JUNE 2020 Ph.D. in Electrical and Electronics Engineering SUDAD J. ASHAJ

REPUBLIC OF TURKEY

GAZIANTEP UNIVERSITY

GRADUATE SCHOOL OF NATURAL & APPLIED SCIENCES

ENERGY MANAGEMENT SYSTEM IN REAL TIME BY IMAGE PROCESSING AND DEEP LEARNING

Ph.D. THESIS

IN

ELECTRICAL AND ELECTRONICS ENGINEERING

BY

SUDAD J. ASHAJ

JUNE 2020

ENERGY MANAGEMENT SYSTEM IN REAL TIME BY IMAGE PROCESSING AND DEEP LEARNING

Ph.D. Thesis

in

Electrical and Electronics Engineering

Gaziantep University

Supervisor

Prof. Dr. Ergun ERÇELEBİ

by

SUDAD J. ASHAJ

June 2020

ABSTRACT

ENERGY MANAGEMENT SYSTEM IN REAL TIME BY IMAGE PROCESSING AND DEEP LEARNING

ASHAJ, SUDAD J

Ph.D. in Electrical and Electronics Engineering

Supervisor: Prof. Dr. Ergun ERÇELEBI

June 2020

92 pages

As complexity and the size of electrical systems perpetually enlarges, heavy loading will require extra systems to manage electrical power and regulate energy consumption in the demand and customers side. To overcome these problems, in this thesis, a smart power management system has been developed. The system has been realized using minicomputer, sensors, image processing and deep learning algorithms. The developed power management system can handle a lot of electrical management crucial aspects such as reducing power consumption to lower possible limits, controlling ambient temperature, turn on/off some devices depending on human activities. The system we developed in the thesis and named as smart building management has the advantage of making the decisions such as automatic control of the air conditioning system to adjust the ambient temperature according to the comfort of the user, lighting and turning the television on and off depending on the human mobility in the interior. In this study, a camera compatible with minicomputer as Raspberry Pi has been utilized for taking image or recording real-time streaming video for detection the existence of human and his activity. The deep learning based on Convolutional Neural Network (CNN) structure has been utilized as an efficient technique in order to classify the objects properly.

Furthermore, the saliency object detection algorithm in addition to image processing tools has been exploited to extract the discriminative features and detect the valuable objects so that speed up both taking the true decision and network learning. The Python computer program language has been utilized because of having huge library functions and its compatibility in programming the real-time embedded system. Experimental results show that the proposed system could be administered electrical consumption with smart and accurate behaviour. Moreover, the recognition algorithm has rapidly produced good results in distinguishing human from other objects, where the accuracy of correct detection can reach to 97.9% and convergence time reach to 0.9 seconds.

Key words: Power Management, Power Saving, Smart Power System, Raspberry PI 3, Deep Learning, CNN, Image Processing

ÖZET

GÖRÜNTÜ İŞLEME VE DERİN ÖĞRENME İLE GERÇEK ZAMANLI ENERJİ YÖNETİM SİSTEMİ

ASHAJ, SUDAD J

Doktora Tezi, Elektrik ve Elektronik Mühendisliği

Danışman: Prof. Dr. ERGUN ERÇELEBİ

Haziran 2020

92 Sayfa

Karmaşıklık ve elektrik sistemlerinin boyutu sürekli büyüdükçe, ağır yükleme, elektrik gücünü yönetmek ve talep ve müşteri tarafında enerji tüketimini düzenlemek için ekstra sistemler gerektirecektir. Bu problemlerin üstesinden gelmek için bu tezde akıllı bir güç yönetim sistemi geliştirilmiştir. Sistem mini bilgisayar, sensörler, görüntü işleme ve derin öğrenme algoritmaları kullanılarak gerçekleştirilmiştir. Geliştirilmiş güç yönetim sistemi, güç tüketimini mümkün olan daha düşük sınırlara düşürmek, ortam sıcaklığını kontrol etmek, insan aktivitelerine bağlı olarak bazı cihazları açmak / kapatmak gibi birçok elektrik yönetiminin önemli yönlerini ele alabilir. Tezde geliştirdiğimiz ve akıllı bina yönetimi diye isimlendirdiğimiz sistem kullanıcı konforuna göre ortam sıcaklığını ayarlamak için iklimlendirme sisteminin otomatik olarak kontrol edilmesi, iç mekanda insan hareketliliğine bağlı olarak aydınlatma ve televizyonun açılıp kapanması gibi kararları gerçek zamanlı olarak yapabilme avantajına sahiptir. Bu çalışmada, Raspberry Pi olarak mini bilgisayar ile uyumlu bir kamera, insanın varlığını ve aktivitesini tespit etmek için görüntü veya gerçek zamanlı akış videosu kaydetmek için kullanılmıştır. Evrişimsel Sinir Ağları yapısına dayalı derin öğrenme, nesneleri düzgün bir şekilde sınıflandırmak için etkili bir yaklaşım olarak kullanılmıştır. Ayrıca, görüntü işleme araçlarına ek olarak göze çarpan nesne algılama algoritmasından, ayrımcı özellikleri ayıklamak ve değerli nesneleri tespit etmek için yararlanıldı, böylece hem gerçek karar hem de ağ öğrenimini hızlandırıldı. Python bilgisayar program dili, büyük kütüphane fonksiyonuna sahip olması ve gerçek zamanlı gömülü sistemin programlanmasına uyumluluğundan dolayı kullanılmıştır. Elde edilen sonuçta göre, sistemin elektrik tüketimini akıllı ve doğru yönetebileceğini doğruladı. Ayrıca, tanıma algoritması, algılama doğruluğunun 97,9% olması ve yakınsama süresinin 0,9 saniyeye ulaşması+ insan formundaki diğer nesneleri ayırmada hızlı bir şekilde iyi sonuçlar üretebilir.

Anahtar Kelimeler: Güç Yönetimi, Akıllı Güç Sistemi, Derin Öğrenme, Konvolüsyon Sinir Ağı, Görüntü İşleme.

“Dedicated to My Family and Beloved Parents”

ACKNOWLEDGEMENTS

In the name of Allah, the most gracious, the most merciful. First, I would like to thank Allah who granted me health and strength to overcome life's difficulties and for his guidance to complete this PhD thesis.

I would like to express my gratefulness to my supervisor, **Prof. Dr. Ergun ERÇELEBİ** for his guidance, patience, kindness, and encouragement throughout this research.
I am indebted to PhD Ahmed Khalid Ali for his support in correcting and ensure the text of the thesis and his assistance throughout my dissertation

I would like to express my thanks to the staff members of the Department of Electrical and Electronics Engineering at the University of Gaziantep and my thanks to all other friends for their helping me in preparing this research.

Finally, my grateful thanks to all my family for providing support to me in my study.

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

$\boldsymbol{\sigma}$	The standard (unit) softmax function
$\mathbf{x}$	Input vector
$\mathbf{p}$	Probability
$\boldsymbol{W_{ij}}$	Weight between the surround patch and center patch
$\boldsymbol{D_{ij}}$	The distance between i and j patch
$\boldsymbol{S_g}$	Global saliency
$\boldsymbol{S_l}$	Local saliency
$\boldsymbol{S_{lg}}$	Final saliency maps
$\boldsymbol{E}$	Average Energy
$\boldsymbol{NL}$	Network Lifetime
$\boldsymbol{T}$	Average Throughput
$\boldsymbol{D}$	Average Delay
$\boldsymbol{I}$	Input image
$\mathbf{o}$	An integration structure or fusion function
$\boldsymbol{n}$	The number of input features in pattern
$\boldsymbol{m}$	The number of output classes
$\boldsymbol{h_j}$	The output of hidden node j
$\boldsymbol{w_{jk}}$	The weight connected between input node j and hidden node k
$\boldsymbol{b_k}$	The bias of hidden node k
$\boldsymbol{Y_k}$	The output of hidden node k
$\boldsymbol{T_k}$	The desired output at node k
$\boldsymbol{\delta_k}$	The back- propagation error at node k
$\boldsymbol{f}$	The activation function
$\boldsymbol{f'}$	The derivative of the activation function
$\boldsymbol{\eta}$	The learning rate
$\boldsymbol{\alpha}$	The momentum coefficient
$\Delta\boldsymbol{w_{jk}}$	Weight adjustment between input node j and hidden node k

LIST OF ABBREVIATIONS

ACO	Ant Colony Optimization
AI	Artificial Intelligence
ANFIS	Adaptive Neural Fuzzy Inference System
ANN	Artificial Neural Network
BS	Base Station
BPNN	Back-Propagation Neural Network
CNN	Convolutional Neural Network
COCO	Common Objects in Context
CONV	Convolutional layer
CPU	Central Processing Unit
DACR	Dynamic power Allocation and Chain based Routing
E2HRC	Energy-Efficient Heterogeneous Ring Clustering Routing Protocol
ED	Euclidean Distance
EMS	Energy Management Systems
EV	Electrical Vehicle
GPIO	General Purpose Input Output
FC	Fully Connected Layer
FP	False Positive
FN	False Negative
GPU	Graphics Processing Units
ICT	Information and Communication Technology
IoT	Internet of Things
mAP	Mean Average Precision
MCU	Microcontroller Units

MHD	Manhattan Distance
MLP	Multi-Layer Perceptron
NILM	Non-Intrusive Load Monitoring
QHCR	QoS-aware and heterogeneously clustered routing
QoS	Quality of Service
PIR	Passive Infrared Sensor
TP	True Positive
TN	True Negative
POOL	Pool max Layer
PV	Photovoltaic
PMS	Power Management System
ReLU	Rectified Linear Unit
SGD	Stochastic Gradient Descent
SVM	Support Vector Machine
VGG	Visual Geometry Group
VOC	Visual Object Classes
WSN	Wireless Sensor Network

CHAPTER ONE

INTRODUCTION

1.1 Motivation

The usage of electricity has always been a concern for business and home. As electric costs, ease of use and automated controlling has been rising; there is an additional need for a smart system that optimize these aspects. The smart system controls the power system to manage the power consumption and to use electrical devices most proficiently. One of the practical ways that can do these systems is by using a smart energy management system. It can help to control the overall process in an automated way with the ability to the user to monitor the most power distribution and total cost as well as the ability to change some settings as he desired. Moreover, the effective smart power system should be taken in consideration many points such as controlling the electrical devices power consumption accurately and offering fully monitoring to electric devices by used some expert systems like artificial neural network, fuzzy logic, Genetic algorithms etc… [1]. The knowledge gained from the deployment of energy efficiency projects is driving a transition from traditional tactical practices to more comprehensive strategic energy management practices. Like many modern management practices, a strategic energy management approach highlights the need for an information system to set goals, track performance, and communicate results.

In the last decades, plenty of researches in the field of power management systems are presented [2-8], most of these researches are focused on utilizing low-cost components such as cheap microcontroller units and single-board computers along with low-cost sensors. Through combining these popular parts, it becomes straightforward to design and implement a general-purpose monitoring and control systems. Some of these studies are focused on using simple sensors that can detect motion

(motion sensor) or light (light sensor), and then switch on a device such as lighting when it detects motion or light variation. These methods have an advantage of simplicity and low in cost. Goswami et al. (2009) [3] presented an attractive design for a power management system that is based on traditional microcontroller and some sensors to assemble climate data inside the building. The experimental results had showed that the system is accurate in assembling the climate data and hence in achieving an efficient power management system.

The accuracy is calculated via how intently the sensors can gauge the genuine or real-world parameters values. The more accurate sensor is, the better it will perform. Samadi et al. (2010) [4] proposed a smart power infrastructure that equipped with energy consumption controller; their work is focused on the interactions between energy provider and smart meters and they provided an algorithm to deal with the energy consumption. In the previous cases, the useable techniques are not employed because they depend on physical changes, for example, a motion sensor detects any movement in its sensing field but it cannot detect if that human, cars, or other things. As a result, the switch can be on even not need to be switch on when anything still moving such as animal. In another hand, the light sensors detect variation of light and it either powers on the light in case were the light is low or powers off the light when light becomes high. Kleiminger et al. (2013) [5] pointed out that power management system in buildings and households can be achieved by using motion sensors and cameras, and they proposed a method that combined motion sensors with energy consumption, passive infrared sensors (PIR sensors) have been installed in homes together with smart meters and plugs to measured power consumption. They are employing a machine learning algorithm to recognize whether anyone is at home or not. They also established an Android application to gather data and store results; it was installed on a tablet in home. This means that users had to explicitly change states from being home or away through the tablet. The reason for doing this was because they needed to label their gathered data to know whether users were home or away, since they used a supervised machine learning approach. However, one cannot be certain that the users always remembered to change states whenever they left their homes or arrived at homes. Their users also admitted to forgetting it sometimes. They only focused on electrical power consumption, which means heating is not considered.

To overcome the previous challenges, a new study which is focused on exploiting image processing techniques with imaging tools such as a camera to sense varieties like motion and light along and utilizing artificial intelligence system for object detection and recognition such as Support Vector Machine (SVM), Artificial Neural Network (ANN), etc. [5]. SVM is essentially used for binary classification and can be extended to multi classification, SVM needs low computational requirements than ANN, and it has good accuracy. ANN is more popular and has many types and applications, ANN can estimate any linear or non-linear critical job in a wide scope of applications and it has been developed to serve the power management systems in recent years. Among tens of ANN types, Convolutional Neural Network, which is commonly referred to by (CNN), is another and precise kind of artificial neural network that can be used efficiently with image processing and recognition. The CNN was published in 2012 by Krizhevsky et al. [6] (also referred as AlexNet), it gives a big improvement in the field of picture order and object localization that improved the classification performance and it applicable to a wide range of visualization tasks [7]. These techniques, however, is dependent on the computational requirement and typically needs a moderate processing specification to work smoothly which is not available in most microcontroller units (MCU) and single board computers and in most cases required a PC based system to achieve that purpose [8]. Hence, to use the ANN with low processing units most microcontroller unit (MCU) or single board computers there is a need to design controller programs that can make the overall processing operation fast enough to work. This work is focused on used CNN to detect some important object that can be served in power management application such as controls lighting, TVs, computers, etc. Since the theory of object recognition accuracy is highly dependable on processing an image (enhancement and segmentation) to extract important features, in such way that with more details of image and efficient segment the objects inside the image and isolated it, high accuracy of recognition can be obtained. In the case where the image has many details and objects and when there is a need to process multi images in same time such as when need to recognized multiple objects from video streaming that may stream 25 fps (frame per second) that means the system should processing 25 images per second to recognize the desired objects. This heavy process needs to use a high computational requirement to recognize the object with high correctness in real time. However, the use of CNN with microcontrollers or even with single board computer (such as Raspberry Pi) to recognize objects in real time is a difficult task, because the low computational requirement that make the process slow and not practical.

The energy efficiency for Wireless sensor networks (WSNs) is extensively research problem by considering the real time applications like building or home automations. At the point when any sensor hub lost his battery control, this will naturally dispose of from the network and may lead to entire network failure as well. The fault tolerance approaches then required to mitigate network failures problem [12] [13].

Other side, the cluster-based protocols suffering from the overhead identified with cluster arrangement, cluster head determination, and information transmission. Recently more research works provided details regarding streamlined bunching approach for essentialness efficiency in WSNs. The unnecessary re-clustering at each interval may leads to burden on sensor nodes that lead to energy consumption.

