Of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

Submitted By

SYED SOHAIB UDDIN (1604-17-733-074)

SAYEED UDDIN SHOAIB (1604-17-733-083)

MOHAMMAD AZHAR ALI (1604-17-733-089)

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT

MUFFAKHAM JAH COLLEGE OF ENGINEERING & TECHNOLOGY

(Affiliated to Osmania University)

Mount Pleasant, 8-2-249, Road No 3, Banjara Hills, Hyderabad-34

2021

Date: 01 / 06 / 2021

CERTIFICATE

This is to certify that the project dissertation titled "**ADVANCED DYNAMIC TRACKING**" being submitted by

1. SYED SOHAIB UDDIN (1604-17-733-074)
2. SAYED UDDIN SHOAIB (1604-17-733-083)
3. MOHAMMAD AZHAR ALI (1604-17-733-089)

in Partial Fulfillment of the requirements for the award of the degree of BACHELOR OF ENGINEERING IN COMPUTER SCIENCE AND ENGINEERING in MUFFAKHAM JAH COLLEGE OF ENGINEERING AND TECHNOLOGY, Hyderabad for the academic year 2018-19 is the bonafide work carried out by them. The results embodied in this report have not been submitted to any other University or Institute for the award of any degree or diploma.

Signatures:

Internal Project Guide
(Dr. Krishna Keerthi Chennam)
(Associate Professor)

Head CSED
(Dr. A.A. Moiz Qyser)

External Examiner

DECLARATION

This is to certify that the work reported in the major project entitled "**ADVANCED DYNAMIC TRACKING**" is a record of the bonafide work done by us in the Department of Computer Science and Engineering, Muffakham Jah College of Engineering and Technology, Osmania University. The results embodied in this report are based on the project work done entirely by us and not copied from any other source.

1. Syed Sohaib Uddin (1604-17-733-074)
2. Sayeed Uddin Shoaib (1604-17-733-083)
3. Mohammad Azhar Ali (1604-17-733-089)

ACKNOWLEDGEMENT

Our hearts are filled with gratitude to the Almighty for empowering us with courage, wisdom and strength to complete this project successfully. We give him all the glory, honor and praise.

We thank our Parents for having sacrificed a lot in their lives to impart the best education to us and make us promising professionals for tomorrow.

We would like to express our sincere gratitude and indebtedness to our project supervisor **Dr. Krishna Keerthi Chennam** for her valuable suggestions and interest throughout the course of this project.

We are happy to express our profound sense of gratitude and indebtedness to **Prof. Dr. Ahmed Abdul Moiz Qyser,** Head of the Computer Science and Engineering Department, for his valuable and intellectual suggestions apart from educating guidance, constant encouragement right throughout our work and making us successful.

With a great sense of pleasure and privilege, we extend our gratitude to **Prof. Dr. Syed Shabbeer Ahmed,** Associated Head of Computer Science and Engineering Department, project in-charge, who offered valuable suggestions as a pre-requisite to carry out support in every step.

We are pleased to acknowledge our indebtedness to all those who devoted themselves directly or indirectly to make this project work a total success.

SYED SOHAIB UDDIN

SAYEED UDDIN SHOAIB

MOHAMMAD AZHAR ALI

ABSTRACT

Data even in its simple form can give multidimensional information. Having the capability to get the best of a given set of data while using it for solving problems is the mantra of technology and our role as an engineer. Video in itself is multi-dimensional data and has the capability to give gigantic amounts of information.

Noting the above, this project implements object tracking in computer vision over a network of cameras generating geo fences with occupancy map allowing the system to track movements of individuals, traffic and objects across any area with attention to features such as direction of motion, velocity, physical appearance etc. This would allow us to leverage the best of multivariate data.

Apart from being a strong business model with implementations in retail, commerce, defense, security and surveillance, this would be a leap in technology-surveillance domain.

In order to implement the above idea, the following are recognized as trivial steps. Object Tracking - With the use of available recognition models, objects of all classes can be identified and tracked. Geofencing - Tracking is a locomotion to be done every instance an object is in the surveillance zone. The zone is referred to as a Geofence. Multiview-Synchronization- Detection of objects in the geofences is enhanced via feed from various point sources. In order to aggregate the information, proper synchronization is required.

Video data is one of the most promising sources of data which is not anticipated with all the potential it carries. Hence this project is a non-GPS based solution and is unlike any other implementation. It will help many businesses, security agencies and the government in a lot of ways. The principal incorporated in the project is to track cum trace objects and create the route taken. There is a scope to even detect and study the movement and behavior of an individual.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

1.1 SURVEILLANCE

Surveillance is the monitoring of behavior, activities, or information for the purpose of information gathering, influencing, managing or directing. This can include observation from a distance by means of electronic equipment, such as closed-circuit television (CCTV), or interception of electronically transmitted information, such as Internet traffic. It can also include simple technical methods, such as human intelligence gathering and postal interception.

1.1.1 Human Tracking

Video surveillance systems not only effectively protect the security of public facilities and citizens, but also seamlessly help to transform to smart city, which has attracted more and more scientific researchers to invest huge funds in research related to intelligent video surveillance. It is observed that the main focus of the current research on intelligent video surveillance mainly lies on video object detection/tracking, and video object activity analysis/recognition.

The video object tracking is not only one of the most important techniques in intelligent video surveillance, but also the base of high-level video processing and applications such as the subsequent video object activity analysis and recognition. However, in the video object tracking, human tracking is the most challenging since human may vary greatly in appearance on account of changes in illumination and viewpoint, background clutter, occlusion, non-rigid deformations, intra-class variability in shape and pose. Human tracking includes human tracking within a camera and human tracking across multiple cameras.

When a person enters into the field of view (FOV) of a camera, human tracking within a camera is needed. However, when he/she leaves the FOV, the human information is no longer available, thus the limited FOV of a camera cannot meet the needs of wide-area human tracking. In order to widen the FOV, human tracking across multiple cameras has to be used since video streams across multiple cameras covering a wider range of areas, which helps to analyze global activities in the real world. Tracking human across multiple cameras includes two different scenarios, i.e., overlapping camera views and non-overlapping camera views. In the overlapping camera views' scenario, there is a common FOV area between two camera's views, and human located in the common area (as shown in the area between cameras 1 and 2 in Fig. 1) will appear simultaneously in both cameras' views. In the non-overlapping camera views' scenario, there is not a common FOV area between two cameras' views, i.e., every camera's view is completely disjointed, and human cannot be seen in the so-called blind area (as shown in the area between cameras 2 and 3 in Fig. 1). Compared with human tracking across overlapping cameras, human tracking across non-overlapping cameras will be more challenging and practical. As a result, human tracking over camera networks is necessary and quite challenging in the intelligent video surveillance.

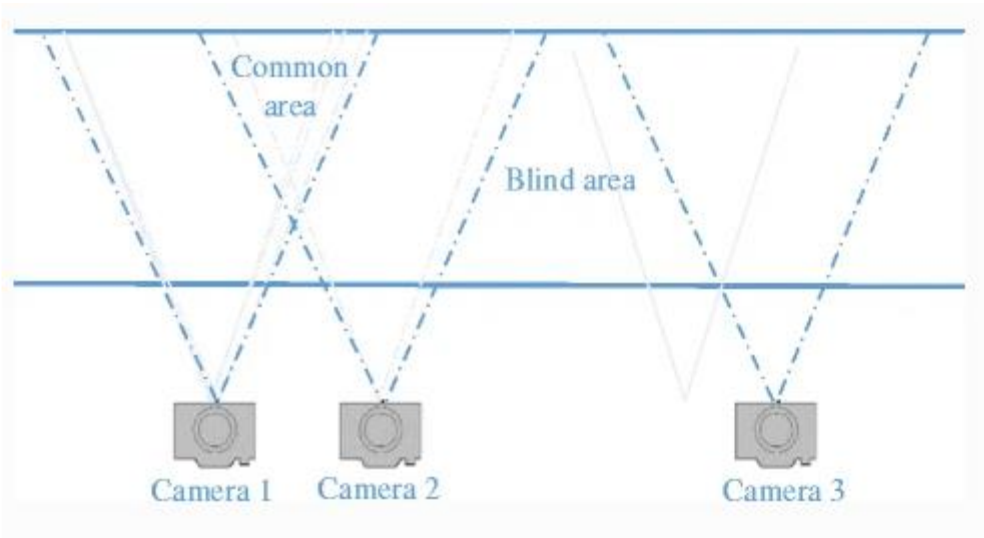


Fig. 1.1 Field of View

1.2 COMPUTER VISION

Computer vision is the field of computer science that focuses on replicating parts of the complexity of the human vision system and enabling computers to identify and process objects in images and videos in the same way that humans do.

Until recently, computer vision only worked in limited capacity. Thanks to advances in artificial intelligence and innovations in deep learning and neural networks, the field has been able to take great leaps in recent years and has been able to surpass humans in some tasks related to detecting and labeling objects. One of the driving factors behind the growth of computer vision is the amount of data we generate today that is then used to train and make computer vision better. Along with a tremendous amount of visual data (more than 3 billion images are shared online every day), the computing power required to analyze the data is now accessible. As the field of computer vision has grown with new hardware and algorithms so has the accuracy rates for object identification.

The goal of computer vision is to derive descriptive information about a scene by computer analysis of images of the scene. Vision algorithms can often serve as computational models for biological visual processes, and they also have many practical uses; but vision problems are often ill-defined, ill-posed, or computationally intractable; nevertheless, successes have been achieved in many specific areas like document processing and industrial inspection, for example. By limiting the domain of application, carefully choosing the task, using redundant data (multi-sensor, multi-frame), and applying adequate computing power, useful solutions to many vision problems can be obtained.

Methods of designing such solutions are the subject of the emerging discipline of vision engineering. With projected advances in sensor and computing technologies, the domains of applicability and ranges of problems that can be solved will gradually expand.

1.3 NEURAL NETWORKS IN COMPUTER VISION

A neural network is a series of algorithms that endeavors to recognize underlying relationships in a set of data through a process that mimics the way the human brain operates [2]. In this sense, neural networks refer to systems of neurons, either organic or artificial in nature. Neural networks can adapt to changing input; so the network generates the best possible result without needing to redesign the output criteria.

Neural networks contains layers of interconnected nodes. Each node is a perceptron and is similar to a multiple linear regression. The perceptron feeds the signal produced by a multiple linear regression into an activation function that may be nonlinear. In a multi-layered perceptron (MLP), perceptrons are arranged in interconnected layers. The input layer collects input patterns. The output layer has classifications or output signals to which input patterns may map.

1.3.1 Convolutional Neural Network (CNN)

CNN is a type of deep learning model for processing data that has a grid pattern, such as images, which is inspired by the organization of animal visual cortex and designed to automatically and adaptively learn spatial hierarchies of features, from low- to high-level patterns. CNN is a mathematical construct that is typically composed of three types of layers (or building blocks): convolution, pooling, and fully connected layers [3]. The first two, convolution and pooling layers, perform feature extraction, whereas the third, a fully connected layer, maps the extracted features into final output, such as classification. A convolution layer plays a key role in CNN, which is composed of a stack of mathematical operations, such as convolution, a specialized type of linear operation. In digital images, pixel values are stored in a two-dimensional (2D) grid, i.e., an array of numbers (Fig. 2), and a small grid of parameters called kernel, an optimizable feature extractor, is applied at each image position, which makes CNNs highly efficient for image processing, since a feature may occur anywhere in the image. As one layer feeds its output into the next layer, extracted features can hierarchically and progressively become more complex. The process of optimizing parameters such as kernels is called training, which is performed so as to minimize the difference between outputs and ground truth labels through an optimization algorithm called backpropagation and gradient descent, among others.

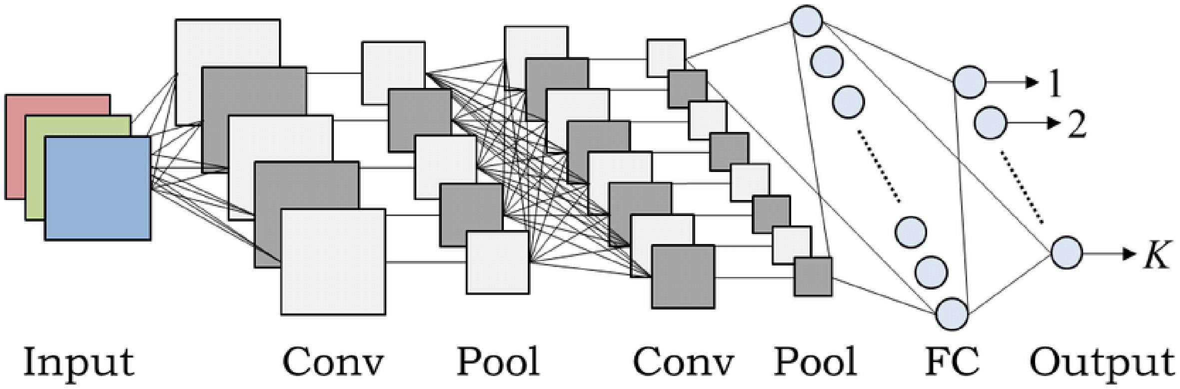


Fig. 1.3.1 CNN Architecture

The architecture of a ConvNet is analogous to that of the connectivity pattern of Neurons in the Human Brain and was inspired by the organization of the Visual Cortex. Individual neurons respond to stimuli only in a restricted region of the visual field known as the Receptive Field. A collection of such fields overlap to cover the entire visual area.

1.3.2 Residual Network (ResNet)

ResNets have an extremely simple yet extremely elegant idea. The idea was to add skip connections or shortcut connections that created a gradient highway. This enabled gradients to flow better during the backward step, and greatly increased convergence, time of training and reduced gradient explosion and vanishing [4].

The subtle beauty of ResNets is that the best-case scenario is when the skip connection actively adds to the output and computes useful information, and the worst-case scenario is when it simply ignores the skipped connection and is at worst the same as a network without a skip connection. So skip connections add so much value and have no downside.

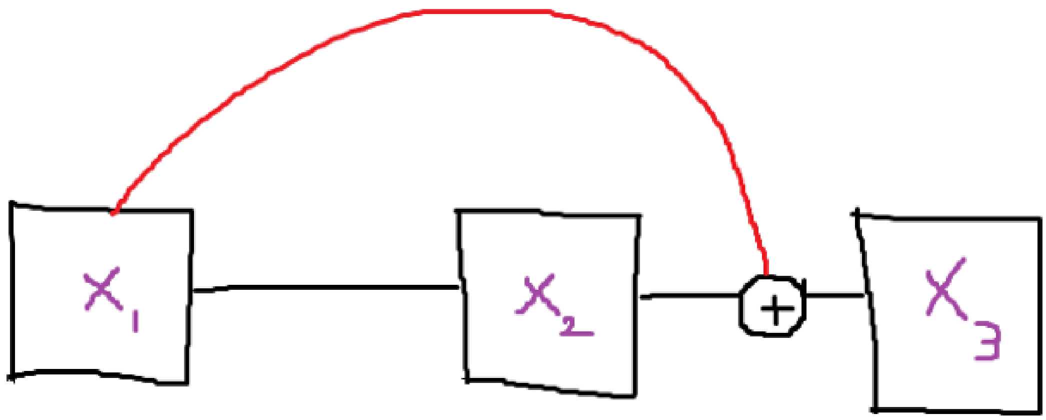


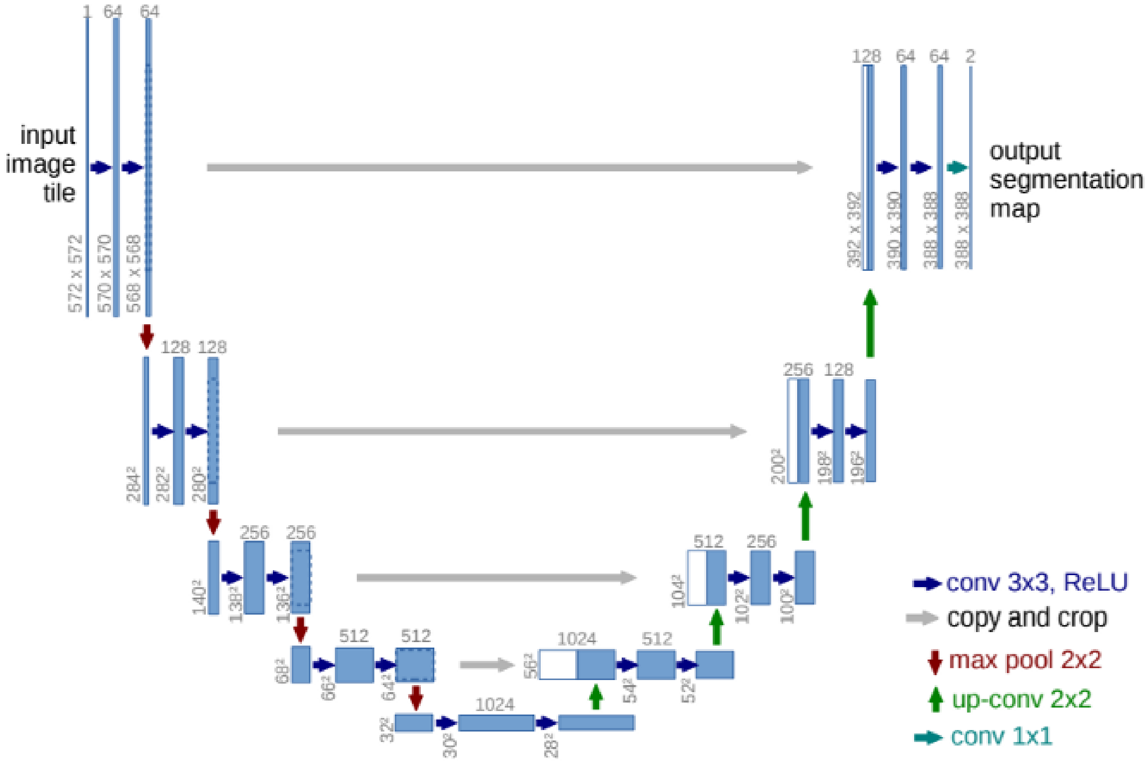
Fig. 1.3.2 Skip Connections

1.3.3 U-Net

U-Nets have 2 parts, the contracting path (downsampling path) and the expansive path (upsampling path). In a traditional image classification convolutional network, the image is fed into a network which performs convolutions and pooling operations, both of which reduce the height and width of the output but increase the depth of the output. With a loss in height and width, the depth gained adds features to the classification output.

However, in segmentation tasks, we want the output to be the same shape as the input image and the added features for labelling pixels. So the downsampling of a traditional Convolutional architecture is supplemented by an upsampling path, to add back the height and width of the image to the output, while maintaining the features. There are many upsampling methods, but the most common one used in most libraries is Transpose convolution upsampling.

Fig. 1.3.3 U-Net Architecture



1.3.4 YOLO

YOLO stands for you only look once. Previously, the popular method for object detection was to reuse classifiers to classify local regions of an image and use a sliding window approach to check if each region of an image has an object. YOLO shifted the paradigm by proposing object detection as a regression problem, where they only use a single network for the entire pipeline and process the whole image at once rather than in regions.

YOLO divides the input image into an SxS grid. And for each grid predicts if the center of an object is present within the grid. If the center of the object is in the grid, the grid will then predict a bounding box with 5 values, x, y, w, h, c. (x, y) are the coordinates of the center of the object relative to the grid, (w, h) is the width and height of the object relative to the whole image and (c) is the class of the object. YOLO is extremely fast. Since the paper frames detection as a regression problem, it doesn't need a complex pipeline. It reasons globally about the image when making predictions. Unlike sliding window and region proposal-based techniques, YOLO sees the entire image during training and test time so it implicitly encodes contextual information about classes as well as their appearance.

YOLO learns generalizable representations of objects. Since YOLO is highly generalizable it is less likely to break down when applied to new domains or unexpected inputs.