One of the main purposes of smart power management system is to reduce energy consumption. To achieve this goal, smart controls must be implemented in a smart home. Moreover, useless power consumption occurs during day time and when the human being is not present in the room, so through this project we are overcoming this problem. This smart home implementation can contribute to major reductions of energy use in the buildings.

1.2 Background

In the last decades, there are plenty of research has been achieved in the field of smart home PMSs such as [12-20]. But there are few works focused on developing smart PMSs having crucial advantages in the design cost and high flexibility which is meeting with the customer's requirements. From this end, this section presents an overview of the fundamental concepts for smart energy management systems using the different approaches.

In [12], a smart self-learning home management system was developed with real-time operation. The smart home energy management system presented here is based on the Internet of Things concept. The presented system can improve management energy consumption in both the demand-side management system and supply-side management system. The authors claimed that the proposed smart integrated system has many advantages; for examples, price forecasting, price clustering, and power alert system to enhance its function. Further to that, a self-learning home management system also demonstrates its ability to customize the model for different types of situations compared to conventional smart home models.

In [13], the control of the executive learning has been proposed by the author, what ones called smart fitting has been replaced by the sensors which can as per nature dealt "on and off" consequently. What more can be expected these sensors, at the diverse conditions can be changed. Moreover, these deep models are brought out along with the ground care, which shows the option of "mist or haze" registration.

In [14], The "Non-Intrusive Load Monitoring (NILM) algorithm" presented for each store, by actualizing the "sub-metering system" in order to evaluate own improvement dependent on receptacle pressing calculations and criticism frameworks constrained by the AI calculation to finish up with a vitality proficient keen home and savvy lattices.

According to [15], designed optical stream criticism convolutional neural system according to the video stream to screen the home for energy efficiency. Their model is employed rule-based channels before a data convolutional layer and the recorded optical stream for managing the optical progression of assortment. Distinguishing human stance is a key factor while fall events take after a falling stance. By sequencing the housings of action, it is possible to see a fall.

In [16], "Internet of Things (IoT)" based home vitality executives the framework proposed with lightweight photovoltaic (PV) framework over powerful home territory systems, which bolsters the development of home scalable, reusable, and interoperable energy management method. The proposed methods with lightweight photovoltaic (PV) framework over powerful home territory systems.

In [18], based on energy saving in marketing concept, on the other hand, a variation of electricity market operation and prices with respect to present times is still a major concern in smart home management systems. The presented hardware design of smart home energy management system could be given as options to consumers. That can easily achieve real-time by using artificial intelligence and machine learning algorithm. The proposed design of smart home energy management system involves a new technique which is based on the price-responsive control strategy for residential home loads, such as electric water heater (EWH), heating, ventilation, and air conditioning (HVAC), electrical vehicle (EV), dishwasher, washing machine, and dryer.

According to [19], they have presented a demand response predictor model by using an adaptive neural fuzzy inference system (ANFIS). A predictor model is integrated to predict, based on the user's lifestyle and different social/ environmental parts, the potential schedules for appliance run times. An aggregator is utilised to accumulate predicted demand from domestic users.

1.3 Contributions and Significance of the Work

This thesis is expected to structure a used artificial intelligence based on a brilliant PMS with a minimal effort single board computer. The proposed framework manages the intensity of building/house by dissected a few angles that can serve this goal. As referenced previously, the "energy management system" could be utilized to control devices like lighting frameworks. It likewise gives metering and checking capacities, which permits them to settle on choices with respect to energy consumption over their destinations, and it incorporates arranging and energy-related production and utilization units. One of the primary motivations behind savvy homes is to decrease energy utilization. To accomplish this goal, brilliant controls must be executed in a savvy home. Besides, pointless force utilization happens during day time and furthermore when the person is absent in the room, so through this understanding, we are defeating this problem. This savvy home usage can add to significant decreases in the consumption of power within the buildings.

To address such challenges, and also in this work, a novel routing algorithm called Dynamic power Allocation and Chain based Routing (DACR) is proposed, it is based on two challenges included: (1) Proposed the simple but effective dynamic power allocation algorithm according to sensor nodes connectivity parameter prior to data transmission to relegate the transmission control according to the necessity of the sensor node. (2) The chain-based approach for path formation and data transmission in which the chain is formed using the Ant Colony Optimization (ACO) algorithm.

1.4 Aim of the Study

This thesis is intended to design a smart power management system based on utilizing artificial intelligence with low-cost single board computer. The proposed system is able to manage power of building/house by the analysis of the few features for instance temperature to control, consumptions of the electricity on order to lessen the consumption of power to lower the limits as much as possible as well as the ability of human to control the on/off of the electronic devices (TV) and the lightening. Moreover, how to allocate and gather information from many sensors is the second proposed system, i.e., collecting data from sensors rapidly with low power requirement and attempting to address and decrease network parameters and complexity.

1.5 Thesis Outlines

This thesis is organized in four chapters besides this chapter and ends with references; each chapter is described as follows:

Chapter Two: covers the background required for PMS. This includes an overview of the power management embedded system with artificial intelligence techniques especially the artificial and convolutional neural networks (ANN and CNN) and the basic algorithms aspects that related to the classification purpose.

Chapter Three: describes the design of the PMS with all hardware and software used in it.

Chapter Four: explains the results and summarizes the main consequences of the work presented in this work.

Chapter Five: presents the conclusions, and recommendations for future work.

CHAPTER TWO

THEORETICAL CONCEPT OF POWER MANGMENT SYSTEM AND DEEP LEARNING

2.1 Power Management Systems (PMS)

PMS have been in presence in the energy sector for many years. The key features of such units are to control, monitor and improve the use and flow of electrical energy. Generally, PMS have massive uses in many electrical applications such as power generation, electrical transmission, distribution systems and optimized power consumption for home and building. The implementation of PMS starts with as low as the usage of single energy reliable devices up to the use of considerable devices and programmable tools for example, the Supervisory Control and Data Acquisition (SCADA) for energy management.

2.1.1 Smart power management system based on artificial intelligence

With the development of sensors technologies and computer sciences, intelligent and remote-control systems have been developed rapidly. The principle operation of these devices is first collecting data from the environment, second analyzing these collected data and then deciding or performing the desired commands based on the settings described from the user [2]. Smart or intelligence system is the scheme that has the capability to learn and take vital actions or make decisions in autonomous way. Smart system in household and buildings can monitor a wide range of devices and applications such as; lighting, air conditioning, heating, home security devices, ventilation, entertainment, health care applications, etc. [5]. These systems can also be considered as part of PMS in home/building, which can gather information from sensors and then control the power distribution automatically. The smart PMS is a sensor-based system that can also minimize waste, and make the environment comfortable [6].

Therefore, sensors are required to monitor and gather requested data. Typically, motion sensors are utilized to identify real activities of people [7]. In this consideration, MIT and TIAX constructed a laboratory to analyze human living behaviors and activities continuously to understand human activities during a whole day [8]. Recognizing human activities are divided into two categorizations: first working with sensors on human body to recognize recurrent motions like hiking, running. The second is to use sensors on objects in areas in order to recognize complex human activities and other moving objects [9]. A smart system can serve these considerations by recognize human activities [10]. These smart systems can be learned via the behaviors, or make inferences regarding the preferences of the people and after that take automated actions and decisions. One example is when a system is learning both cooking time duration and human activity. It could switch on/off the oven according to human activity. The learning procedure or mechanism is probably quite challenging and the accuracy is highly significant since it is directly associated with action and decision making. There are several approaches presented for this purpose. The widespread use and techniques of the Artificial Neural Network (ANN) provides opportunities for both learning process and activity recognition in a smart system. Even so, there is no explicit solution for the activity recognition. Other solutions included are; selection strategy and optimization.

2.1.2 PMS challenges

Accomplishing a smart PMS is not fully easy due to many challenges that facing this technology. These challenges can be summarized as in following:

1. **Cost:** The PMS cost is expensive in terms of the installation costs and device prices. Customers aren't ready to spend in systems whose incomes hardly satisfy their investment.

2. **No Standards for PMS:** There isn't any way for the design and implementation of these systems as each seller presents its own system with an exclusive control and design strategies.

3. **Low Consumer Awareness:** Customers are unaware about the usefulness of the EMS. Often consumers are confused concerning several PMS solutions available in the market.

4. **Choice of Information and Communication Technology (ICT):** It is an enabling technology to the excellent implementation of EMS. Some customers are concerned about the health issues of the wireless signals penetration that are a part of EMS.

5. **Designing System Intelligence:** It is hard to design a system that satisfies several levels of consumers' knowledge regarding EMS.

2.2 Artificial Intelligence and Pattern recognition

The purpose of Artificial Intelligence (AI) is to develop computerized algorithms that can provide same activities which are performed by human intelligence. Several fields of AI applications are automated problem-solving methods for knowledge engineering and knowledge representation for pattern recognition and machine vision, for automatic programming, artificial learning, etc. Obviously, the limitations of AI are not totally well described, and are still improving with time. AI approaches are not totally disconnected from other basic computational techniques, for example, data analysis. There are a large variety of applications that AI can employ for example: data fusion, dimensionality reduction, neural networks, supervised and unsupervised automated classification, object recognition, expert systems, texture and fractals, image understanding, learning, image comparison, fuzzy logic, and pattern recognition [21].

Pattern recognition is the probability of finding arrangements of data or features that have generally yield details regarding a data set or presented system. In a practical context, a pattern can be repeating sequences of data over time that can be used to predict trends, specific features settings in images that recognize objects. Most pattern recognition approaches involve artificial neural network (ANN), SVM, decision tree (DT), and also other rule-based classification systems. The ANN classification method is a supervised practical methodology with a lot of accomplishments in many classification tasks. It can be used to make complicated relations between inputs/outputs or often to find patterns. However, its classification efficiency and accuracy are generally a challenge. Different strategies have been achieved in order to enhance recognition accuracy. In this work, we shall discuss the accuracy gain via the utilization of deep learning with saliency method that can improve the detection and recognition of an object after isolating it from its environment.

2.2.1 Machine learning and deep learning

Machine learning is a subfield of AI that is significantly important, and is frequently used out in the industry to resolve several tasks. Even so, AI is not a new term in computer science, all of it started once “Alan Turing” suggested the concern "Can machines think?" [9]. Focusing on AI has been shifted around several areas. Provided the enormous amounts of data in existence nowadays, it is no wonder that the data-driven procedure of machine learning is so preferred. Thus, what represents learning for a machine? Mitchell [10] describes learning as follows: "A computer program is suggested to learn from experience with regard to some tasks class of task T and abilities measure probability P, if its performance at tasks in T, as measuring by P, enhances with experience Mitchell [10]. To put this into framework, the classical spam filter can be used as an example. It can be measured by the experience of the data set employed for training, and the efficiency can be measured as a ratio of the correctly classified data. Other favorite areas of usage are social networks, and recommender systems which all utilize machine learning solutions. For example, Facebook does suggest people one might know based one’s activity on the site. Deep Learning is a branch of Artificial Intelligence [22] which focuses on systems that can learn data representations without the need for handcrafted feature extraction methods. Deep learning is a new and promising area in machine learning to solve problems of artificial intelligence. The well-known state-of-art fields which have been proven to be successful with the deep learning approaches are computer vision, signal processing, natural language processing and speech recognition [23] [24].

2.2.2 Artificial neural network (ANN)

The name of neural networks comes from the ability of these networks to simulate human brain in solving problem. The sophisticated recognition could be achieved by using suitable type of network of neurons. Usually, the feed forward neural networks (FFNNs) are commonly adopted that have no feedback with simply present inputs and outputs [25]. The basic unit of the neural network is the neuron; it is a multi-inputs and multi-outputs function. The different inputs to the network can be represented by a mathematical symbol X_M and every connection has a weigh value called W_{MN} when input values multiply the linking weights, it is usually called the weighted inputs. The neuron output is a threshold of these weighted inputs. Figure 2.1 shows the basic artificial neuron function.

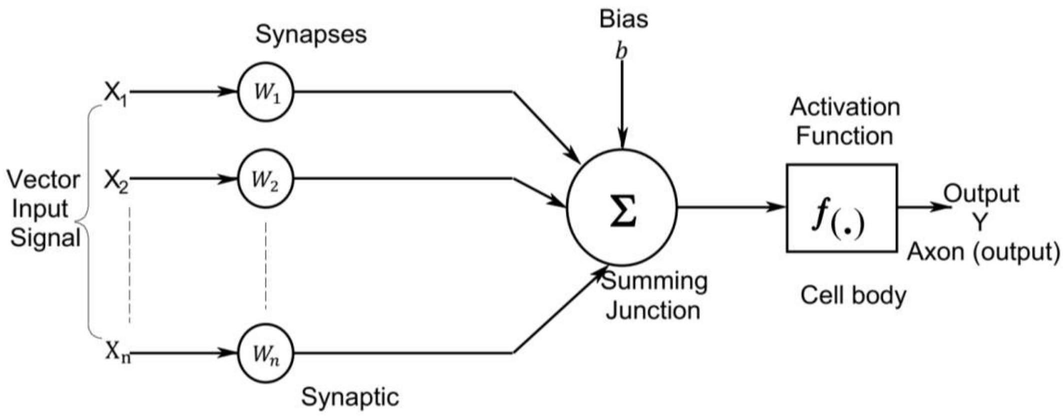


Figure 2.1 Basic artificial neuron

2.2.2.1 Activation function

In ANNs, the input data is transferred along the layers and is converted into output data that can be multi-class labels in a classification problem by firing rate of the neurons. Thus, the firing rate is modeled with activation function f. This activation function f can be represented as a linear or a non-linear function. In general, the non-linear activation functions are more successful than linear functions in terms of learning the correlation between variety of input data and output. In figure 2.2, there are activation functions which commonly used in both ANNs and CNNs. One of the most popular non-linear activations is the Sigmoid function. When learning and predicting a probability value as an output, sigmoid function can be used since it squashes output to the range of [0,1]. Moreover, Hyperbolic tangent activation function ($tanh$) is another activation which is very similar to sigmoid. However, unlike sigmoid, $tanh$ squashes the output within [-1, 1] range and it is zero-centered as well. Some activation functions such as sigmoid and tanh kill the gradients and make learning process difficult in the complex task such as multi-class classification. Another disadvantage of such these functions is that they use exponential function when activating neurons and this causes high computational cost for the optimization task. Hence, it is more convenient to use Rectified Linear Unit (ReLU) non-linearity for CNN models. Because, ReLU does not saturate at the positive region, keeps, and converges the gradients well by stochastic gradient descent. In addition to these, ReLU activation function is faster and more cost-efficient than other activation functions. Other functions shown in figure 2.2 (Leaky ReLU, Maxout, ELU) are generally improved versions of ReLU. However, we use ReLU activation in most of CNN models which are used in this thesis.