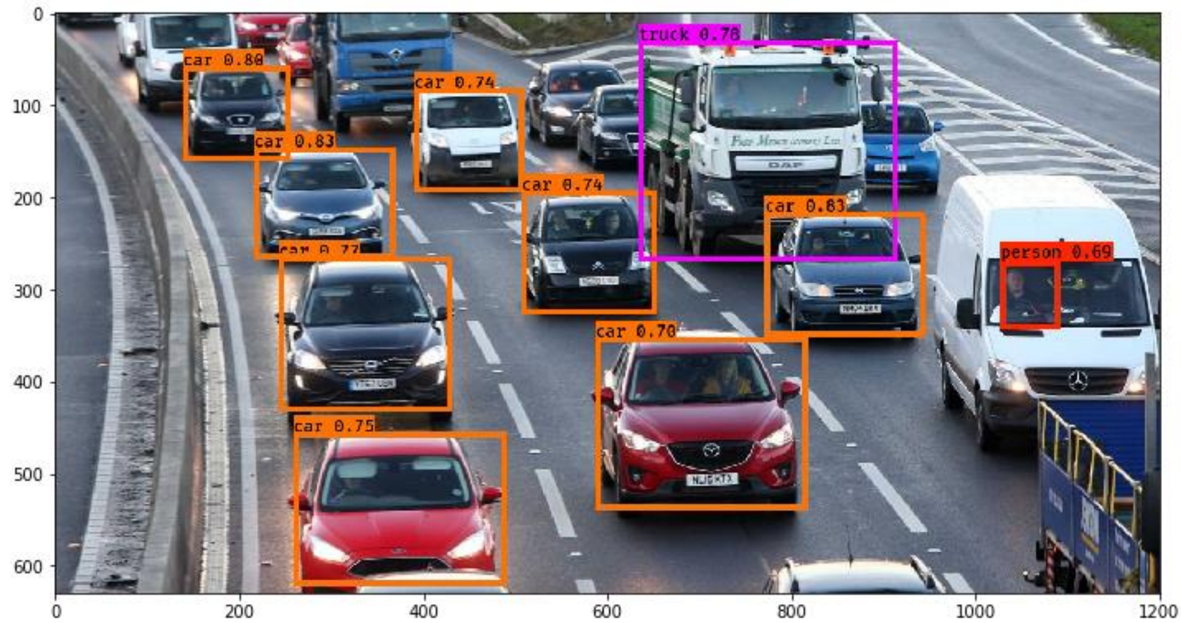


Fig. 1.3.4 YOLO

1.3.5 Generative adversarial network (GAN)

GANs are a neural network pair that are trained through an adversarial process. The 2 parts of a GAN are a Generator and a Critic/Discriminator. The role of the generator is to generate high-quality data that is similar to training data, and the role of the critic is to differentiate between the generated data and the real data. The objective function of the generator is to maximize the loss of the critic, and the objective function of the critic is to minimize its loss. Think of this process as analogous to a thief and the police. The thieves want to fool the police and keep improving their tools and techniques, and the police want to catch the thieves so they improve too. The generator is like the thief and the critic is like the police.

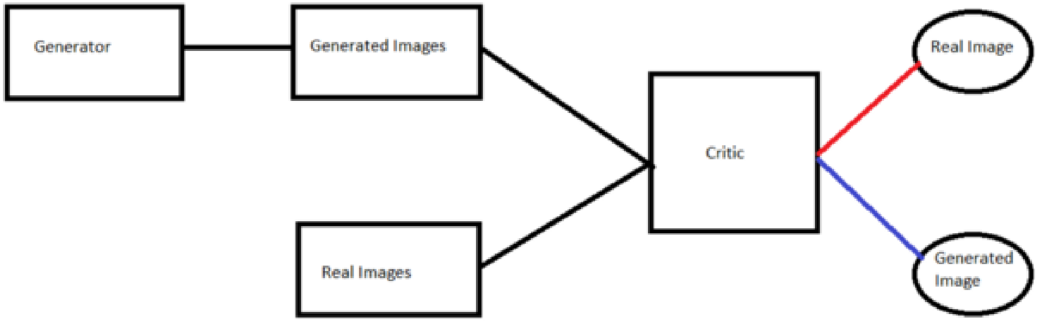


Fig. 1.3.5 GAN

1.4 HUMAN FACE RECOGNITION BASED ON CNN & AUGMENTED DATASET

With the rapid development of computer science and technology, artificial intelligence (AI) is increasingly prevalent in our daily life. Among others, deep neural network (DNN) has been successfully utilized in a variety of tasks e.g. character recognition, image recognition, and speech recognition, etc. In particular, face recognition based on DNN has been under intensive investigations in the past few years.

For example, a novel coupled mapping approach was proposed for the recognition of low resolution face images based on convolutional neural network (CNN), which is one of the architectures of DNN. Many face recognition experiments were carried out on the common face databases in Zhang et al. (2015), and the results showed that DNN can effectively extract the facial features based on diverse image preprocessing approaches. Moreover, joint Bayesian algorithm was applied for the face recognition.

The final experiment results can validate the superiority of the hybrid approach on the recognition accuracy. Nevertheless, it is worth noting that most of DNN-based face recognition methods are usually developed based on a large original dataset. Obviously, a large original dataset can provide more features of the face images, but it is usually difficult to be obtained in comparison with a small original dataset.

As a result, some of the existing successful methods could lead to poor performance on the small original dataset. Besides, though a larger original dataset can bring higher accuracy of the model and stronger generalization ability of the network, the data labelling of a large original dataset is really a tedious and time-consuming work. Therefore, from a practical point of view, it is a promising topic to develop the DNN-based face recognition methods on the small original dataset.

1.4.1 Preliminary of Convolutional Neural Network

Convolutional neural network (CNN) is a kind of neural network with convolutional layers. In general, CNN contains two kinds of hidden layers, i.e. convolutional layers and pooling layers, which are usually arranged alternately in the neural network.

Similar to biological neural network, the connection weights of CNN can be shared in the whole neural network, which can not only reduce the amount of the connection weights, but also simplify the complexity of the network model. Thus, the training time of CNN can be remarkably shortened in most cases. In particular, when an image is the input of CNN, the image can be put into the neural network directly to avoid several complicated works, such as feature extraction and data reconstruction.

Owing to the advantages of weight sharing, pooling and local receptive field, CNN has a robust performance on several image transformation operations, e.g. translation, rotation, and scaling. For the sake of completeness, the preliminary of CNN is briefly introduced in the rest of this section.

1.4.2 Activation function

The performance of a neural network is closely related to not only its structure but also the adopted activation function, which is usually selected as a nonlinear function to deal with some complex issues. Three frequently used activation functions in CNN are sigmoid, hyperbolic tangent (tanh) and rectifiedlinear unit (ReLU), which is illustrated in Figure 1.4.2.

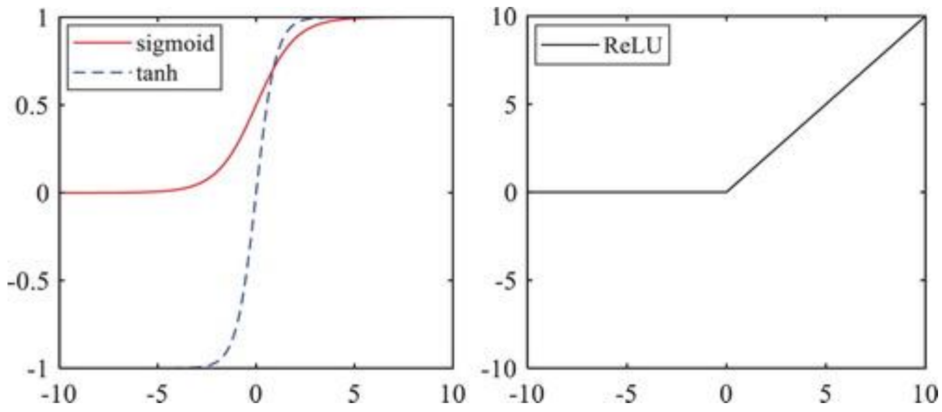


Figure 1.4.2 Activation functions.

1.4.2.1 Back propagation algorithm

Back propagation (BP) algorithm is one of the most frequently used algorithms to train a neural network, and the mapping of the input and output data is actually a nonlinear optimization problem of the connection weights [5]. Based on the gradient descent (GD) of BP algorithm, the connection weights of neural network can be updated iteratively by minimizing the mean square error (MSE) between the real and expected values of the output.

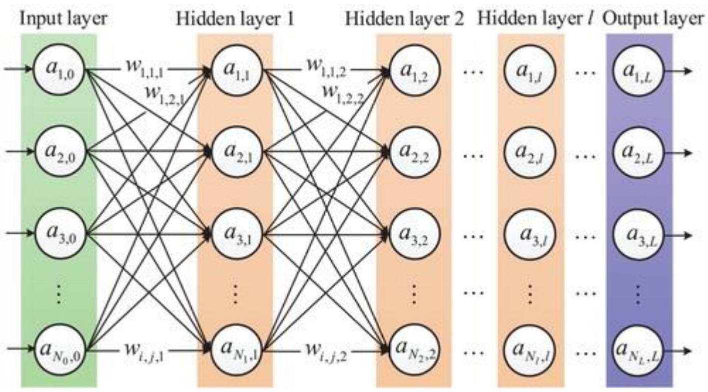


Figure 1.4.2.1 Structure of neural network.

1.4.2.2 Convolution operation

Convolution is a kind of mathematical operation that has been widely used in image processing. The result of convolution can be sorted as three modes, i.e. the modes of Full, Same and Valid, which can be utilized in different occasions. For example, Valid mode is usually applied for forward propagation to facilitate the feature extraction of image, and Full mode is often employed in the back propagation to obtain the optimal weights.

In the convolution operation, the operation of edge zeroing is implemented for the input image, where the layer amount of the edge can be determined according to the size of the convolution kernel. The purpose of edge zeroing is to ensure the rationality of the results, i.e. the elements of the input image and the convolution kernel can be weighted and summated sequentially. Additionally, the convolution kernel should be turned around and flipped up and down as shown in Figure 1.4.2.2, where the kernel is actually rotated 180 degrees around the center.

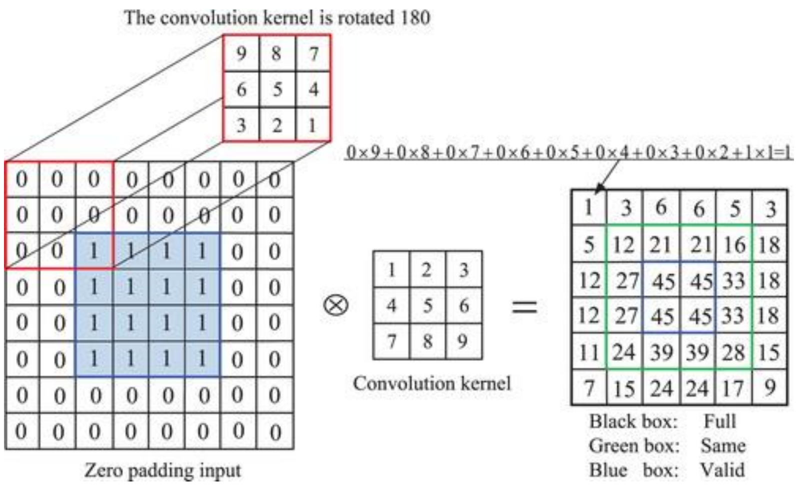


Figure 1.4.2.2 Convolution operation.

It is worth noting that convolution operation can achieve sparse multiplication and parameter sharing, which can compress the dimension of the input data. In comparison with DNN, it is not necessary for CNN to provide connection weights separately for all neurons of the input data. Actually, CNN can be regarded as a common feature extraction process like most neural networks used for feature extraction.

1.4.3 Receptive field

In CNN, the receptive field is the local connection field of a neuron in the hidden layer. Suppose that the input of the neural network is an image with 100×100 pixels and there are 100 neurons in the hidden layer, there will be 100×100×100 connection weights between the input and the hidden layers if every pixels of the image are connected to all neurons of the hidden layer as shown in Figure 1.4.3(a). There is no doubt that the huge computation load will decrease the training efficiency of the neural network.

In contrast, if each hidden neuron is connected to a local field of the input image (e.g. 10×10 pixels) as shown in Figure 1.4.3(b), the amount of connection weights will be reduced to 10×10×100, which is 1/100 of the full connection case.

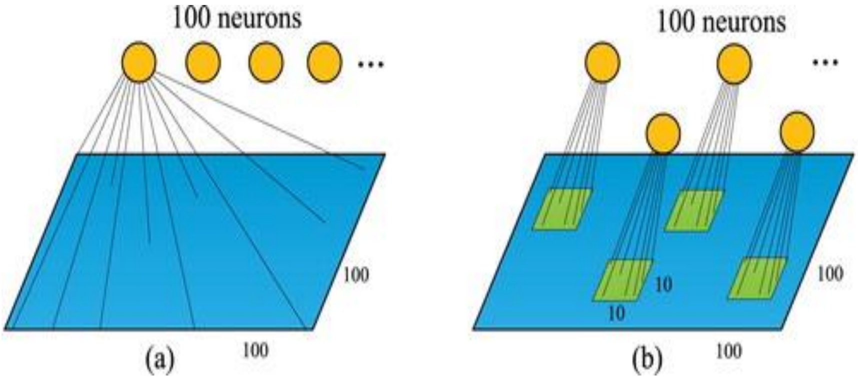


Figure 1.4.3 Receptive field.

In practice, the connection weights can be further reduced by using the shared weight method, i.e. all neurons have the same weights in one convolution kernel. Thus, the amount of connection weights can be reduced from 10×10×100 to 10×10. Accordingly, the training speed of the neural network can also be significantly promoted.

1.4.4 Pooling

The pooling layers, which are usually located behind the convolutional layers, are mainly used to compress the output feature data of the convolutional layers. After the pooling layer, the improved output results can reduce the likelihood of over-fitting in the neural network. Besides, the feature of image can be further extracted through pooling operation without influencing on the information acquisition of the image.

Actually, pooling is a reduction processing of the image, which can be classified as mean-pooling, max-pooling, overlapping-pooling, stochastic-pooling, and global average pooling. For instance, mean-pooling can extract the average value of the feature points and has the effect of maintaining the relative background; while max-pooling can extract the maximum value of the feature points and achieve better texture extraction. Specifically, for the mean-pooling, if a feature map with size of 4×4 is sampled by using a kernel with size of 2×2, the output is a feature map with size of 2×2 as shown in Figure 5.

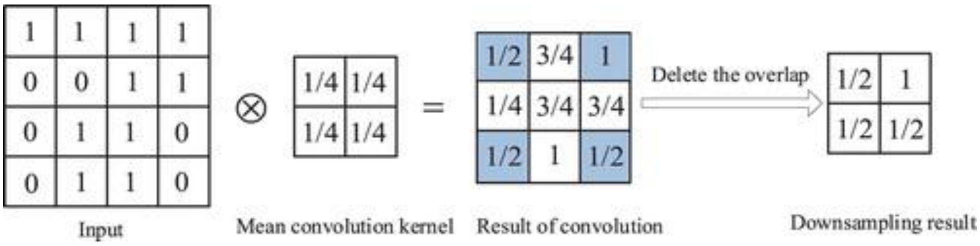


Figure 1.4.4 Pooling operation.

1.4.5 FACE RECOGNITION METHOD

1.4.5.1 CNN model for face recognition

A CNN model is developed to improve the accuracy of face image classification. The structure of the model is similar to the classical LeNet-5 model, but they are different on some parameters of the model, such as input data, network width and full connection layer. The developed CNN is composed of two convolutional layers (C1 and C2) and two pooling layers (S1 and S2). These layers are arranged alternately in the form of C1-S1-C2-S2 as sketched in Figure 1.4.5.1 [6].

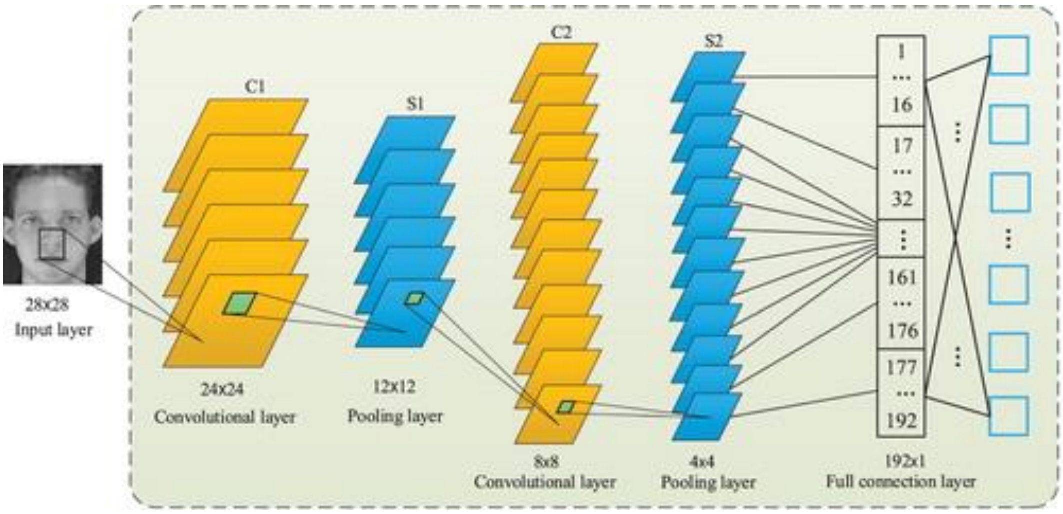


Figure 1.4.5.1 CNN model for face recognition.

There is only one feature map in the input layer, which is used to put the normalized face image into the CNN model. C1 is the first convolutional layer that includes 6 feature maps, in which each neuron is convoluted with a randomly generated convolution kernel with size of

5×5. S1 is the first pooling layer, whose output is 6 feature maps calculated based on the output of previous layer. Each element in the feature map is connected with the mean convolution kernel of the corresponding feature map in C1 layer, and the receptive fields of the elements will not overlapped with each other. C2 and S2 are, respectively, the second convolutional layer and pooling layer, both of which have 12 feature maps and similar calculation steps with their previous counterparts. Moreover, a fully connected single-layer perceptron is placed between the S2 layer and the output layer. As shown in Figure 6, the final output is a 40-dimensional vector for the face recognition of 40 individuals, where the sigmoid function is used for the multi-label classification.

1.4.5.2 Dataset and its augmentation

Olivetti Research Laboratory (ORL) face dataset is a collection of 400 human face images from 40 individuals. Each individual devotes 10 face images in diverse states to the dataset. The size of each image is 92×112 pixels, and each of the images is saved as the file types of BMP and PGM. ORL face dataset is a widely used standard face dataset that is easier to be labelled than others such as MIT or Yale face datasets. However, the amount of images is not abundant to train the deep neural network for accurate face recognition. To deal with this problem, the image amount of the dataset is augmented by using four data augmentation methods, including horizontal flip, shift, scaling, and rotation as shown in Figure 1.4.5.2-(a).

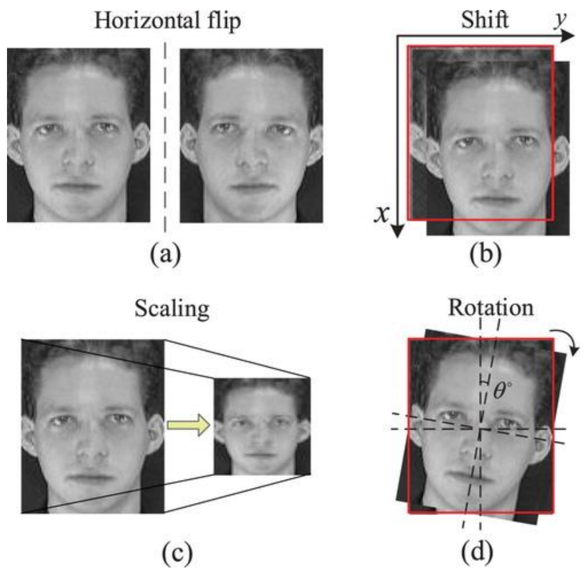


Figure 1.4.5.2-(a) Four methods for data augmentation.

Actually, the dataset can be augmented tremendously by tuning the parameters of the augmentation methods, see Figure 1.4.5.2–(b) for example. In this paper, the dataset is augmented by 1000 times after aforementioned operations. Then, the images are scaled, normalized and labelled before they are put into the face recognition system.

It can be predictable that the augmented dataset can not only reduce the probability of over-fitting but also improve the robustness of the system.

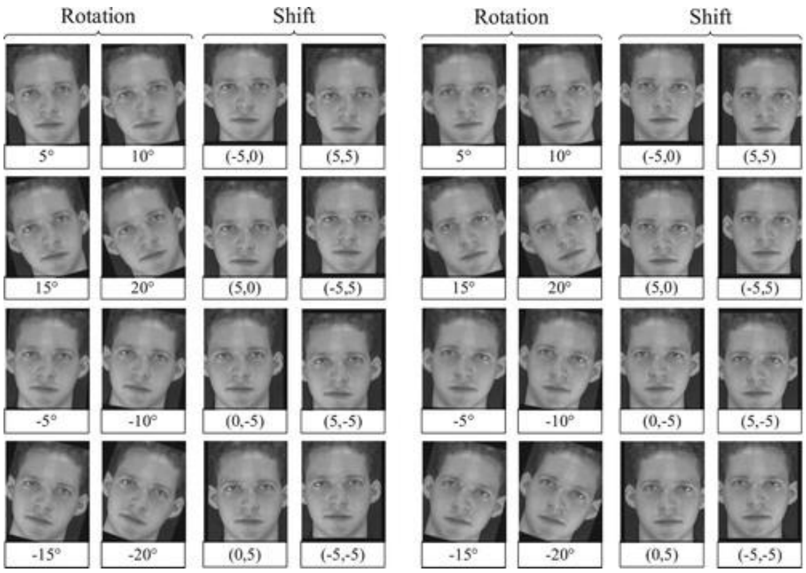


Figure 1.4.5.2–(b) Data augmentation for the first face.

1.5 PROBLEM STATEMENT

To develop a system that dynamically tracks movements in a given area using camera and triggering necessary actions, this project implements object tracking in computer vision over a network of cameras generating geofences with occupancy map allowing the system to track movements of individuals, traffic and objects across any area with attention to features such as direction of motion, velocity, physical appearance etc. This would allow us to leverage the best of multivariate data.