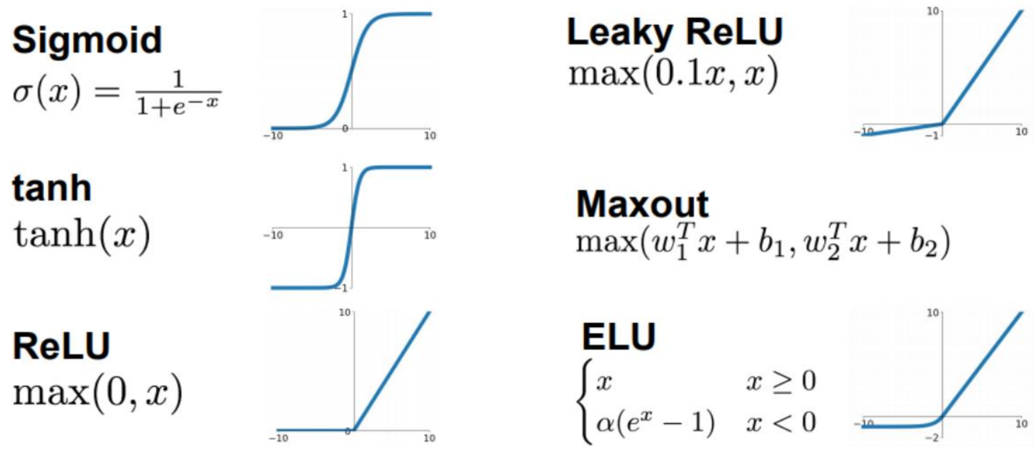


Figure 2.2 Generally used activation functions in ANN and CNN [22]

The softmax function is usually used when there is a try to handle multiple classes. It can normalize and press the outputs between zero and one [26]. Moreover, the softmax function can be employed in the output layer of the classifier for ideally getting the probabilities to define the class of each input [27].

$$f(z) = \frac{e^{z_j}}{\sum_{i=1}^{k} e^{z_i}} \; for\; j = 1, \dots, k \tag{2.1}$$

Finally, the sigmoid function is a good choice for output layer if output is used for binary classification, [28].

2.3 Learning Rules Kinds of ANNs

Mainly, there are two rules of neural network learning:

• Supervised learning

The learning rule is supplied with a group of patterns (the training group) for appropriate network behavior in supervised learning. When the input is applied at each instant of time, the desired response (d) of the system is supplied by the teacher, as shown in figure 2.3. The distance $p(d, y)$ between the desired response (d) and the actual response (y) serves as the measure of the error. The learning rate is utilized to correct the weights and biases for the network in order to make the network outputs closer to the desired response [29].

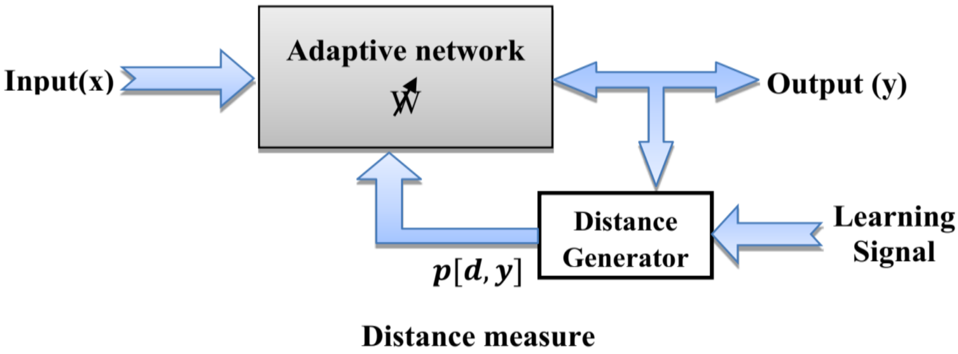


Figure 2.3 Block diagram of the supervised learning [29].

• Unsupervised learning

In this rule, weights and biases are adjusted in response to the network inputs only. There are no desired responses; as shown in figure 2.4. Most algorithms use the clustering operations. The input patterns are categorized into finite number of classes. This type of NN is particularly useful in such the vector quantization applications [30].

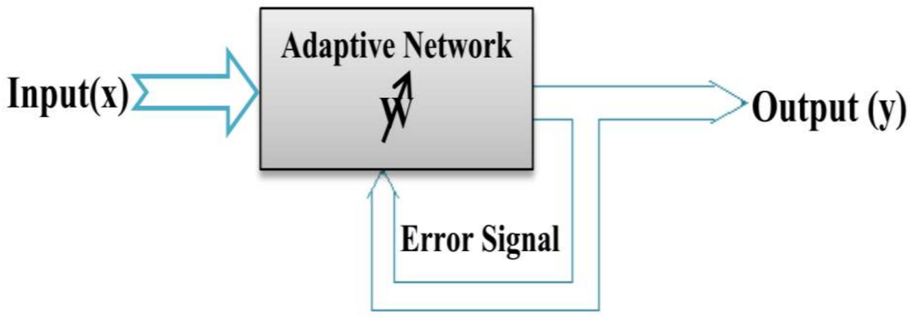


Figure 2.4 Block diagram of the unsupervised learning [30]

2.4 Back-Propagation Algorithm

The Back-Propagation (BP) algorithm was suggested earlier to determine the weights, and thus to train the Multi-Layer Perceptron (MLP) [31]. BP also indicates "propagation of error" is considered a supervised learning way. It requires a teacher who knows, or can be calculated, the desired output for any specified input. In BP, the errors propagate backwards from nodes of the output to nodes of the inner. This training algorithm is much used to calculate network error with respect to adjustable network weights [32]. Many factors are to be considered in the design like:

number of inputs, hidden and output nodes, activation function, learning rate and momentum rate. The architecture of BP neural network is shown in figure 2.5.

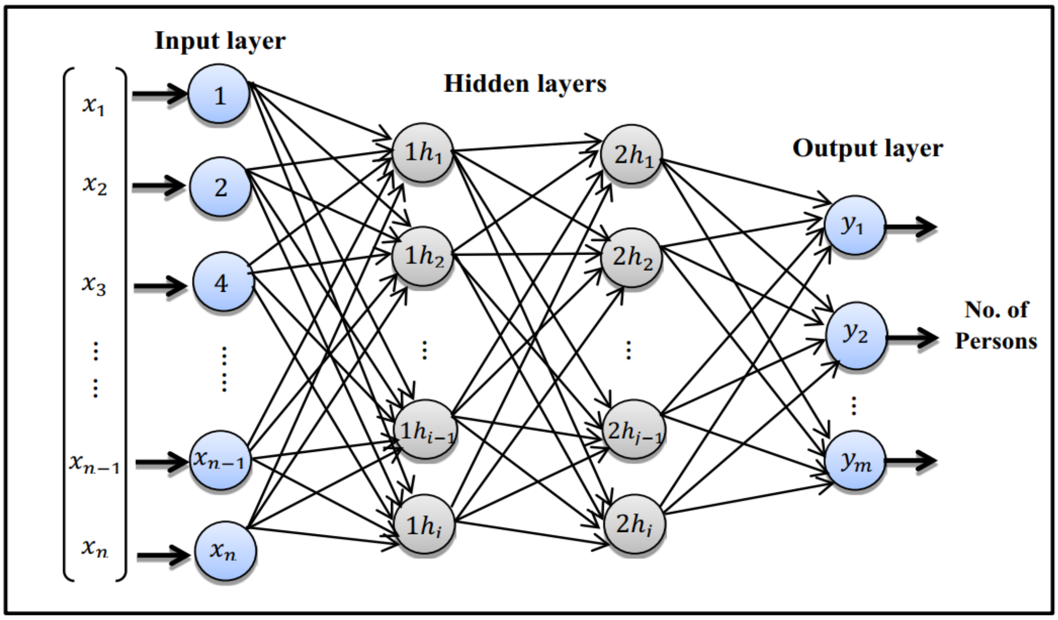


Figure 2.5 BPNN architecture

In Figure 2.5, n represents the number of features in pattern. While i represents the number of neurons in the first hidden layer, j represents the number of neurons in the second hidden layer, and m represents the number of neurons in output layer (number of classes).

The full BP with Stochastic Gradient Descent (SGD) training algorithm steps are [25]:

Step1: Initialize the network weight value.

Step2: Sum the weighted inputs and applied the activation function to compute the output of hidden layer.

$$h_j = f\left(\sum_i a_i \, w_{ij} + b_j\right) \tag{2.2}$$

where:

a_i : the input training vector

w_{ij} : the weight connected between input node i and hidden node j

b_j : the bias of hidden node j

h_j : the output of hidden node j

Step3: Sum the weighted hidden layer and applied the activation function to compute the output of output layer.

$$Y_k = f\left(\sum_j h_j\, w_{jk} + b_k\right) \tag{2.3}$$

where:

w_{jk} : the weight connected between input node j and hidden node k

b_k : the bias of hidden node k

Y_k : the output of hidden node k

Step4: Calculate the Back-propagation error.

$$\delta_k = (T_k - Y_k).\, f'(\sum_j h_j\, w_{jk}) \tag{2.4}$$

where:

T_k : the desired output at node k

δ_k : the error at node k

f' : the derivative of the activation function

Step5: Compute the correction term of weight between the hidden and the output.

$$\Delta w_{jk}(n) = \eta \delta_k h_j + \alpha \Delta w_{jk}(n-1) \tag{2.5}$$

where:

η : the learning rate

α : the momentum coefficient

Δw_{jk} : the weight adjustment between input node j and hidden node k

Step6: Sum delta input for each hidden unit one and calculate error term.

$$\delta_j = \sum_k \delta_k w_{jk} \, . f'\left(\sum_l a_l \, w_{lj}\right) \tag{2.6}$$

where:

δ_j : the error at node j and a_l : the input training vector at node l

w_{lj} : the weight connected between input node l and hidden node j

Step7: Sum the delta input for each hidden unit two and calculate the error term.

$$\delta_l = \sum_j \delta_j w_{lj} \, . f'(\sum_i a_i \, w_{il}) \; ..... \tag{2.7}$$

where δ_l : the error at node l

Step8: Calculate the correction term of the weight between the input and the hidden.

$$\Delta w_{ij}(n) = \eta \delta_j a_i + \alpha \Delta w_{ij}(n-1) \tag{2.8}$$

where Δw_{ij}: the weight adjustment between input node i and hidden node j

Step9: Update the weights.

$$\begin{aligned} w_{jk}(\mathrm{n}+1) &= w_{jk}(\mathrm{n}) + \Delta w_{jk}(n) \\ w_{lj}(\mathrm{n}+1) &= w_{lj}(\mathrm{n}) + \Delta w_{ij}(n) \end{aligned} \tag{2.9}$$

Step10: Repeat step2 for given the number of error.

$$MSE = \frac{1}{2P} \left[\sum_{R=1}^{P} \sum_{K=1}^{L} (T_K^R - Y_K^R)^2\right] \tag{2.10}$$

where (p) is the number of patterns in the training set.

Step11. End.

2.5 Convolutional Neural Networks CNNs [33]

As classical neural networks, convolutional neural networks (CNNs or ConvNets) are composed of multiple layers structure. It is made of neurons that have learnt tunable weights and biases assembled sequentially, i.e., one after the other in such a way that the outputs from one layer serve as the inputs for the following one. It represented mathematically as a dot product followed with a non-linearity functions. They use a set of different multilayer perceptron's to achieve various types of non-linearity for modeling input layers to the next stages. CNN is an effective algorithm of deep learning that can take an image as input, determines its importance (biases and learnable weights) for numerous aspects/objects, and has the ability to differentiate one image from another. The preprocessing needed in a CNN is far lower in comparison to other classification algorithms. The parameters were reduced by introducing architecture that forces spatial relationships and consequently, progress feed-forward and back propagation training.

The characteristics of the standard neural networks are applied to CNNs like gradient descent, regularization, and backpropagation, etc. [34]. They are also known to be shift invariant, which means they tend to be behaving the same wherever the focal point for the occurrence of a feature of interest is.

Generally, five basic structures are located in a CNN such as Input layer, Convolution layer, ReLU layer, Pooling layer and Fully-Connected Layer. Therefore, the input layer is used for holding the raw pixel values of images. Convolution layer is used for computing the output neurons, which is connected to the local regions of input and computes dot product between small regions and weights. ReLU layer is an activation function used to convert the applied data to another domain. Maximum Pooling layer is used to perform a down-sampling process across spatial dimensions and finally fully connected layer is used for computing the class scores.

2.5.1 CNN concepts

CNNs have a set of unique concepts that make it different from other types of neural network architectures. The main ones are illustrated as follows:

1. Input/output Volumes

CNNs are usually applied to image or video data. Every image is a matrix of pixel values. The range of values that can be encoded in each pixel depends upon its bit size. Most commonly, 8 bit or 1 Byte-sized pixels. Thus, the possible range of values a single pixel can represent is [0, 255]. However, with colored-based images, particularly RGB (Red, Green, Blue) images, the presence of separate color channels (3 in the case of RGB images) introduces an additional depth field to the data as shown in figure 2.6, making the input 3-dimensional with $3 \times 8 = 24$ bits per each pixel. It can convert the colored image into gray scale representation with 8 bits per each pixel.

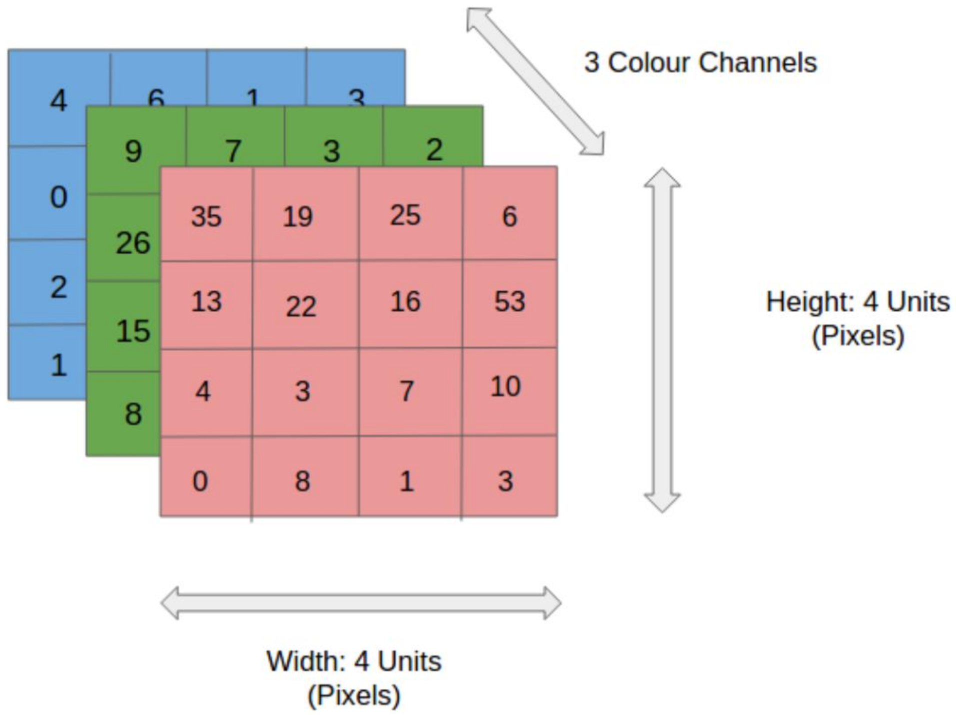


Figure 2.6 An input volume of size: $4 \times 4 \times 3$ [33]

2. Features

A feature is a distinct and useful observation or a pattern obtained from the input data that aids in performing the desired image analysis. The CNN learns the features from the input images. Typically, they emerge repeatedly from the data to gain prominence.