1.6 OBJECTIVE

To introduce a new surveillance system for:

1. Continuous tracking of people throughout the area.
2. Anomaly detection reporting.
3. Issuing alerts on unauthorized movements.
4. GPU availability with CUDA on local machines for video processing.
5. Machine learning (Computer vision) with Python using face_recognition library

2. LITERATURE SURVEY

Tracking people via cameras has been a hectic task in the past and required man work in abundance. Over the years, object tracking and detection has emerged as one of the most important aspects of UAV applications such as surveillance, reconnaissance, etc. We present a tracking-by-detection approach for real-time Multiple Object Tracking (MOT) of footage from a drone-mounted camera. Tracking-by-detection is the leading paradigm considering its computational effectiveness and improved detection algorithms. Our algorithm builds on the baseline Deep SORT algorithm implemented for MOT benchmarks. However, to circumvent the challenges posed by videos captured from a significant height we use a combination of YOLOv3 and RetinaNet for generating detections in each frame.

2.1 Iconography Classification

Iconography in art is the discipline that studies the visual content of artworks to determine their motifs and themes and to characterize the way these are represented. It is a subject of active research for a variety of purposes, including the interpretation of meaning, the investigation of the origin and diffusion in time and space of representations, and the study of influences across artists and art works.

With the proliferation of digital archives of art images, the possibility arises of applying Computer Vision techniques to the analysis of art images at an unprecedented scale, which may support iconography research and education. In this they introduced a novel paintings data set for iconography classification and present the quantitative and qualitative results of applying a Convolutional Neural Network (CNN) classifier to the recognition of the iconography of artworks.

Qualitative analysis of the results shows that the CNN focuses on the traditional iconic motifs that characterize the representation of each saint and exploits such hints to attain correct identification. The ultimate goal of our work is to enable the automatic extraction, decomposition, and comparison of iconography elements to support iconographic studies and automatic art work annotation.

2.2 Yolo: Real time object detection

All prior detection systems repurpose classifiers or localizers to perform detection. They apply the model to an image at multiple locations and scales. High scoring regions of the image are considered detections.

We apply a single neural network to the full image. This network divides the image into regions and predicts bounding boxes and probabilities for each region. These bounding boxes are weighted by the predicted probabilities. Finally, we can threshold the detections by some value to only see high scoring detections.

The model has several advantages over classifier-based systems. It looks at the whole image at test time so its predictions are informed by global context in the image. It also makes predictions with a single network evaluation unlike systems like R-CNN which require thousands for a single image. This makes it extremely fast, more than 1000x faster than R-CNN and 100x faster than Fast R-CNN.

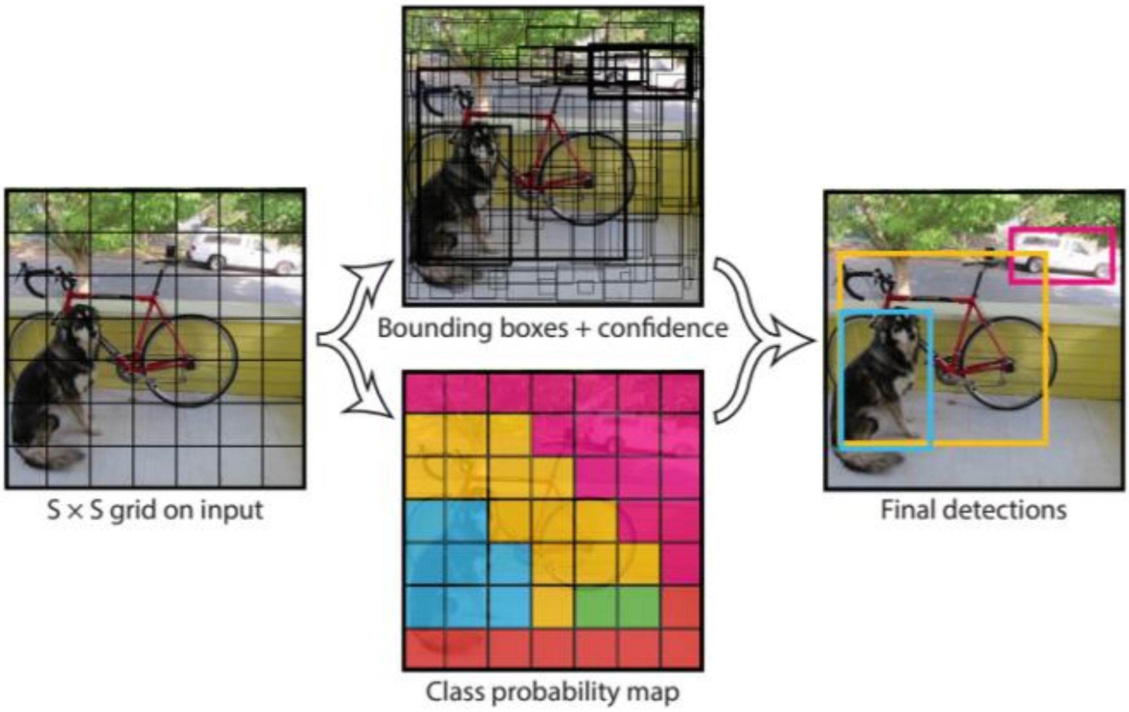


Fig 2.2 Working of YOLO

2.3 SoDA: Multi-Object Tracking with Soft Data Association

Robust multi-object tracking (MOT) is a prerequisite fora safe deployment of self-driving cars. Tracking objects, however, remains a highly challenging problem, especially in cluttered autonomous driving scenes in which objects tend to interact with each other in complex ways and frequently get occluded. It uses attention to compute track embeddings that encode the spatiotemporal dependencies between observed objects. This attention measurement encoding allows our model to relax hard data associations, which may lead to unrecoverable errors. Instead, our model aggregates information from all object detections via soft data associations.

The resulting latent space representation allows our model to learn to reason about occlusions in a holistic data-driven way and maintain track estimates for objects even when they are occluded. Our experimental results on the Waymo OpenDataset suggest that our approach leverages modern large-scale datasets and performs favorably compared to the state of the art in visual multi-object tracking.

2.4 Caffe: Convolutional Architecture for Fast Feature Embedding

Caffe provides multimedia scientists and practitioners with a clean and modifiable framework for state-of-the-art deep learning algorithms and a collection of reference models. The framework is a BSD-licensed C++ library with Python and MATLAB bindings for training and deploying general-purpose convolutional neural networks and other deep models efficiently on commodity architectures. Caffe fits industry and internet-scale media needs by CUDA GPU computation, processing over 40 million images a day on a single K40 or Titan GPU (≈ 2.5 ms per image).

By separating model representation from actual implementation, Caffe allows experimentation and seamless switching among platforms for ease of development and deployment from prototyping machines to cloud environments. Caffe is maintained and developed by the Berkeley Vision and Learning Center (BVLC) with the help of an active community of contributors on GitHub. It powers ongoing research projects, large-scale industrial applications, and startup prototypes in vision, speech, and multimedia.

2.5 You Only Look Twice: Rapid Multi-Scale Object Detection

Detection of small objects in large swaths of imagery is one of the primary problems in satellite imagery analytics. While object detection in ground-based imagery has benefited from research into new deep learning approaches, transitioning such technology to overhead imagery is nontrivial. Among the challenges is the sheer number of pixels and geographic extent per image: a single DigitalGlobe satellite image encompasses >64 km2 and over 250 million pixels. Another challenge is that objects of interest are minuscule (often only ~10 pixels in extent), which complicates traditional computer vision techniques. To address these issues, they proposed a pipeline (You Only Look Twice, or YOLT) that evaluates satellite images of arbitrary size at a rate of >0.5 km2/s. The proposed approach can rapidly detect objects of vastly different scales with relatively little training data over multiple sensors. We evaluate large test images at native resolution, and yield scores of F1 > 0.8 for vehicle localization. We further explore resolution and object size requirements by systematically testing the pipeline at decreasing resolution, and conclude that objects only ~5 pixels in size can still be localized with high confidence.

2.6 Point Linking Network for Object Detection

Object detection is a core problem in computer vision. With the development of deep ConvNets, the performance of object detectors has been dramatically improved. The deep ConvNets based object detectors mainly focus on regressing the coordinates of bounding box, e.g., Faster-R-CNN, YOLO and SSD. Different from these methods that considering bounding box as a whole, we propose a novel object bounding box representation using points and links and implemented using deep ConvNets, termed as Point Linking Network (PLN). Specifically, we regress the corner/center points of bounding-box and their links using a fully convolutional network; then we map the corner points and their links back to multiple bounding boxes; finally an object detection result is obtained by fusing the multiple bounding boxes. PLN is naturally robust to object occlusion and flexible to object scale variation and aspect ratio variation. In the experiments, PLN with the Inception-v2 model achieves state-of-the-art single-model and single-scale results on the PASCAL VOC 2007, the PASCAL VOC 2012 and the COCO detection benchmarks without bells and whistles.

3. SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

In the security world, there are two types of cameras, the first one is Analog surveillance camera and another one is Network surveillance camera or IP camera. These cameras send video signals to DVR or NVR. If it is a DVR, it converts the analog signal to digital and stores it in the DVR. If you are using NVR, it just stores the video in NVR.

The main component of a surveillance system is the DVR/NVR, which records live video feed and gives the ability to play recorded video anytime when you want. DVR/NVR uses a hard drive to store the videos coming from cameras. They are connected to a Monitor to view the video feed of each camera. DVR/NVR can also be connected to a network or modem so as to make the video feed accessible over the internet.

Fig. 3.1 Existing System

These systems have all the basic features of recording, storing and viewing the video stream from the cameras.

3.1.1 Problems with Existing System

3.1.1.1 Lack of object detection

Surveillance systems are not equipped with any object recognition software which can help keep an eye on goods, personal items or people in real-time.

3.1.1.2 Adaptation to system changes

The flexibility to extend or update the necessary services is not provided in the current system. These systems are rigid and only give access to basic things or services provided at the time of buying. The systems that are currently used resist updates or changes to given services.

3.1.1.3 Absence of object tracking service

The existing surveillance systems cannot perform the task of taking an initial set of object detections, creating a unique ID for each of the initial detections, and then tracking each of the objects as they move around frames in a video, maintaining the ID assignment.

3.1.1.4 Movement tracing for existing entities

Existing systems fail to record or capture movement of the objects that are present in the camera frames. The inability of the surveillance system results in failure to provide a path taken by any particular object.

3.2 PROPOSED SYSTEM

Our system employs a Facial recognition technique to detect each person present in the frames and assign IDs. It stores all the detected faces with their associated ID in the database for future reference. As the frames are analysed, the faces are compares with existing faces present in the database for:

- Generating the log of each face detected.