3. Filters (Convolution Kernels)

Generally, it refers to an operator applied to the entirety of the image such that it transforms the information encoded in the pixels. In practice, however, a kernel is a smaller-sized matrix in comparison to the input dimensions of the image that consists of real valued entries.

The kernels are then convolved with the input volume to obtain so-called ‘activation maps’. Activation maps indicate ‘activated’ regions, i.e. regions where features specific to the kernel have been detected in the input. The real values of the kernel matrix change with each learning iteration over the training set, indicating that the network is learning to identify which regions are of significance for extracting features from the data. The convolved value obtained by summing the resultant terms from the dot product forms a single entry in the activation matrix. The patch selection is then slided (towards the right, or downwards when the boundary of the matrix is reached) by a certain amount called the ‘stride’ value, and the process is repeated till the entire input image has been processed. The process is carried out for all color channels. For normalization purposes, divide the calculated value of the activation matrix by the sum of values in the kernel matrix. The process is demonstrated in figure 2.7; the convolution value is calculated by taking the dot product of the corresponding values in the Kernel and the channel matrices.

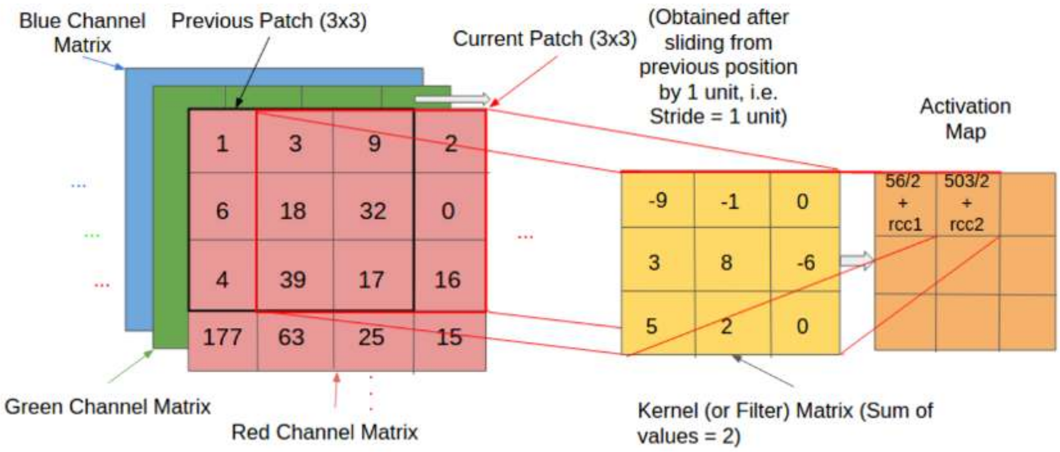


Figure 2.7 Kernel filter operations [33]

4. Receptive Field

Connect all neurons with all possible regions of the input volume. It would lead to too many weights to train, and produce too high a computational complexity. Thus, instead of connecting each neuron to all possible pixels, specify a 2-dimensional region called the ‘receptive field [35]’. Extending to the entire depth of the input. It is over these small regions that the network layer convolution operates and produce the activation map.

5. Zero-Padding

Padding has the following benefits: It allows us to use a CONV layer without necessarily shrinking the height and width of the volumes. This is important for building deeper networks, since otherwise the height/width would shrink as we go to deeper layers. In addition, it helps us keep more of the information at the border of an image. Without padding, very few values at the next layer would be affected by pixels as the edges of an image. Zero-padding refers to the process of symmetrically adding zeroes to the input matrix as shown in figure 2.8.

0	0	0	0	0	0
0	35	19	25	6	0
0	13	22	16	53	0
0	4	3	7	10	0
0	9	8	1	3	0
0	0	0	0	0	0

Figure 2.8 A zero-padded 4×4 matrix becomes 6×6 matrix [33]

2.5.2 The CNN architecture

A simple CNN consisting of main types of layers like Convolutional Layer, Pooling Layer, Rectified Linear Unit Layer (ReLU), Fully-Connected Layer and other different functionally layers. These layers are a stack to shape a full CNN architecture [33].

1. Convolutional Layer

As the name suggests, this is the core structure block of a CNN. It involves a set of filters which have been convolved across the image height and dimensions.

Each convolution layer consists of a set of layers that comprise a group of neurons, every neuron in the same layer shared the same weights and biases. The convolution method can handle three ideas which can help boost a machine learning system, specifically sparse interactions, equivariant representation and parameter sharing. In addition, it likewise to some degree would make it invariant to distortions, scaling and shifts. The result of the input convolution with one applied filter is known as a feature map, or often called an activation map. As a result, each filter in the layer have one feature map, and altogether they constitute the output depth. The spatial size of every feature map is determined by the input image size, filter size, stride and padding. Since the filter is smaller in size than the input, it can cause sparse interactions. Overlay the filter to the input, perform a dot product between their weights and input volume as shown in figure 2.9.

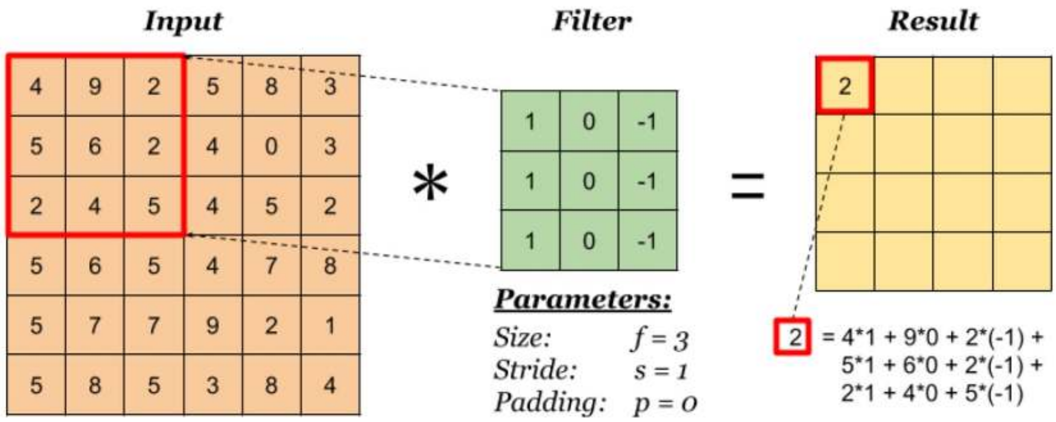


Figure 2.9 Convolution operation [33]

Then move the overlay right one position (or according to the stride setting), and do the same calculation above to get the next result. And so, on as in figure 2.10.

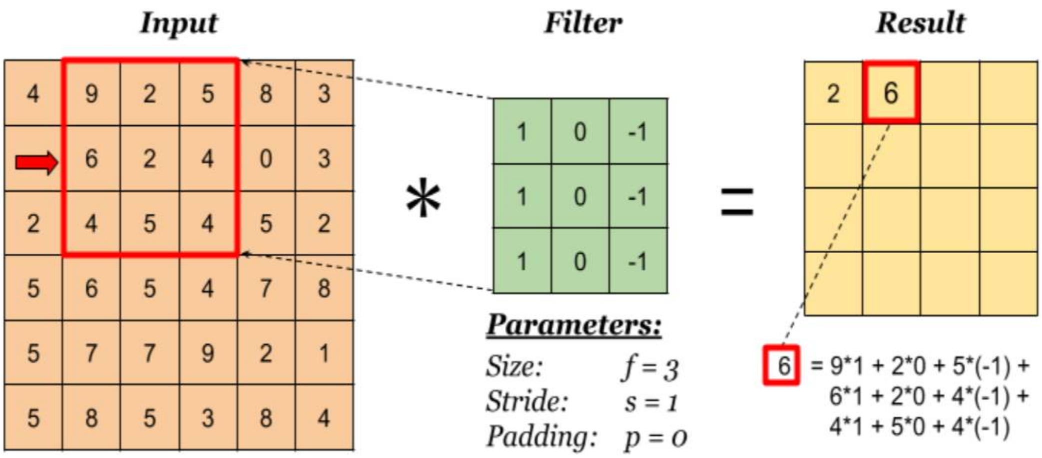


Figure 2.10 Convolution operations with stride [33]

Convolutional layers yield small connections, localized region of the input image. This kind of localized construction ensures that the learned filters can produce the strongest response on local input. Therefore, at lower layers, simple features of few parts of the input can be generated. While, at the upper layers produce a more abstract representation of larger areas.

During the forward process, the filters are convolved with the image across the input volume dimension (width, height and depth) and calculating dot products with input at any position as showing in figure 2.11. The filters are activated when they visual features like the edge of some orientation or color.

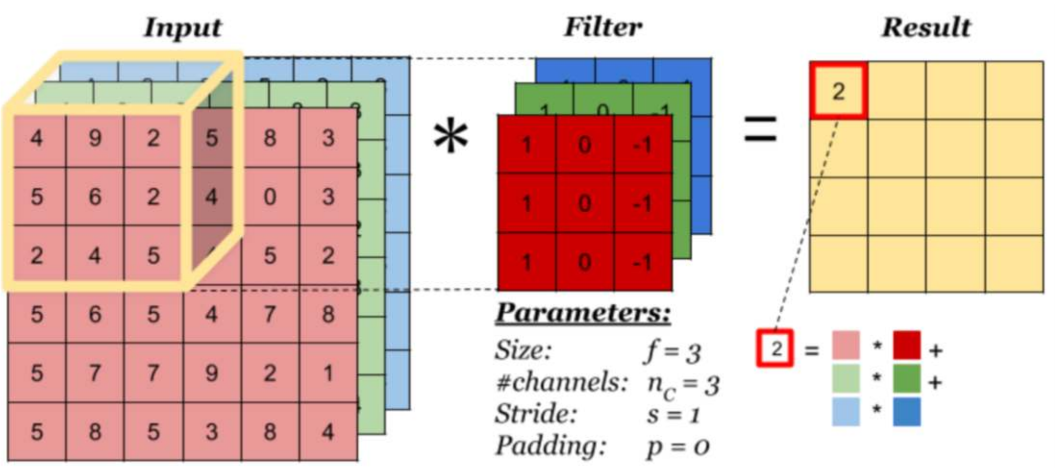


Figure 2.11 Convolution operation on volume [33]

The sizes of the filters used in the convolution layer are usually 3 × 3 and 5 × 5 [36]. The output depends on the size of the filter that are used and the amount of zero padding and called after the filtering process feature map, or activation map and calculated as:

$$a_{x,y} = f\left(\sum_{f1}\sum_{f2}(w_{f1,f2} \, . \, a_{j+f1,k+f2} + b\right) \tag{2.11}$$

Where $a_{x,y}$ activation map and b the filter bias, $f1, f2$ set from zero to the size of the filter, thus $w_{f1,f2}$ kernel weights, j, k the local receptive field of the input map and $a_{j+f1,k+f2}$ local the input values and finally f is the activation function.

Each filter produces a unique feature in one layer and each output filter make up a depth output. The output size depends on the size of the filter, padding and the size of the imported image.

Multiple filters can be used in a convolution layer to detect multiple features as shown in figure 2.12. The output of the layer then will have the same number of channels as the number of filters in the layer.

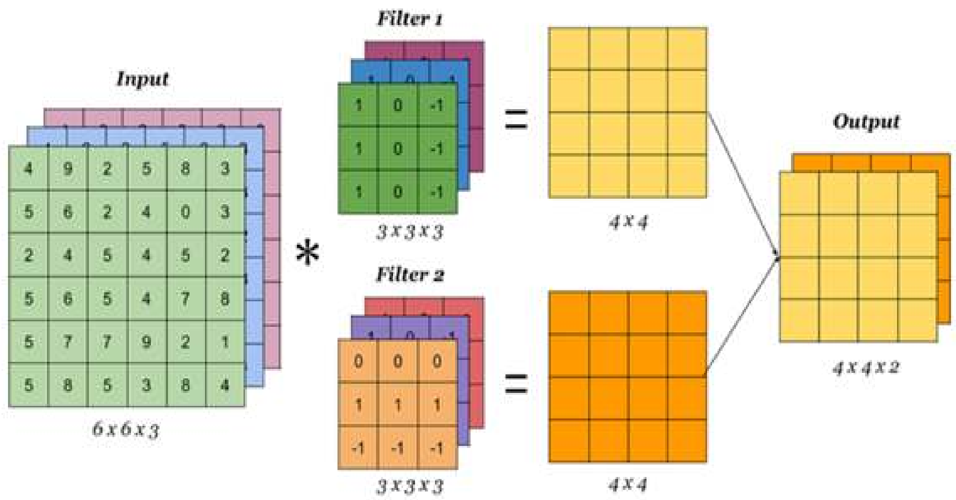


Figure 2.12 Convolution operation with multiple filters [33]

The most important features of the convolution processes can reduce the dimensions and number of learner parameters experienced by standard neural networks. This essential reduction of weights of the CNN allows creating bigger and deeper networks. As the kernel is smaller than the input, the convolution can work with inputs of the variable size and get sparse interactions. This means that fewer parameters were needed to store that reduces the memory requirements and improves statistical efficiency and the computing output have fewer operations [33]. The parameter sharing vital concept that using by the convolution operation which explains instead of a separate learning set of parameters for every location, in matrix data such as images which its pixels often highly correlated, forming distinctive local subjects that are easily detected. The local statistics of images and other signals are invariant to location. In other words, if a representation can appear in one part of the image, it could appear anywhere. Therefore, the idea of units in different locations is to share the same weights and detecting the same pattern in different parts of the array [34]. Almost all units on a feature map share these corresponding parameters, it is usually interpreted as though a feature map, identifying several features such as vertical and horizontal edges, making independent decisions on which edge of the input to be detected. On the other hand, it's the input’s relative positioning that's of interest. This variable sharing will save an appropriate amount of memory. Even so, the separate feature maps are not going to share parameters, as they are detecting several features. Besides that, this type of parameter sharing in case of convolution would make the function equivalent to translation, which mean when the input changes the output can change in the same way.

2. Pooling Layer

The pooling layer is always following the convolution stage, it is a subsampling layer that reduces the matrix dimensionality and offers some robustness to spatial shifts in some element's numbers. The role function of the pooling layer is to combine nearly similar features into one because any feature that can found its exact location, it will not be affected by other rough location relative features. So, in this way only the most activated weights and unit were kept [34]. There are several pooling actions that exist through using small filter to sample the maximum or average values in a rectangular area of the output from the previous layer. The common pooling layer with kernel size of 2×2 that is generally applied along both width and height with a stride of 2 down samples every depth slice. Figure (2.15) shows several samples of pooling layers operation, the box to the left is the input to the pooling layer and it has been down sampled by getting the maximum value of each 2×2 sub-regions.

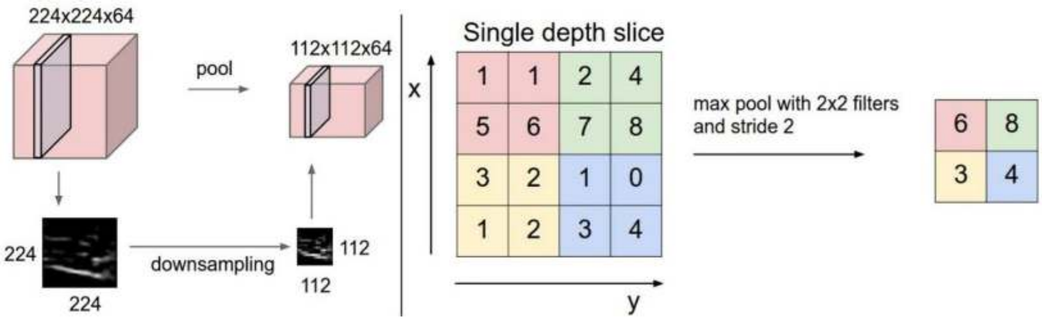


Figure 2.13 Examples of max pooling layers [34]

Another difference of CNNs with standard neural networks is the existence of pooling layers that aims to simplify the information at the output of the convolutional layers by reducing it [26].

3. Rectified Linear Unit (ReLU) Layer

Rectified Linear Units (ReLU) are widely used layers that operate as positively half wave rectification circuit, the activation function is with $f(x) = \max(0, x)$ in hidden layers of the neural network. It gives an output x if x is positive and 0 otherwise. Figure 2.14 illustrates how this works. It can be established that the network has the ability to train several times faster by using this non-saturating function, in comparison to using saturating functions for example: the tangent function with $f(x) = \tanh(\alpha x)$, or the sigmoid function with $f(x) = \frac{1}{1+\exp(-\alpha x)}$. ReLU performs simpler mathematical operations so it has less computationally expensive than the other activation functions. The main advantages only a few neurons are activated that yielding the network sparse making. Basically, it is efficient, easy computational and does not saturate like sigmoid and tanh functions which means it is robust and has resistant to the vanishing gradient problem in the positive region. The only drawback of ReLU units is that the outputs are not zero centered.

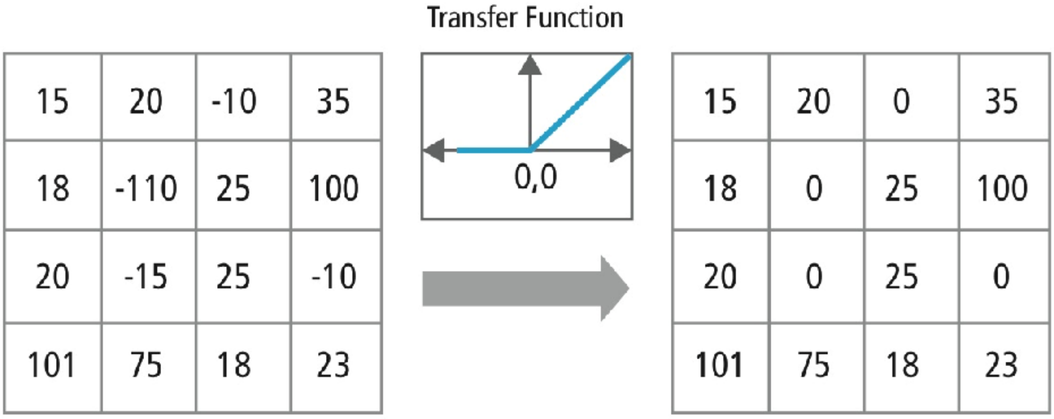


Figure 2.14 An example of how a rectified linear unit layer operates

4. Normalization layer:

Numerous types of normalization layers are generally proposed to normalize the data within a certain range like normalizing the applied input data between 0 and 1; however, it is not yet proven to be a beneficial practice and has not gained any solid ground.

5 Loss layer

Typically, it is the latest layer in the network that computes the task objective, including classification, by e.g. making use of the softmax function, as in equation 2.12.

$$f(z) = \frac{e^{z_j}}{\sum_{k=1}^{k} e^{z_k}} \tag{2.12}$$

For $j = 1, \dots, k$ and $z = z_1, \dots, z_k$

Where $f(z)$, The standard (unit) softmax function; z, the input vector.

6. Fully connected layer

Convolutional Neural Networks used fully connected layers to perform the high-level reasoning. This layer can be connected to all activations in the previous layers. Their activation functions can be easily implemented with a matrix multiplication followed by a bias offset [34]. In other words, neurons in such layer are totally connected to every activation in the prior layers, as in standard neural networks. These are generally at the network end, such as outputting the class probabilities.

By Combined of the described layers it can be utilized to form a CNN architecture. A regular architecture pattern for a CNN can be described in follows:

$$\begin{gathered} INPUT \rightarrow \big[[CONV \rightarrow RELU] * N \ \rightarrow \ POOL?\big] * \\ M \ \rightarrow [FC \ \rightarrow \ RELU] * K \rightarrow FC \end{gathered} \tag{2.13}$$

Where, "*" indicates repetition, "POOL?" indicates the optional pooling layer. The symbols N, M, and K are integers equal or higher than zero. N is generally lower than or equal to 3 and K strictly lower than 3. Pool shows that the pooling layer is optional. It's typically a good idea to stack extra convolutional layer prior to the pooling layer for bigger and deeper networks, as the convolutional layer can easily detect further complex features from the input volume prior to the destructive pooling process. FC indicates the fully connected layer which is the last CNN layer. It is popular to make use of dropout through the training on the fully connected layers. In order to reduce overfitting, we can use a dropout rate which is a simple way. At the time of training, specific nodes are deactivated by certain probability $(1 - p)$, or held it active by used of probability p. The outgoing and incoming connections to a deactivated node may be dropped. Besides minimizing overfitting, it can also reduce the amount of computational requirement and makes it possible for better performance. However, through testing every node is activated. During weights initialization of the network, it is significant to not set all of them to zero, because it may lead to unwanted balance in the updates. Rather it is often a good idea to set them to little, random numbers, for example by sampling them from a Gaussian distribution. For training, there is a need to loss expression for minimization.

2.5.3 CNN sample complete network

This is a sample network with two convolution layers. At the end of the network, the output of the pooling layer is flattened and is connected to a logistic regression or a softmax output layer.

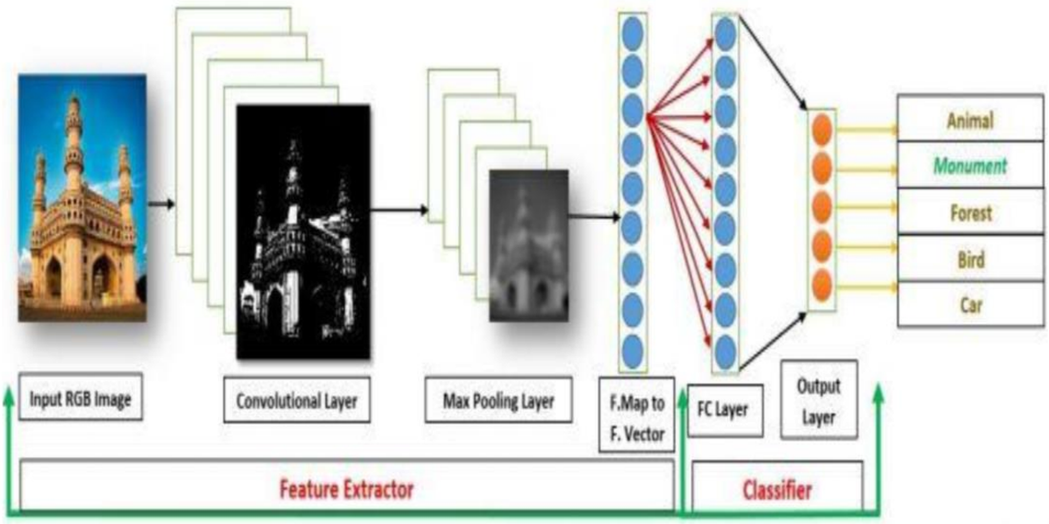


Figure 2.15 CNN architecture for computer vision task [37]

The CNN architecture is characterized by the number of layers, the number of filters per layer, the size of kernel filter of convolutional layer, the size of the subsampling window; the architecture is as shown in figure 2.15.

• Input layer is the first layer, or the images. There could be 1 channel or 3 channels to accommodate color.

• The second layer is convolution layer which calculates the output of neurons that composed of 3 feature maps. To get from the input layer, the function is mapped, the input layer is convolved with kernel, added with a bias, and then passed through activation function like as a sigmoid function. The kernels are initialized randomly and later updated after each run of the backpropagation algorithm [36].

• The third layer performs a down sampling operation with the spatial dimensions by applying pooling layer.

• Fully-connected layer will compute the class scores.

• Output layer predict the class score.

Using more than one convolutional layer before the pooling layer is often a good procedure in order to recognize more complex features of the input array stream before pooling operation [35]. It is useful to apply dropout during training on the fully connected layers to reduce overfitting during training. When initializing the weights and biases of the network, it is important not to set all to zero as this can lead to undesirable symmetries during updates. Instead, it's a good idea to put them on small random numbers [38].

2.5.4 Existing networks

Many different designs of CNN architectures have been developed, some types of them are explained as in follows:

1. LeNet 1998 (or LeNet-5) [39]

In 1998 LeCun has been successfully developing a CNN application, well known as LeNet-5, which has been used for handwriting recognition. LeNet-5 architecture is shown in figure 2.16. It involves seven layers, without counting the input layer. The size of input images that used is 32×32. The first layer of CNN consists of six 5×5 filters, that after the convolution reduces the image size to 28×28. Next the convolution, a sub-sampling layer is implemented by max pooling, and after that another 16 filters of size 5×5 for the second convolution layer, then the final sub-sampling layer. Finally, the feature maps have finally been brought down to have a size of 5×5, prior to going into the fully connected layer.

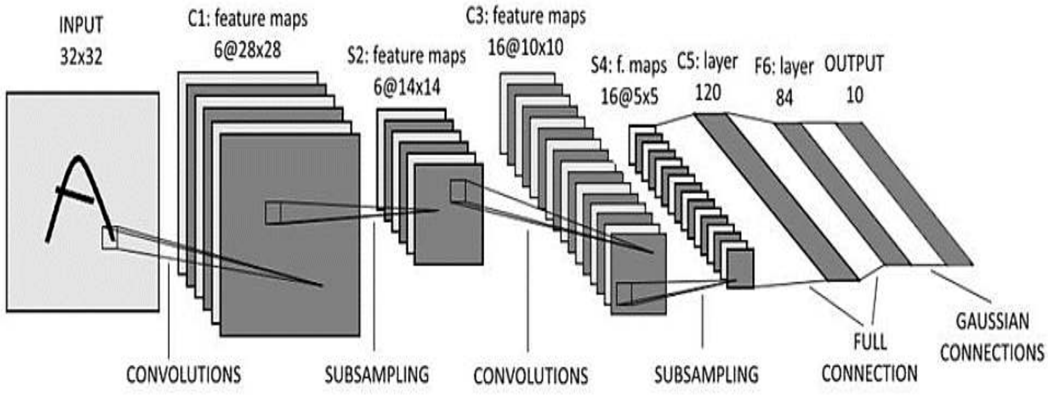


Figure 2.16 LeNet-5 (1998) Architecture

For more details, it can follow the example in figure 2.17 bellow:

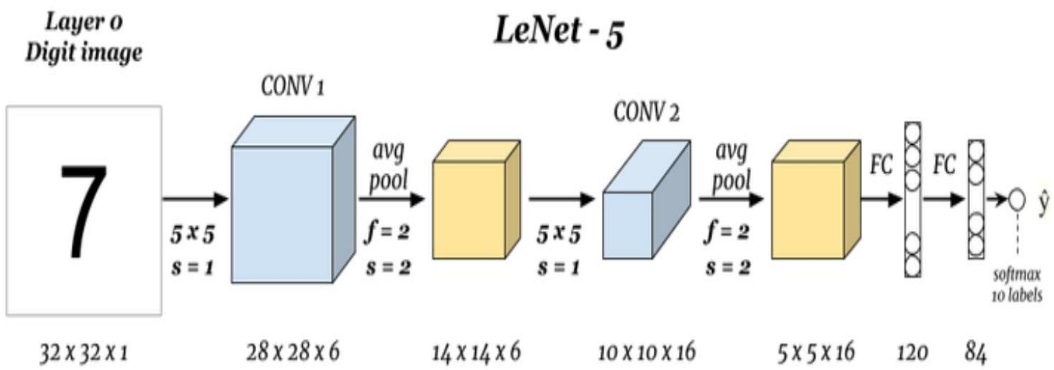


Figure 2.17 An example to show the architecture of LeNet-5 [39]

1. AlexNet 2012 [40]

AlexNet has become one of CNN architecture in 2012, which was named, related to one of its creators Alex Krizhevsky. The network had a very similar architecture as LeNet but was deeper, with more filters per layer, and with stacked convolutional layers. It consisted 11×11, 5×5, 3×3 convolution filter sizes, max pooling, dropout, data augmentation, ReLU activations, SGD with momentum, Figure 2.18 shows the AlexNet architecture. The images of size 227×227×3 has been used as input and the first convolutional layer made use of 11×11 filters with stride four to produce 227×227×3, while the rest of the convolutional layers make use the filters of size 3×3. The main differences between AlexNet and LeNet are that it is a deeper and bigger network, it utilizes ReLU layers, and trained on 2 GPUs with more data. Notable is that AlexNet is also used a normalization layer, that was very favorite at the time but is not generally used anymore. AlexNet was trained for 6 days simultaneously on two Nvidia GeForce GTX 580 GPUs which is the reason for why their network is split into two pipelines.

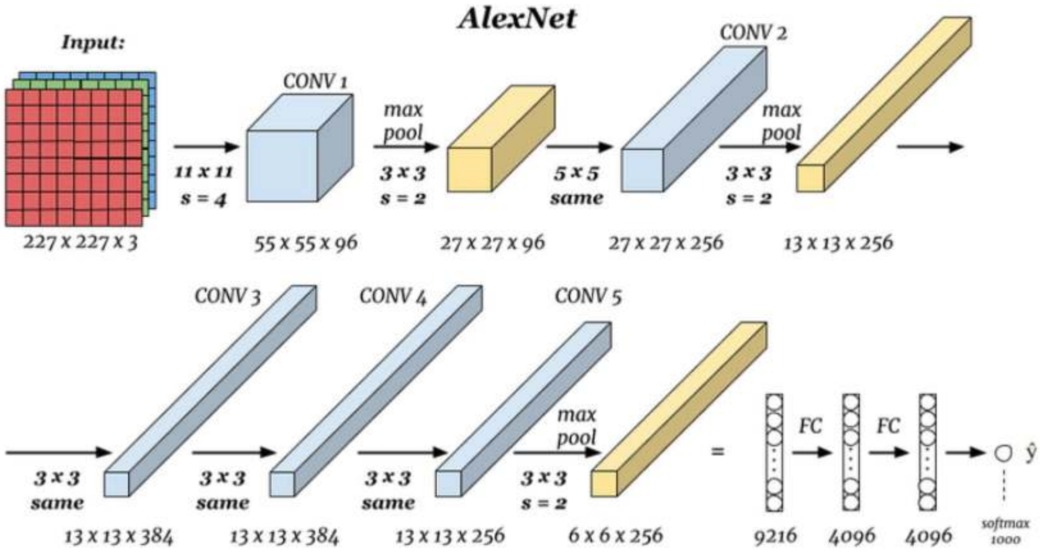


Figure 2.18 AlexNet (2012) architecture [40]

2.GoogLeNet, 2014

The GoogLeNet was published in 2014, which is a deep network of twenty two layers; Figure 2.19 shows the structure of the GoogLeNet network. It presented a "network in network" inception model, and it make use of a 12th of parameters that AlexNet used.