- Checking the occurrence of any new entity in the frames.
- Tracking of each person using the generated log.
- In addition to this, the system also provides a way to get the path taken by a particular person using the generated log.

3.2.1 Advantages

- Unique Object detection using face attributes.
- Tracking of each person using their unique ID
- Getting the path taken by each person.

3.2.2 Limitations

The main problem is that there is a lot of computation required to process the frames which involve the detection of all the objects present in the frame, storing each of these objects and tracking these objects in the next frames with the knowledge of previous frames. Due to heavy computation the FPS (frames per second) of the output stream may decrease to an extent. With better processors this issue can be resolved.

3.3 FEASIBILITY STUDY

Feasibility study is the test of a system proposal according to its workability, impact on the organization, ability to meet user needs, and effective use of resources. It focuses on the evaluation of existing systems and procedures analysis of alternative candidate system cost estimates. Feasibility analysis was done to determine whether the system would be feasible.

The development of a computer-based system or a product is more likely plagued by resources and delivery dates. Feasibility study helps the analyst to decide whether or not to proceed, amend, postpone or cancel the project, particularly important when the project is large, complex and costly. Once the analysis of the user requirement is complete, the system has to check for the compatibility and feasibility of the software package that is aimed at. An important outcome of the preliminary investigation is the determination that the system requested is feasible.

3.3.1 Types of Feasibility

- ➢ **Technical Feasibility**
- ➢ **Operational Feasibility**
- ➢ **Economical Feasibility**

3.3.1.1 Technical Feasibility

The application can be developed with the current equipment and has the technical capacity to hold the data required by the system.

- This technology supports the modern trends of technology.
- Easily accessible, more secure technologies.

Technical feasibility on the existing system and to what extent it can support the proposed addition.
We can add new modules easily without affecting the Core Program. Most of the parts are running in the server using the concept of stored procedures.

3.3.1.2 Operational Feasibility

This proposed system can easily be implemented, as this is based on HTML coding. The database created is with MongoDB which is more secure and easy to handle. The resources that are required to implement/install these are available.
The personnel of the organization already has enough exposure to computers. So, the project is operationally feasible and user friendly.

3.3.1.3 Economical Feasibility

If benefits outweigh costs, then the decision is made to design and implement the system. An entrepreneur must accurately weigh the cost versus benefits before taking an action. This system is more economically feasible and budget friendly which assesses the brain capacity with quick & online tests. So, it is economically a good project.

3.4 EFFORT AND COST ESTIMATION:

COCOMO (Constructive Cost Model) is a regression model based on LOC, i.e. number of Lines of Code. It is a procedural cost estimate model for software projects and often used as a process of reliably predicting the various parameters associated with making a project such as size, effort, cost, time and quality. It was proposed by Barry Boehm in 1970 and is based on the study of 63 projects, which make it one of the best documented models.
The key parameters which define the quality of any software products, which are also an outcome of the COCOMO are primarily Effort & Schedule:

- **Effort:** Amount of labour that will be required to complete a task. It is measured in person-months units.

- **Schedule:** Simply means the amount of time required for the completion of the job, which is, of course, proportional to the effort put. It is measured in the units of time such as weeks, months.

Different models of COCOMO have been proposed to predict the cost estimation at different levels, based on the amount of accuracy and correctness required. All of these models can be applied to a variety of projects, whose characteristics determine the value of constant to be used in subsequent calculations. These characteristics pertaining to different system types are mentioned below.

Boehm's definition of organic, semidetached, and embedded systems:

1. **Organic –** A software project is said to be an organic type if the team size required is adequately small, the problem is well understood and has been solved in the past and also the team members have a nominal experience regarding the problem.

2. **Semi-detached –** A software project is said to be a Semi-detached type if the vital characteristics such as team-size, experience, knowledge of the various programming environment lie in between that of organic and embedded. The projects classified as Semi-Detached are comparatively less familiar and difficult to develop compared to

the organic ones and require more experience and better guidance and creativity. Eg: Compilers or different Embedded Systems can be considered of Semi-Detached type.

3. **Embedded** – A software project with requiring the highest level of complexity, creativity, and experience requirement fall under this category. Such software requires a larger team size than the other two models and also the developers need to be sufficiently experienced and creative to develop such complex models.

3.4.1 Basic COCOMO Model

E=a*(KLOC)^b

D=c*€^d

Where E=Effort applied in person-months, D is the development time in chronological months and KLOC is estimated number of delivered lines of code for project expressed in per thousand. The coefficients a, b, c, d come from the given table considering our project to be organic.

Table 3.4.1 Coefficients for Basic COCOMO Model

SOFTWARE PROJECTS	**A**	**B**
Organic	2.4	1.05
Semi Detached	3.0	1.12
Embedded	3.6	1.20

Software Projects A B Organic 2.4 1.05 Semi Detached 3.0 1.12
Embedded 3.6 1.20 Table 3.4.1 Coefficients for Basic COCOMO model
The KLOC for our project are as follows:

E=2.4*(0.493)^1.05

=1.14 person-months

D=2.5*(1.14)^0.38

= 4 months

The number of people required to complete this project were 3.

3.5 SOFTWARE REQUIREMENTS SPECIFICATION

INDEX

5. Other Non-functional Requirements

5.1 Performance Requirements

5.2 Software Quality Attributes

5.2.1 Usability

5.2.2 Efficiency

5.2.3 Robustness

1. INTRODUCTION

1.1 Purpose

In our proposed solution we focus on developing a system that dynamically tracks movements in a given area using camera and triggering necessary actions, this solution implements object tracking in computer vision over a network of cameras generating geofences with occupancy map allowing the system to track movements of individuals, traffic and objects across any area with attention to features such as direction of motion, velocity, physical appearance etc. This would allow us to leverage the best of multivariate data.

1.2 Scope

In our proposed system, we developed a user interface that detects the malicious movements in the area via the surveillance network. It differentiates the legitimate movements from the malicious movements and aims in regulating unauthorized movement. It is used by security administrators to monitor the area.

2. OVERALL DESCRIPTION

2.1 Product Perspective

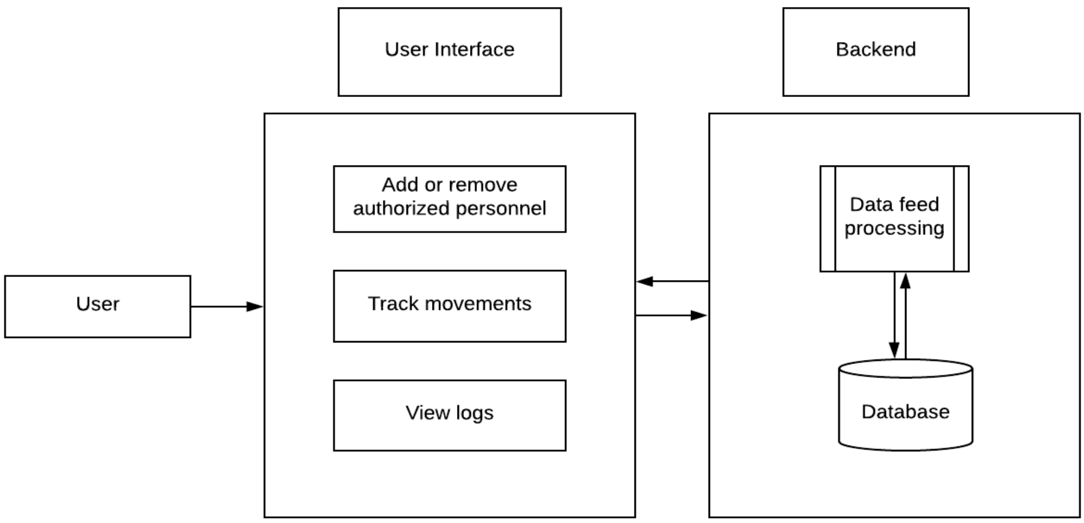


Fig. 3.5.2.1 Architecture Description

2.2 Product Functions

2.2.1 Adding or removing authorized personnel

The surveillance administrator is an authority that has exclusive access to the system as a whole and can add or remove authorized personnel from the surveillance system. This process is as simple as adding a few demographics from the user interface and feeding it to the system which then recognizes the person across the area of interest.

2.2.3 Tracking

The purpose of the system to recognize and identify people is coupled with tracking appropriate movements.

2.3 User Classes and Characteristics

A surveillance administrator is responsible for installing, maintaining, monitoring and upgrading any software or hardware required to efficiently run the system. A surveillance administrator is designated with the responsibility including maintaining security infrastructures with emphasis on surveillance. Responsibilities may vary between organizations, but on-site monitoring, software-network interactions as well as network integrity/resilience are the key areas of focus.

2.4 Operating Environment

- **Hardware:** Surveillance system network over the entire area with cameras supporting 60 fps video streaming.
- **Database**: MongoDB is a document database, which means it stores data in JSON-like documents. We believe this is the most natural way to think about data, and is much more expressive and powerful than the traditional row/column model.
- **Operating system**: Linux is a family of open-source Unix-like operating systems based on the Linux kernel, an operating system kernel

3. EXTERNAL INTERFACE REQUIREMENTS

3.1 User Interfaces

It provides the list of authorized personnel, frequency of motion, geofences and analytics.

3.2 Software Interfaces

- **Python 3.6**: Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built-in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together.

- **Flask**: Flask is a micro web framework written in Python. It is classified as a microframework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions.

- **Face recognition**: The 'face_recognition' library recognizes and manipulates faces from Python or from the command line with the world's simplest face recognition library. Built using dlib's state-of-the-art face recognition built with deep learning.

- **Database**: MongoDB is a document database, which means it stores data in JSON-like documents. We believe this is the most natural way to think about data, and is much more expressive and powerful than the traditional row/column model.

- **Hardware:** Surveillance system network over the entire area with cameras supporting 60 fps video streaming.

- **Operating system**: Linux is a family of open-source Unix-like operating systems based on the Linux kernel, an operating system kernel

4. SYSTEM FEATURES

4.1 Detection and recognition

4.1.1 Description and Priority

In the system, the library 'face_recognition' recognizes and manipulates faces using dlib's state-of-the-art face recognition built with deep learning. This feature involves performing detection of faces. It is a high priority function as it is the main functionality and purpose of the system to recognize the people and identify people in order to track appropriate movements.

4.1.2 Stimulus/Response Sequences

This feature involves performing detection by using the application. It is a high priority function as it is the main functionality and purpose of the system-to recognize the people and identify people in order to track appropriate movements.

4.1.3 Functional Requirements

Req-1: The surveillance system should have a clean video feed streaming at approximately 60fps.
Req-2: The system hosting the processing module should be powerful enough to run deep learning models.

4.2 Tracking

A tracking system, also known as a locating system, is used for the observing of persons or objects on the move and supplying a timely ordered sequence of location data for further processing. In our system, tracking is achieved by parsing the log records generated.

4.2.1 Description and Priority

The purpose of the system to recognize and identify people is coupled with tracking appropriate movements. This is achieved by creating a log for each detection made from the input video stream. Below is a representation of the data structure.

```
type
< Datetime, camera, person[list], location, ...>
```

For every detection made, the corresponding date-time, id of camera, id of the person (in-case of authorized users) etc are stored in a log.

4.2.2 Stimulus/Response Sequences

This feature involves performing detection by using the application. It is a high priority function as it is the main functionality and purpose of the system-to recognize the people and identify people in order to track appropriate movements.

5. OTHER NON-FUNCTIONAL REQUIREMENTS

5.1 Performance Requirements

The software system to able to fulfil its purpose i.e., detecting people and tracking movements with the best possible utilization of all necessary resources (time, storage, transmission channels, cameras etc).

5.2 Software Quality Attributes

5.2.1 Usability

Our application can be viewed as a tool for security admins to monitor any suspicious activity in the area. The application compares the results of the previous data and detects whether the subject is authorized.

5.2.2 Efficiency

The software system to able to fulfil its purpose i.e., detecting people and tracking movements with the best possible utilization of all necessary resources (time, storage, transmission channels, cameras etc.).

5.2.3 Robustness

Robustness reduces the impact of operational mistakes, erroneous input data, and hardware errors. Our user interface-based application is robust and the consequences of an error in its operation, in the input, in relation to the application, are inversely proportional to the probability of the occurrence of this error in the given application.

4. SYSTEM DESIGN

4.1 SYSTEM ARCHITECTURE

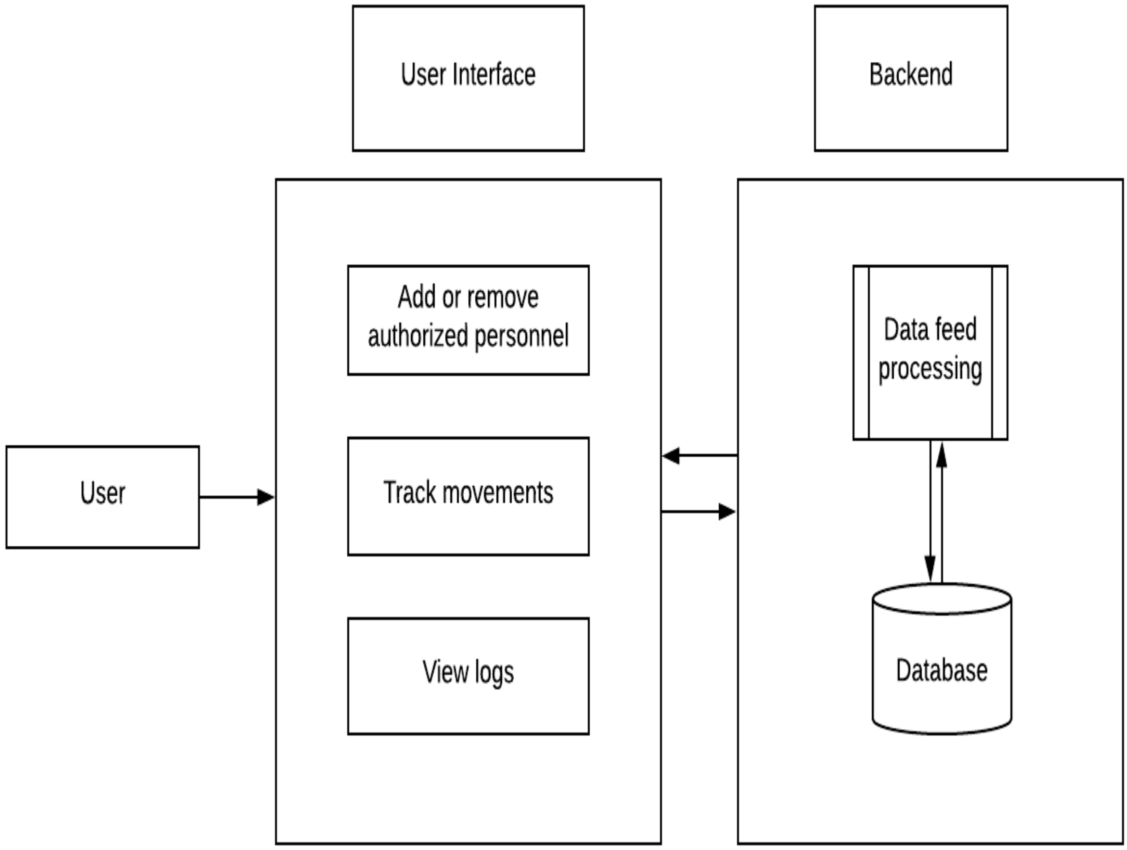


Fig. 4.1 System Architecture

The entire system is modelled into two main components:

- Backend or the intelligence component
- User interface

The intelligence component of the system deals with processing of the frames and creating the log. This process is computing resource intensive and accesses required data from the database.

The user interface is an admin portal that allows the ability to add or remove authorized users, view the logs and analytics. All the movements across the geofences can be checked upon.

4.2 UML DIAGRAM

4.2.1 Use Case Diagram

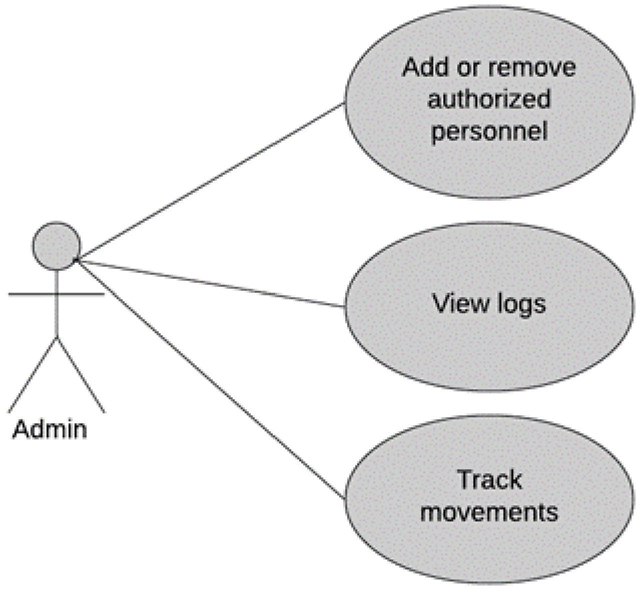


Fig. 4.2.1 Use Case Diagram

The admin has access to mainly 3 components. Each of this is responsible for an exclusive task.

- Add or remove authorized personnel: Through the portal, admin could add certain people as authorized. These people can then be tracked using a set of special ids. If an area is made accessible to authorized people only, any intrusion can be detected.

- View logs: This allows the admin to establish a ground for looking at the recent movements of a subject. This would be handy in order to detect erroneous movements.

- Track movements: In addition to the above, movements of a specific individual can be traced using the log generated.

5. IMPLEMENTATION

5.1 MODULES

List of Modules

- Facial detection and Log generation
- User Interface and geofencing
- Analytics

5.1.1 Facial detection and Log generation

In the system, the library 'face_recognition' recognizes and manipulates faces using dlib's state-of-the-art face recognition built with deep learning. The facial detection process is as follows:

Resize input frame of video to 1/4 size for faster face recognition processing.

- Using the input frame, a 2d array of bounding boxes of human faces is created using the CNN face detector. Using a GPU, this gives faster results since the GPU can process batches of images at once.
- From the faces extracted, we compare a list of face encodings against a candidate encoding to see if they match.
- If a match occurs, we get a Euclidean distance for each comparison face. The distance tells how similar the faces are.
- On finding familiar faces, we extract an id corresponding to the encoding and use that for further processing.
- The purpose of the system to recognize and identify people is coupled with tracking appropriate movements. This is achieved by creating a log for each detection made from the input video stream.
- Once a face is detected and identified, a new entry in the log is created using additional details.
- These entries in the log are created per frame per face detected. These are further used to track and generate analytics.

5.1.2 User Interface and Geofencing

The user interface is a dashboard which is accessible by the admin and has a range of options that are offered by the system.

- Add or remove authorized personnel: Through the portal, admin could add certain people as authorized. These people can then be tracked using a set of special ids. If an area is made accessible to authorized people only, any intrusion can be detected.
- View logs: This allows the admin to establish a ground for looking at the recent movements of a subject. This would be handy in order to detect erroneous movements.
- Track movements: In addition to the above, movements of a specific individual can be traced using the log generated.

A geofence is a virtual perimeter for a real-world geographic area. It could be dynamically generated—as in a radius around a point location, or a geo-fence can be a predefined set of boundaries.

5.1.3 Analytics

This system can execute various analytics that are cumbersome otherwise. These include performing a real time head count, identifying common movement patterns, heat maps, generating statistics w.r.t time and location per head, detailed analysis of traffic flow etc.

6. TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and /or a finished product.

6.1 Types of Testing

6.1.1 Unit Testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests endure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

Test Cases:
A test case is a set of conditions or variables under which a tester will determine whether an application, software system or one of its features is working as it was originally established for it to do.