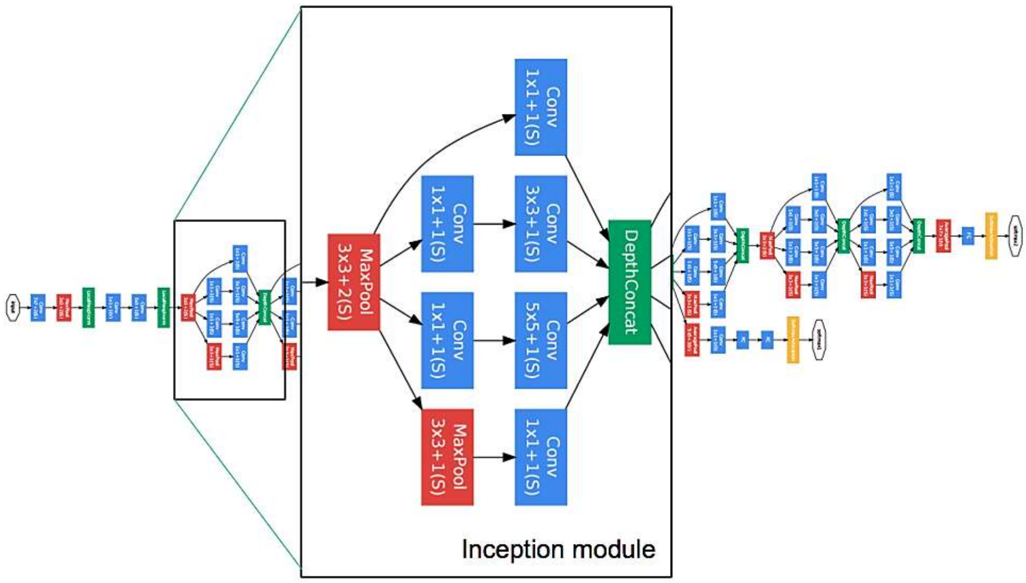


Figure 2.19 Very Complex GoogleNet Network Structure

3. VGGNet, 2014

VGGNet published in the same year as GoogLeNet (published in 2014). This network has a configurations range from eleven to nineteen weight layers, that is fully connected with convolutional layers, with a over-all number of 133 million weights for the minimum configurations, to 144 million weights in the largest sized configurations. Although GooGleNet performs better than VGGNet, VGGNet is widely used due to its less complex architecture. Figure 2.20 shows a table configuration of VGGNet. This offers an advantage, as training complex models demand weeks of works on one or extra GPUs. Also, these models can be saved and then reused for a number of additional applications, with no or minimal supplementary work.

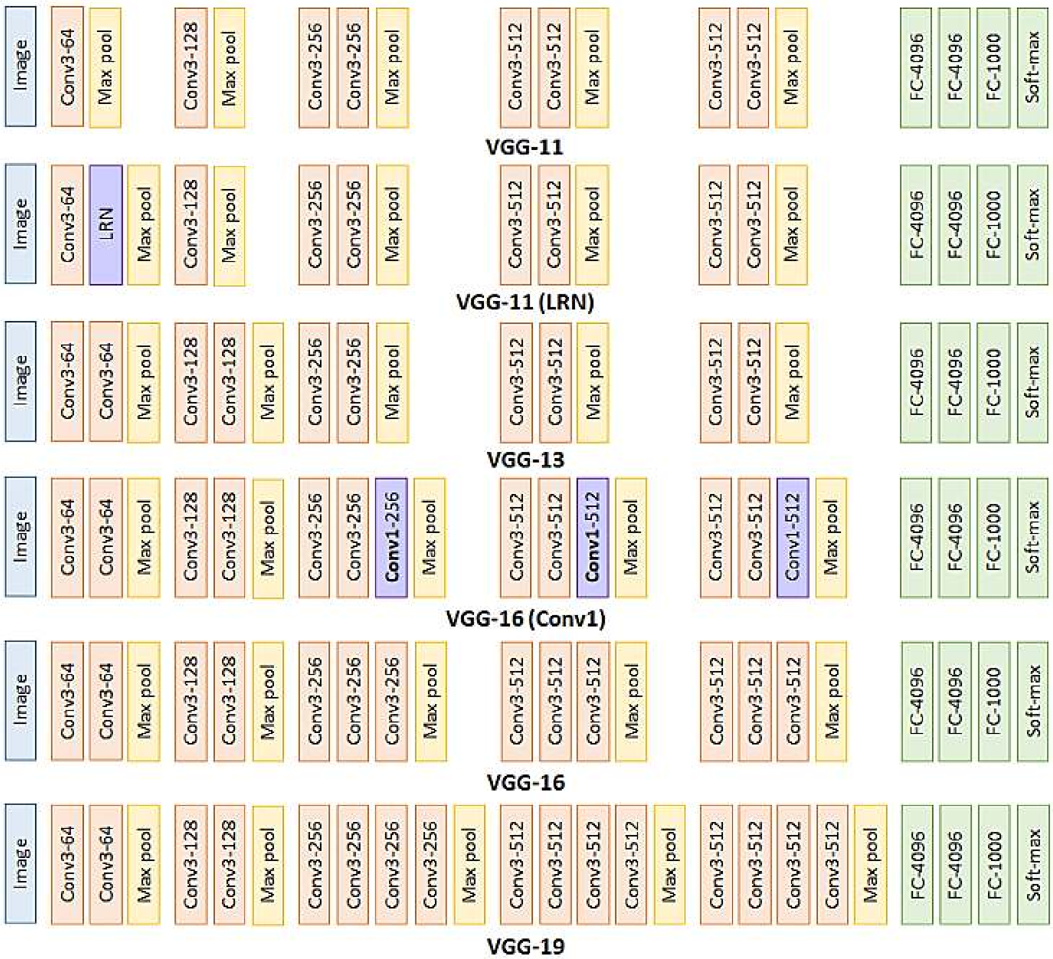


Figure 2.20 CNNs of VGGNet Configurations is shown in the columns. The depth rises from left to right

VGG16 Net as an example for VGGNet is one of the most effective model structures, as it reinforces the concept that convolutional neural systems should have a deeply layered network to represent different levels of information. The famous Visual Geometry Group (VGG) in Oxford developed a VGG16 network for object recognition and achieved good results on the ImageNet data set. VGG supports the simplest 3 × 3 convolution filter size and the 2 × 2 grouping is used in the network. The Oxford teams made the structure and weights of the network formed freely available online. The drawback of VGGNet is big and includes around 160 million parameters. Most settings are used in fully connected layers (FC). It can detect one of 1000 different object categories with an error rate of 7.3%. The architecture of the model is shown in Figure 2.21 [41]. In this approach, the VGG16 network was used for feature extraction. Only the fully connected layers were removed.

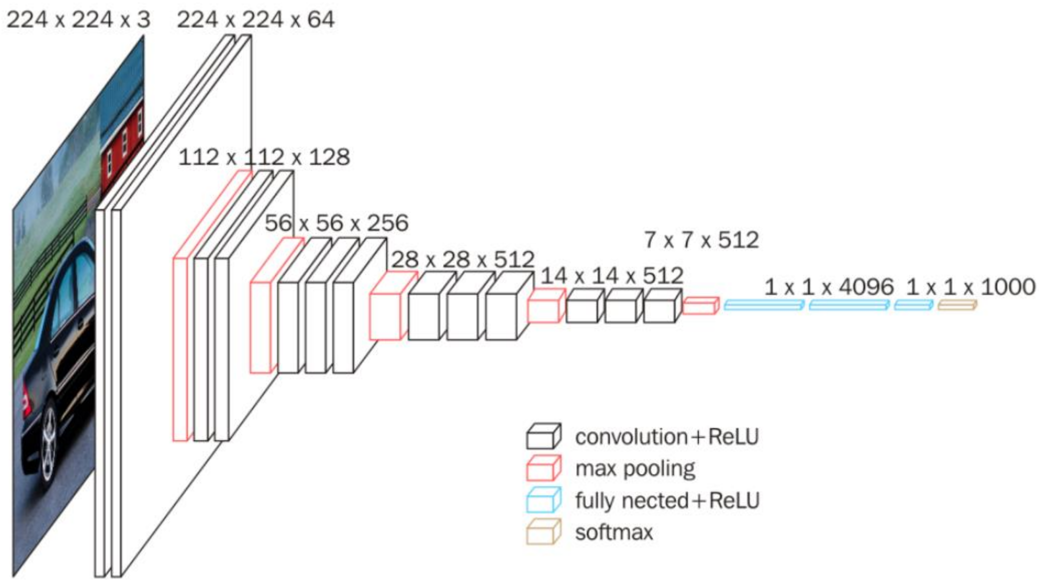


Figure 2.21 VGG16 architecture [41]

2.6 Hardware for Deep Learning

Because of its large size and large number of parameters that is used by deep learning algorithm, the training often will take a very long time and the selection of hardware is therefore of utmost importance. This part aims to outline some options relating to the existing CPU and GPU reviews.

2.6.1 Central processing unit (CPU)

The central processing unit often referred to as the CPU is the main brain of the computer and deals with all computations carried by a computer.

Typically, ANN has to be trained mainly on a computer's CPU that operates as the 'brain' of a computer and works by sequentially executing calculations.

2.6.2 Graphics processing unit (GPU)

A GPU is a specific type of processor that performs exceptionally well at parallel computing. CPUs in personal computers commonly have between a single up to many cores, (e.g. 5 cores), however, the CPUs of high-end server can have higher cores reach to 16 cores. In other hand, GPUs make these numbers light compared with GPUs cores which they boast thousands of cores in high level GPUs. The nature of parallel operation for both CPU and GPU is represented as in figure 2.22. However, GPUs are slow at sequenced operations in comparison to their CPU equivalent, but does well in tasks that can be performed in parallel. This is regular in three-dimensional graphics, in which a three-dimensional landscape kept in the memory should be planned onto a two-dimensional image to be displayed on a screen. Current CPUs generally have a GPU integrated inside them however it can also be a dedicated chip on its own, which is often named a dedicated GPU. In cases where GPU is integrated, then it shares with the CPU the system memories and it take a part of these memory to use. In other hand, when GPU is dedicated it generally has its own memory that is much faster than the system memory and as a result speeds up the operations. Laptops frequently have integrated GPUs however; there are also laptop models which have a separate GPU in use. For most desktop computers, it commonly has a separate GPU which works as a graphics card that can be placed and detached from the computer. In some cases both of GPU (integrated and separate) may be used by the related computer for power-saving functionality. As a dedicated GPU required higher power consumption compared to the integrated GPU, thus, the integrated GPU is put to use until the computer requires to execute more intensive calculations, after that it switches to the dedicated GPU.

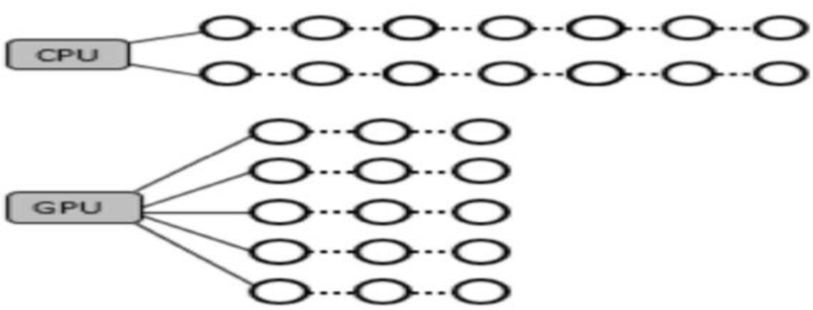


Figure 2.22 Parallel computing styles of CPU and GPU

As the deep learning algorithm training operations is usually done in parallel, thus, GPUs have considerably increased to become a highly useful tool as they help to make the training many times faster when compared to CPU alone. However, it needs the capability to program the GPU to run several codes, and must be supported by the manufacturers. Nowadays, two major manufacturers are present - which are NVIDIA and AMD. For NVIDIA GPUs which is capable of processing over 40 million images in a day with a single GPU (approximately 2 ms per image) [7]. Such a performance level is of great importance for deep models research, NVIDIA GPU offers CUDA language, that's designed to enable code to be run with their GPUs. In other common types of processors of other manufacturers, it can be use OpenCL that is utilized for parallel computing. The OpenCL is designed by The Khronos Group that is a non-profit consortium which creates open common application programming interfaces (API). As the OpenCL applicability is very attractive, CUDA has better supported by machine learning software. Thus, NVIDIA cards become certainly the most often graphics card that utilized when working with algorithms of machine learning. In recent years many manufacturers are working with the goal to have a part of this growing new market. For example, Intel company has been tweak their server CPUs to achieve better with machine learning, Also, Google was providing a chip which will perform machine learning activities with energy effectively. AMD has even released a new GPUs line that manufactured specially for machine learning that was released in 2017, besides a software tools to provide better performance.

In addition to these hardware, recent studies focus on the use of microprocessing unit and single board computers for machine learning and tries to find solution that processes these operations in a low computational hardware.

2.6.3 Programming Languages

They're significant numbers of programming languages that have been designed for Raspberry Pi. Python programming language is advised by Raspberry Pi foundation. Generally, any programming language that can be made for ARMv6 can operate on the Raspberry Pi. As a result the users aren't restricted to only use the Python. In addition, there are several preinstalled languages in Raspberry Pi such as C, Scratch, C++, Ruby and Java.

2.7 Salient Region Detection and Segmentation

In computer vision field, a saliency map is an image in which displays each pixel's unique quality. The objective of a saliency map is to make simpler and/or change the image representation into things that is more significant and much easier to analyze. To illustrate, if a pixel has got a high grey level or other exclusive color quality in image, that pixel's quality shows in the saliency map and in an obvious way. Saliency is an image segmentation kind. Determining visually salient regions is important in applications including adaptive content delivery, object-based image retrieval and smart image resizing.

2.7.1 Saliency models

There are two types of saliency models: local and global. In order to measures the saliency of an image patch, the local contrast takes the saliency of image patches regarding local neighborhoods. While, in global contrast, the contrast is computed regarding the patch statistics along with the whole image. The process of calculation is as follows:

Local saliency ($\boldsymbol{S_l}^{\boldsymbol{c}}$): it represents the average weighted significant difference between the center patches region (L) (red rectangle in figure 2.23) and the surrounding patch (i) in the neighborhood (blue region in figure 2.23) and the:

$$S_l{}^c(p_i) = \frac{1}{L}\sum_{i=1}^{L} W_{ij}{}^{-1} D_{ij}{}^c \qquad (2.14)$$

Where W_{ij} is the weight between the surround patch i and center patch j and D_{ij} is the distance between i and j patch in the element space.

Therefore, all those patches further apart from the center patch should have fewer effects on the saliency of the center patch.

Global Saliency ($\boldsymbol{S_g}^{\boldsymbol{c}}$): It commonly appears that a local patch is identical to its neighbors, however, the whole area (that are, local & surrounding) is yet in global in the whole scene. If uses only the local saliency might reduce areas inside a homogeneous area causing blank holes, that often impedes object-based focus (for example, a uniformly textured object could solely be salient at its edges). To solve such disadvantage, in this work the global saliency is built by using an operator guided through the saliency measure of data. Rather than used each pixel, in this method we calculate the possibility of every patch $P(p_i)$ across the whole scene and make use of it is inverse to get global saliency S_g by used this formula:

$$\log\,({S_g}^c(p_i)) = -\log\left(P(p_i)\right) = -\sum_{j=1}^{n} \log\left(P\left(\alpha_{ij}\right)\right) \qquad (2.15)$$

In order to calculate the possibility of every patch $P(p_i)$, we consider that coefficient α_{ij} happens to be conditionally independent from one another. This is to some level stated via using sparse coding algorithm. For every patch coefficient, its description vector (which is an initial binned histogram (100bins)) is determined from each of the patches in the scene, and then transformed to a $P\left(\alpha_{ij}\right)$ by dividing it by its sum. In a case where the patch is hard to find in one of the features, the previous product gets a little value causing higher global saliency for the overall patch.

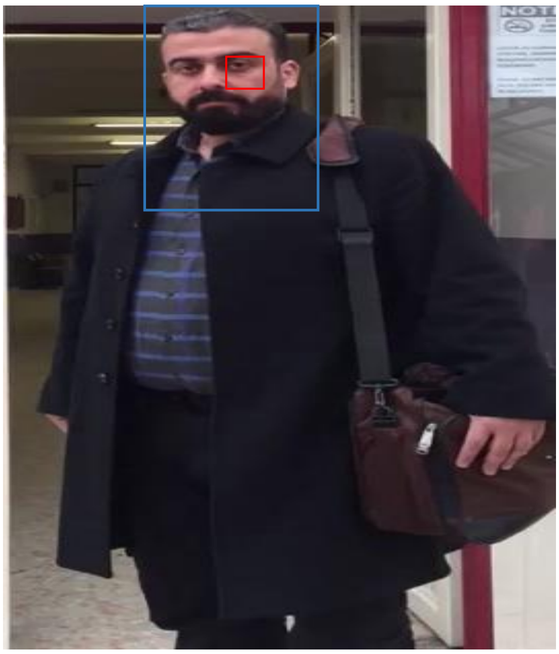


Figure 2.23 Illustration of local/global saliency of an image patch [45]

2.8 Euclidian Distance

The Euclidean Distance (ED) compares the similarity between two feature vectors of image by estimating the square root of the sum of the squared absolute differences [42]. The ED is given by:

$$ED = \sqrt{\sum_{i=1}^{N}(x_i - y_i)^2} \tag{2.16}$$

where

N =the number of all features in one template

x_i = template stored in database and y_i = measured output.

Manhattan Distance (MHD) is also called as city block distance that measures the similarity between two images by taking the sum of the absolute values of the differences between the two feature vectors [42]. The MHD is given by:

$$MHD = \sum_{i=1}^{N}|x_i - y_i| \tag{2.17}$$

The computational complexity is different for different distance measures. This is computed based on number of multiplications, additions, and subtractions. For a feature vector of size N, ED requires N subtractions, N squares of the previous, $N - 1$ further additions and one square root at the end. Similarly, MHD requires N subtractions for, $N - 1$ addition. Where is the feature vector of the test image, while is the feature vector of another image available in [42].

2.9 Template Database Generation

Essentially, in order to train any classifier approach like the ANN or CNN and test its performance, the preprocessed input image patterns should be divided into two groups: training group and testing group.

As explained in the previous sections, the Convolutional layers plus Pooling layers act as feature extractors from the input image or the input frame, and the Fully Connected layer acts as a classifier. As in the traditional classifiers (figure 2.24), the CNN has two phases: the training phase and the prediction phase. During the training phase, a convolutional neural network has been trained, using a dataset composed of cloud images or videos and the corresponding labels of these cloud visions. During the prediction phase, a trained model has been used to predict the labels of some test images or frames.

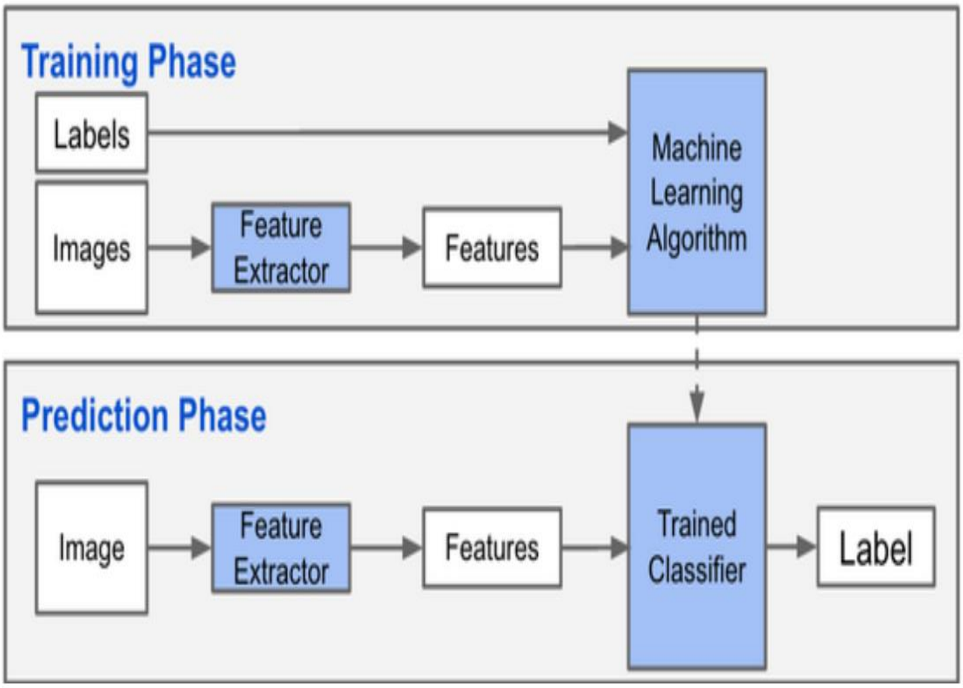


Figure 2.24 Training and Prediction Phase [43]

In order to start the classification, we need to define the input data and their labels. These data leads to extract the features and then while using these features with their labels, a machine-learning algorithm is applied to produce the classification model. After getting the trained model, it will be able to predict the correct label for any applied data with high accuracy or low error.

In this work, the percentage of image samples that are engaged into training group were approximately half or two-thirds percentage of the whole samples' numbers against half or one-third percentage for testing patterns group.

CHAPTER THREE

PROPOSED SYSTEM DESIGN

3.1 General

This chapter presents an experimental setup of power management system which includes system hardware and software design in addition to specify the hardware component used and the program operation and its algorithm.

3.2 System Methodology

The aim of this work is to propose and design a smart low-cost power management system based on efficient artificial intelligent approaches. The system must control power consumption and includes some appliances that can serve that purpose. Besides that it can control the electrical devices such as switching lights, TV; air-condition units, ventilation units, and other such devices on/off depending on some aspects such as the presence of individuals, temperature, odors. This system should control all these aspects in a smart automated way for most building area, such as rooms, halls, corridors, etc. The system is composed of control unit which control all process, sensors (such as light sensors to detect light intensity, temperature sensors to detect temperature in desired area, ultrasonic sensor to detect motion and computer distance, an odors sensors to detect the odors), the power meter to measure power consumption, a camera to streaming constant video so as to distinguish the presence of individuals and their activities. Figure 3.1 demonstrates an example of the power the executive's framework in buildings:

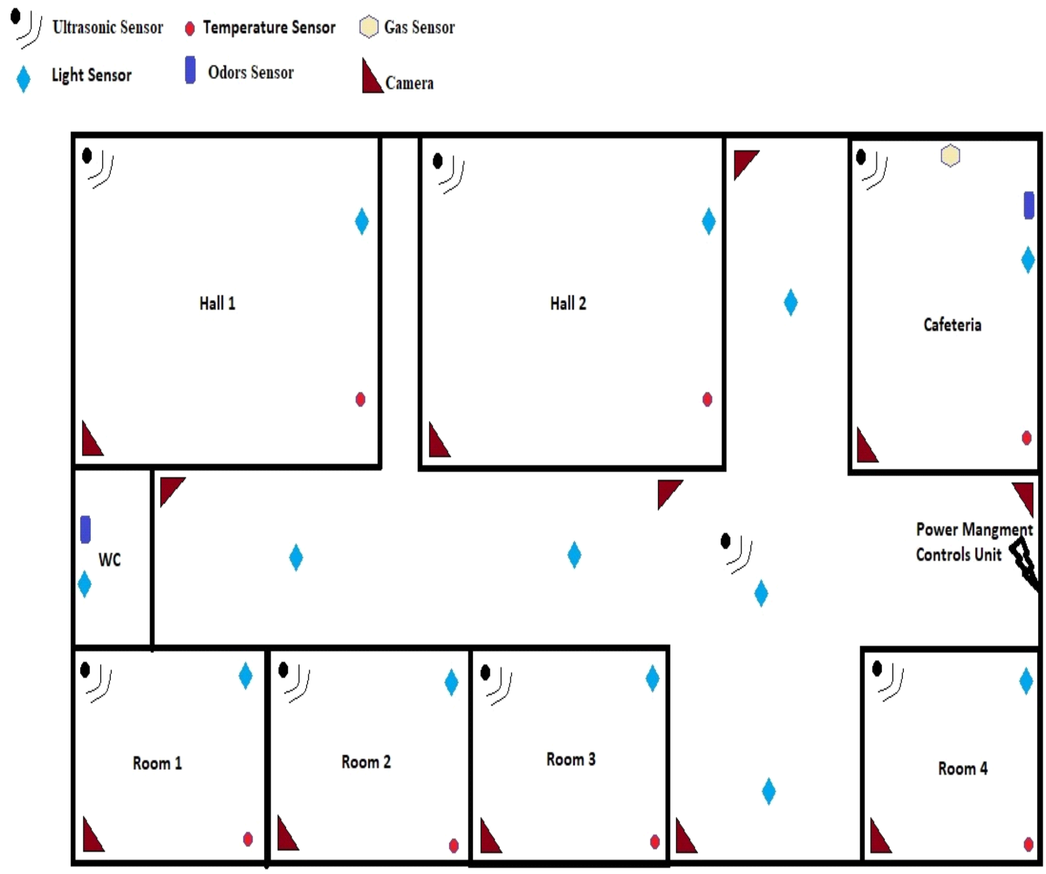
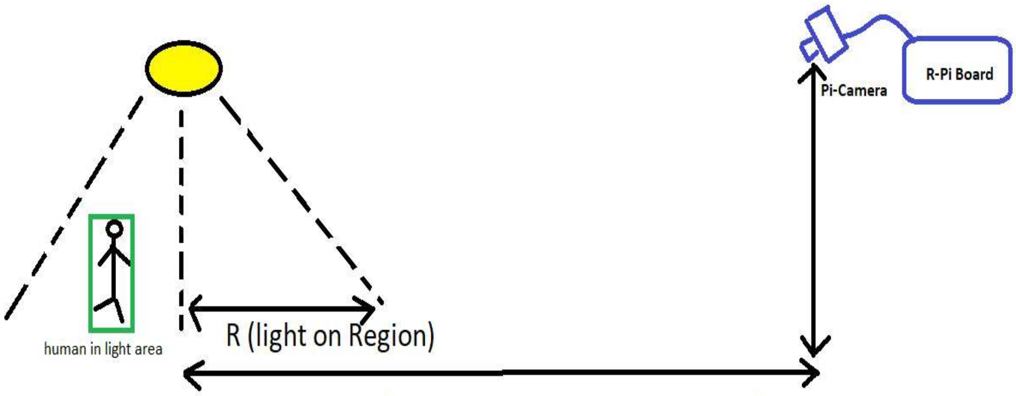


Figure 3.1 A sample of a power management system that covers all buildings area

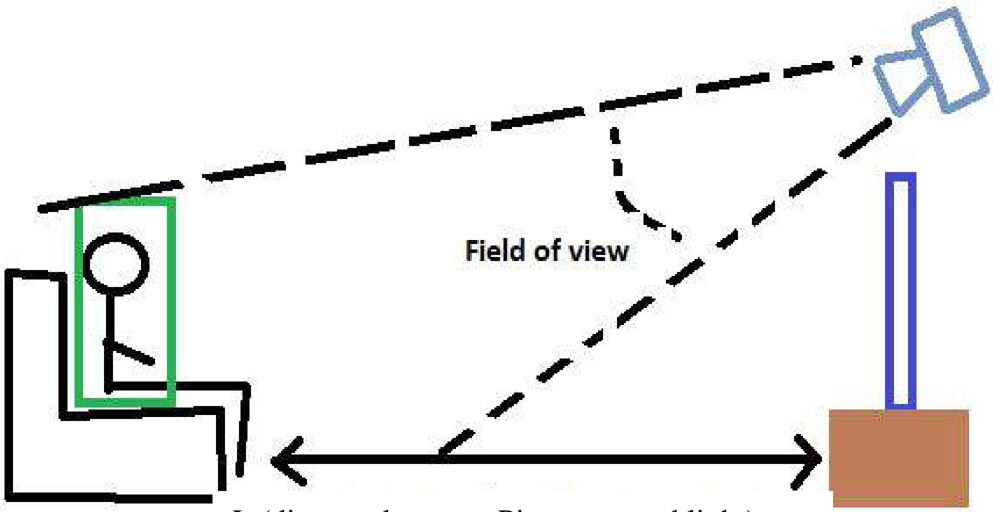
As shown in Figure 3.1, the power management system has multiple sensors distributed to cover all area. These type of sensor used is dependent on section need to control its power, for example light sensor used to control lighting that will switch on light when low light intensity detected, however, after switch lights on the system program start to analyze the area to identify if there is an individual in that area, if it recognized human, the lighting continue to switch on, otherwise the system would switch off lighting. Additionally, in the case of the large-scale room such as halls, the lighting needs to be in multiple groups and the program starts to compute the distance of person/persons from the lighting area. For example, if hall has three lighting groups (A, B and C), in case when the person exists in the area of group (A) then the system will switch on group A lights and switch off lights of groups B and C, in case there are many persons that located in the area of group A and B the system will switch on lights of group A and B and switch off group C. Figure 3.2 illustrated the switching light strategy.



L (distance between Pi camera and light)

Figure 3.2 Switching light methodology by the proposed power management system

The other task that can be achieved by used recognition process is to analyze human activity in order to control switching devices automatically in an intelligent way. For example, the system can predict that the person needs to watch TV, this can be achieved by our proposed method that assumes in case when person exists in hall and he set on sofa or chair with face directed to TV and the distance in desired range then the system is switch on TV and when person left the area the counter start and it then shut down TV after desired time (such as 3 minutes). This strategy can be also serving in some other application such as switching on/off computers, etc. Figure 3.3 illustrated the strategy of switching on/off the TV.