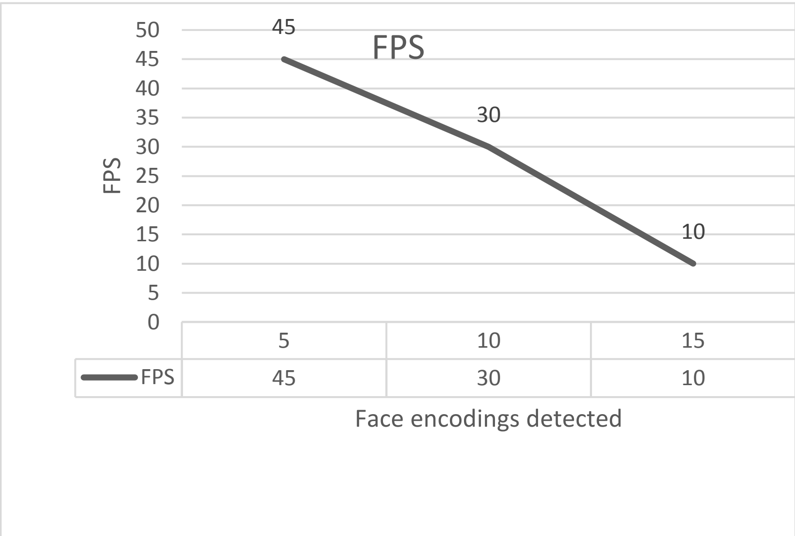


Fig 6.1.1 Unit testing on FPS

7. SCREENSHOTS

7.1 LANDING PAGE

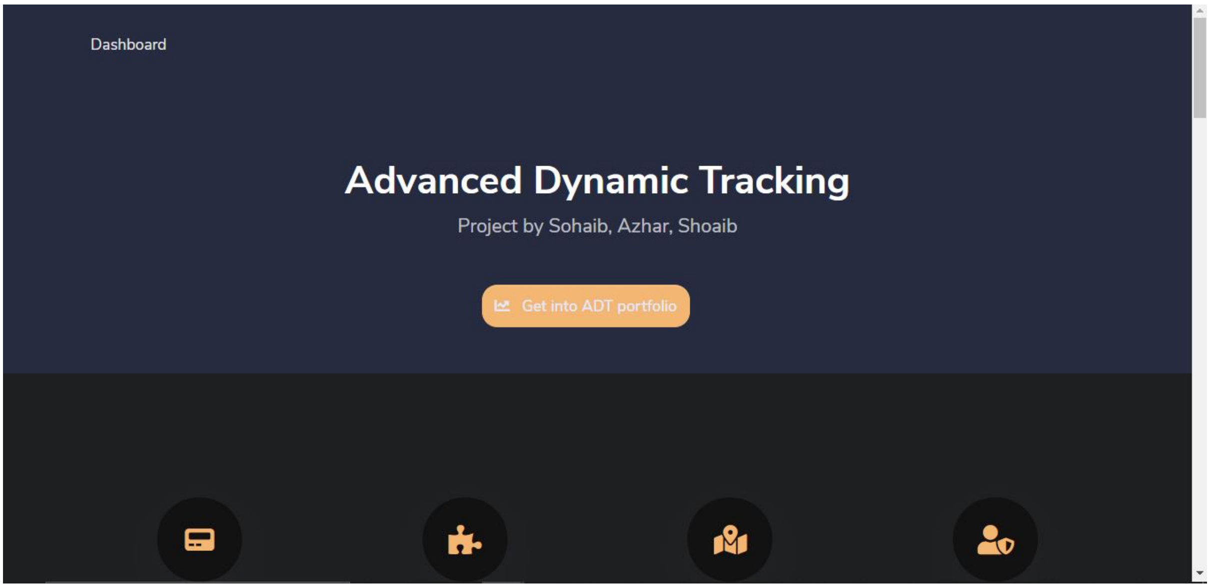


Fig. 7.1 Landing Page

7.2 DASHBOARD

Dashboard contains all the options for the admin to utilize various system features

Fig. 7.2 Dashboard

7.3 LOG RECORDS

The log records are used to trace a particular subject in the area of observation.

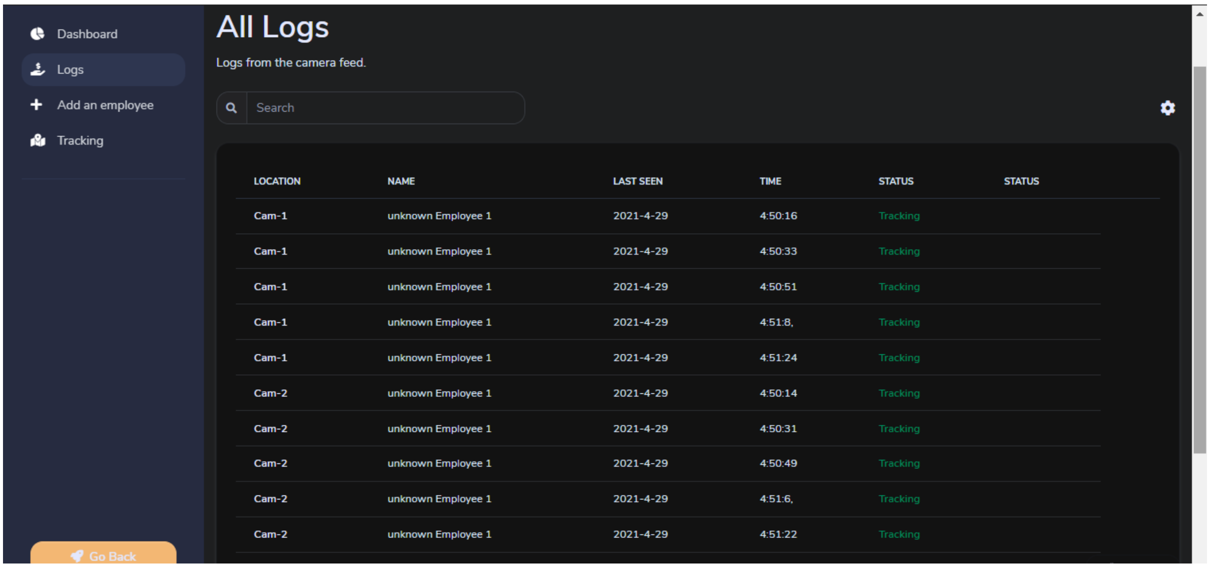


Fig. 7.3 Log Records

7.4 GEOFENCING AND TRACKING

The tracking section shows the live status of cameras and the availability of various people or a subject in a geofence. The green areas show where the subject, in this case 'Agent-1' has visited.

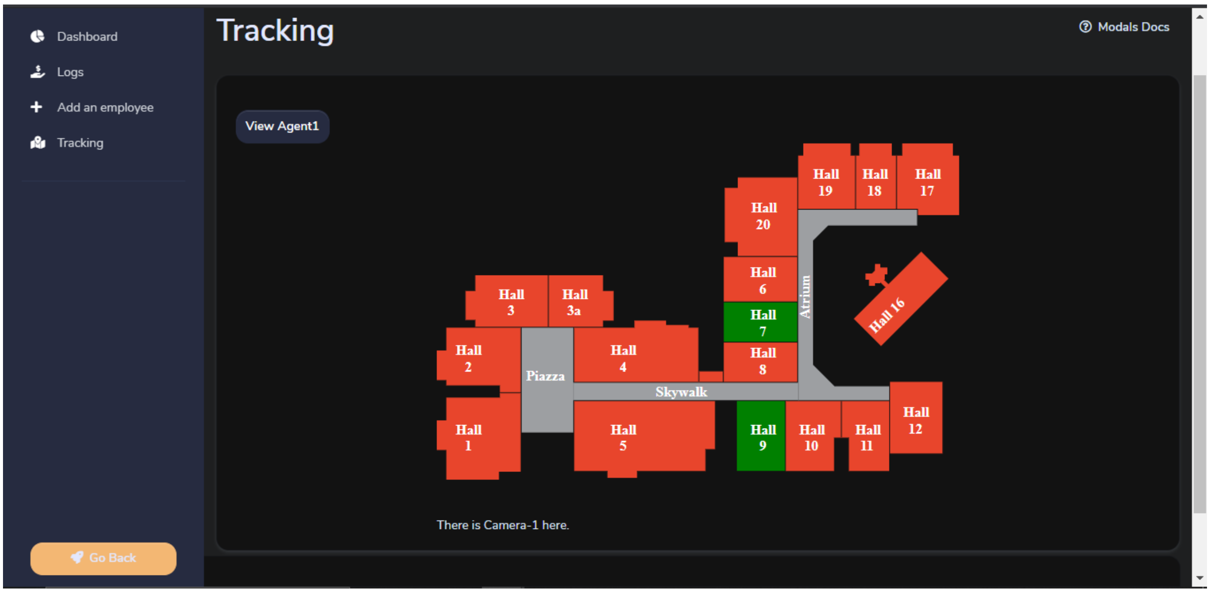


Fig. 7.4 Geofencing and tracking

7.5 LOG GENERATION

The log records are used to trace a particular subject in the area of observation and are generated per face encoding detected in every frame.

```
date 2021-4-29
time 4:50:16
people ['unknown', 'Employee 1']
location Cam-1
startdate 2021-4-29
enddate None
count 3


date 2021-4-29
time 4:50:33
people ['unknown', 'Employee 1']
location Cam-1
startdate 2021-4-29
enddate None
count 3


date 2021-4-29
time 4:50:51
people ['unknown', 'Employee 1']
location Cam-1
startdate 2021-4-29
enddate None
count 3


date 2021-4-29
time 4:51:8,
people ['unknown', 'Employee 1']
location Cam-1
startdate 2021-4-29
enddate None
count 3
```

Fig. 7.5 Log file

7.6 FPS RESULTS

```
(venv2.6) C:\Users\techs\OneDrive\Desktop\100AI\100AI>python run.py
[INFO] sampling frames from webcam...
[INFO] elasped time: 14.57
[INFO] approx. FPS: 30.06
2

(venv2.6) C:\Users\techs\OneDrive\Desktop\100AI\100AI>
```

Fig 7.6 FPS results

8. CONCLUSION AND FUTURE SCOPE

We have implemented a system that dynamically tracks movements in a given area using cameras and triggers necessary actions. This project implements object tracking in computer vision over a network of cameras generating geofences with occupancy map allowing the system to track movements of individuals, traffic and objects across any area with attention to facial features.

Apart from being a strong business model with implementations in retail, commerce, defense, security and surveillance, this would be a leap in technology-surveillance domain

9. REFERENCES

[1] Szeliski, Richard. *Computer vision: algorithms and applications*. Springer Science & Business Media, 2010.

[2] Khan, Salman, Hossein Rahmani, Syed Afaq Ali Shah, and Mohammed Bennamoun. "A guide to convolutional neural networks for computer vision." *Synthesis Lectures on Computer Vision* 8, no. 1 (2018): 1-207.

[3] Kalchbrenner, Nal, Edward Grefenstette, and Phil Blunsom. "A convolutional neural network for modelling sentences." *arXiv preprint arXiv:1404.2188* (2014).

[4] Tai, Ying, Jian Yang, and Xiaoming Liu. "Image super-resolution via deep recursive residual network." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3147-3155. 2017.

[5] Cilimkovic, Mirza. "Neural networks and back propagation algorithm." *Institute of Technology Blanchardstown, Blanchardstown Road North Dublin* 15 (2015): 1-12.

[6] Karahan, Samil, Merve Kilinc Yildirum, Kadir Kirtac, Ferhat Sukru Rende, Gultekin Butun, and Hazim Kemal Ekenel. "How image degradations affect deep CNN-based face recognition?" In *2016 international conference of the biometrics special interest group (BIOSIG)*, pp. 1-5. IEEE, 2016.

[7] Kruegle, Herman. *CCTV Surveillance: Video practices and technology*. Elsevier, 2011.

[8] Costin, Andrei. "Security of cctv and video surveillance systems: Threats, vulnerabilities, attacks, and mitigations." In *Proceedings of the 6th international workshop on trustworthy embedded devices*, pp. 45-54. 2016.