L (distance between Pi camera and light)

Figure 3.3 Switching TV on/off methodology by the proposed system

The other control strategy of the proposed system is to control air conditioner units; this is done by first indicates if there is a person in the room/hall area. In case there is a person, the system will be gathering temperature readings from a temperature sensor in that area, and then it switches on the air conditioner with ability to modify the temperature, mode, and fan direction to satisfying results.

The sensors such as gas sensor and odors can be used from the power management system to control ventilation strategy that will start vent fan. Furthermore, to archives good power consumption some parts of the system such as camera doesn't work until the system detects motion in a specific area then the system will power on all component located in that area to start analyzing the process. Additionally, the proposed system can be operating as a central unit or in dedicate units that work separately for each building section.

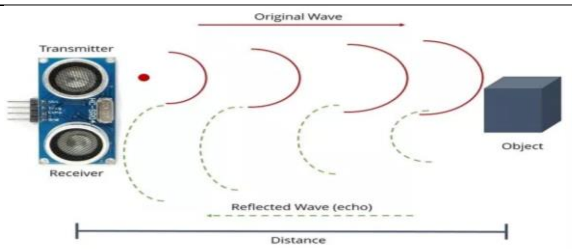
3.3 Hardware Design

The core of the power management hardware system is the Raspberry pi microcontroller unit, there are many versions of the Raspberry pi as shown in table 3.1. As can see in table 3.1, Raspberry Pi 3 B+ (that used in our proposed system) has a fast ARM cortex CPU of about 1.4GHz, 512 GB Ram, Dual Band wireless LAN and Gigabit Ethernet (via USB), in addition to lots of interesting hardware tweaks. The hardware components (sensors) with its configurations are listed as in table 3.2.

Table 3.1 All Raspberry pi versions [18].

Family	Model	Form Factor	Ethernet	Wireless	GPIO	Released	Discontinued
Raspberry Pi 1	B	Standard (85.60 × 56.5 mm)	Yes	No	26-pin	2012	Yes
	A		No			2013	Yes
	B+		Yes		40-pin	2014	
	A+	Compact (65 × 56.5 mm)	No			2014	
Raspberry Pi 2	B	Standard	Yes	No		2015	
Raspberry Pi Zero	Zero	Zero (65 × 30 mm)	No	No		2015	
	W/WH			Yes		2017	
Raspberry Pi 3	B	Standard	Yes	Yes		2016	
	A+	Compact	No			2018	
	B+	Standard	Yes			2018	
Raspberry Pi 4	B (1GB)	Standard	Yes	Yes		2019[25]	
	B (2GB)						
	B (4GB)						

Table 3.2 Hardware components

Hardware Component	**Properties**	**Scheme**
HC-SR04 Ultrasonic Sensor	4 Pins that based on emits ultrasound at 4000 Hz, Power Supply: +5V DC	
DHT11 Humidity / Temperature Sensor	4 Pins single row pin package that needs 5 V to operate	
MQ-4 methane gas sensor	Detect concentrations of natural gas in the range from 200 to 10000 ppm	

The above hardware components have been connected as in the following configuration: The camera has related to Raspberry Pi via onboard CSI camera port. The HC-SR04is connected to Raspberry Pi pins where VCC connected to pin 2, the echo with pin 32 (GPIO 12), GND to pin 34, and trig with pin 36 (GPIO 16). The relay is also connected with Raspberry Pi pins where VCC connected with pin 4, GND with pin 6 and the TV to pin 3 (GPIO 2) and light to pin 5 (GPIO 3). Figure 3.4 shows the proposed system setup and pins connection.

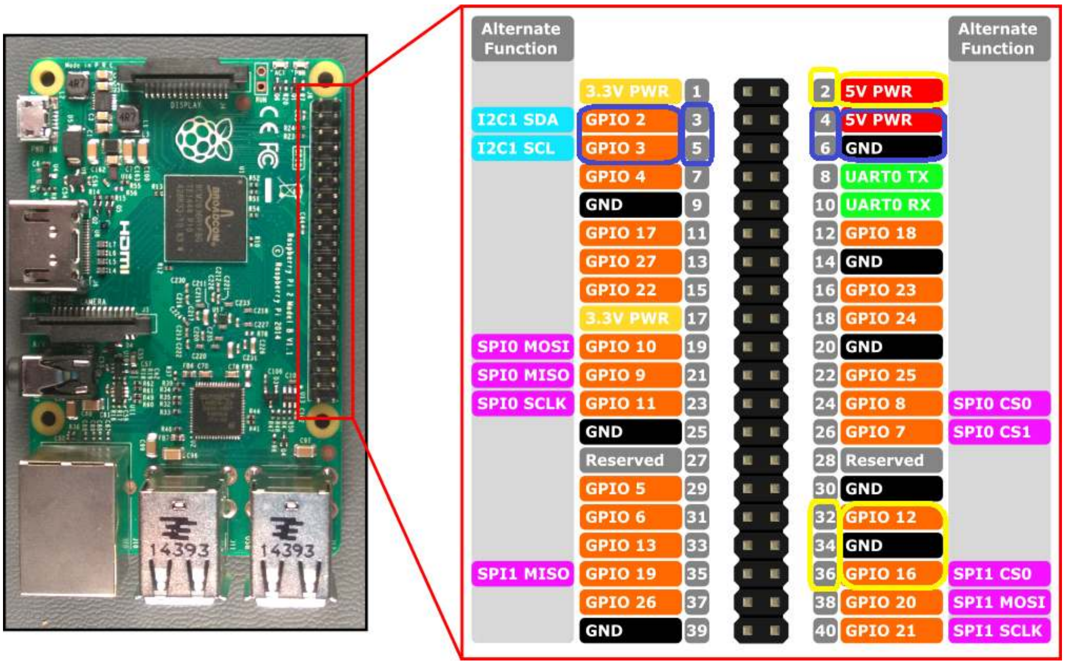


Figure 3.4 Raspberry Pi pins connection

3.4 Software Design

The software includes Raspbian operating system, which is programmed using the Python programming language.

3.4.1 Operation System (OS)

In this work, we have used Raspbian which is the establishment's legitimate upheld working framework. We have utilized Raspbian extend with desktop version 4.14 of size over 4 GB. It comes pre-introduced with a lot of programming for training, programming, and general use. It has Python, Scratch, Sonic Pi, Java, Numerical and the sky is the limit from there.

3.5 AI System Algorithm

The program has been written by Python 3.6 language; it is designed to gather information from all sensors as well as streaming video from Raspberry Pi camera in real time. The proposed system manages the overall processes by utilizing a deep neural algorithm for object classification purpose and has been enhanced by using saliency map algorithm for object detecting purpose. The proposed system can detect some aspects such as existing of human and some object in specific areas such as chair or sofa. It continuously gathers information from different spread sensors. Ultrasonic sensor (HC-SR04) is used to detect any movement, when movement is revealed the system switch all sensors that are located in that area and then employs the ultrasonic sensor to compute distance from the target and give the order to relay the switching on/off depending on human existence and his activity. The proposed method is based on measures saliency in each color space, after that merged them in a final saliency map. For every color channel, at first, the input image is separated to a nonoverlapping patch, and then each one of those patches is symbolized via a coefficients vector of saliency from index tree of patches coming from natural scenes. After that, the two measurements of saliency for both local and global will be calculated and combined to signify saliency of each patch.

The saliency contrast maps are consolidated, and the output can be calculated by the following formula:

$$S_{lg} = \int S(p)P(p|I)dp \tag{3. 1}$$

Where, S_{lg} is the final saliency maps, I is the input image, p is pixel for each patch, $P(p|I)$ is the probability of pixel p for the image.

Local and global saliency maps can be normalized and merged as:

$$S_{lg}{}^{c}(p_i) = N\left(S_l^c(p_i)\right) \text{ o } N\left(S_g^c(p_i)\right) \tag{3. 2}$$

Where, o is an integration structure or a fusion metric between many functions (such as -, +, ∗, min, or max). The values of the saliency $S_{lg}{}^{c}(p_i)$ of an image patch p_i in every channel are normalized and sums again and again to make the saliency of a patch in every color system.

This process will need to repeat the process for every channel for Lab and RGB color systems and merge saliency maps of every sub-channel of a color space to make a saliency map for every color system.

The optimized CNN algorithm that been used in a recognition program has two parts:

3.5.1 Foreground Object Detection [44]

The proposed method of foreground object detection consists of the below steps:

- ***Image Pre-processing:*** In this part, we first converted the colored image into grayscale image then resize the image to fix size with new coordinates and center.
- ***Image Segmentation:*** In this part, we have been segmented image by used SLIC (simple linear iterative clustering) super pixels algorithm based on worked done by Achanta et al [10]. The SLIC algorithm is over-segment the image to many super pixels (patches). Every patch has been represented via a feature vector, and every one of these feature vectors constitutes the feature matrix.
- ***Extract features from the image:*** In this part, an input image is portioned into small and perceptually homogeneous features. We tend to first extract the low-level features that include Gabor filter, steerable pyramids and RGB color to generate a dimension feature description.
- ***Create an index tree:*** Together with super pixels, an index tree is created to encode construction data through hierarchical segmentation. As a result, we initially calculate the appreciation of each surrounding patch by getting the first and second order reachable matrix. After that, we use the algorithm "graph-based image segmentation" to combine spatial neighboring patches based on their affinity. The graph-based image segmentation algorithm generates a granularity-increases segmentations sequence. In every granularity layer, segments are corresponding to nodes in the equivalent layer within the index tree. Especially, the granularity is controlling by an affinity threshold. At last, we get a fine-fine-course hierarchical segmentation from the input image.
- ***Structured matrix decomposition:*** after extracted features and created the index tree, we used “Structured Low-Rank Matrix Factorization” to decompose feature matrix to structured-sparse component and a low-rank component. After collectively imposing the Laplacian regularization and structured-sparsely, the

input feature matrix is decomposed straight to organized components structured-sparse component and a low-rank component.

- ***Post-processing:*** Following decomposing, we proceed with the outcome from the feature to some preprocessing algorithms in order to get improvements for the foreground object. Depending on the structured matrix, we specify the function of straightforward foreground estimation for each patch. After combining all patches together and executing context-based propagation, we obtained the final foreground map of the input images.

3.5.2 Classification using CNN

All models for the experiments in the two objects (human and chair/sofa) were trained using the implementation of the Single Shot Detector (SSD) algorithm. The experiments were performed with various information highlight maps and info picture sizes. Herein, the datasets used are the Pascal Visual Object Classes (Pascal VOC) and Common Objects in Context (COCO) datasets. We had configured a shifted set of default bouncing boxes, of various scales and angle proportions to guarantee most items could be caught. These images set were trained using Stochastic Gradient Descent (SGD) with an initial learning rate of $\eta = 0.01$, $\alpha = 0.9$ momentum, and batch size 32. The image has been resized to 32×32 in both training and recognition process in order to boost the recognition time, while for detecting purpose the image or the video frame size is 300×300×3. The training processes are ran by high-level PC of 3.33 GHz core i7 Extreme CPU with 64GB ram and two Nvidia Quadra. The output weight is then saved and used with a power management program for object detection in Raspberry Pi.

3.5.3 CNN model architecture

Image classification problem is predicting the label of visual object classes. Towards making at an indispensable complex task as cataloging and detection real-life objects, a simple and powerful deep CNN object classification algorithm was proposed which is capable of reaching a better performance while demanding least hardware supported and computational costs.

Basically, the dataset must be split into train and test datasets then the training data is dividing into train and validation samples in order to build the model by assigning the optimum weights and biases. The test dataset is used for assessing the quality of the model through measuring whether objects classes are true. Moreover, it can use many metrics like accuracy, error, etc. to evaluate the designed system.

Our developed CNN model for training the input images is as shown in figure 3.5 below, it has used 4 convolution layers maps with filter size 3×3 and one zero padding, 2 maxpooling layers with pool size 2×2, 3 fully connection layer with 1024, 500 nodes and 10 for last layer.

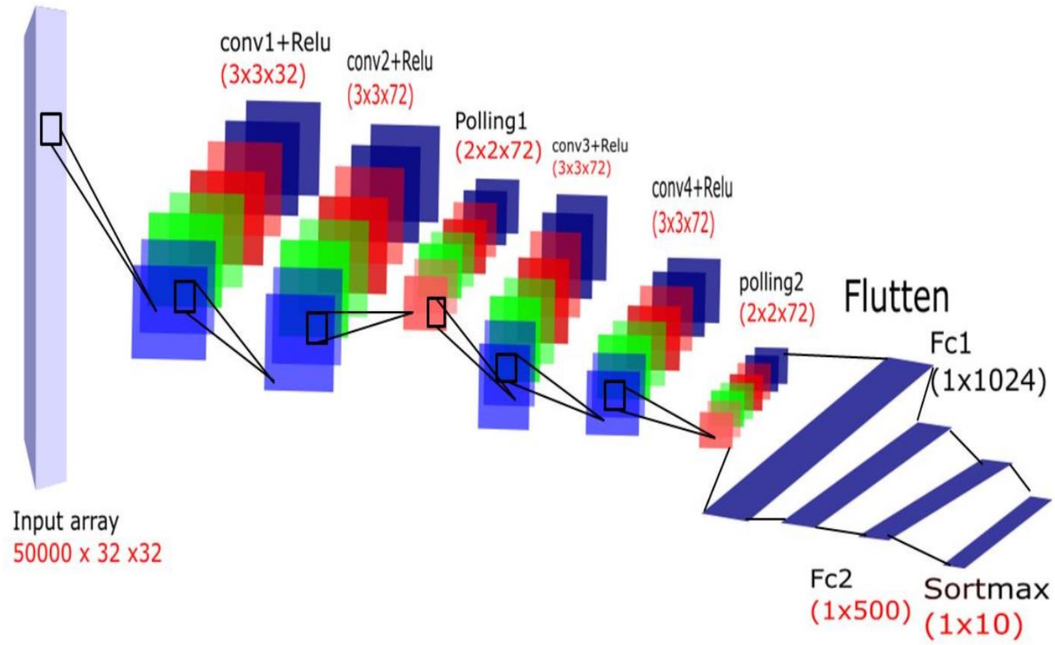


Figure 3.5 The developed CNN model architecture

Table 3.3 CNN trainable parameters

Layers	Activation Shape	Activation Size	Parameters
input	32×32	1024	-
Conv1	32× 32× 32	32768	896
Conv2	32×32×72	73728	20808
Maxpooling1	15× 15×72	16200	0
Conv3	15×15×72	16200	46728
Conv4	13× 13×72	12168	46728
Maxpooling2	6×6×72	2592	0
Flatten	2592	2592	0
FC1	1024	1204	2655232
FC2	500	500	512500
Softmax	10	10	5010

3.6 Dynamic power Allocation and Chain based Routing (DACR) protocol

The periodic data collection from digital sensors such as passive infrared (PIR) sensor and transmission to sink is important process in Wireless Sensor Networks (WSNs) for home/building automation application. As WSNs are resource constrained, it becomes vital challenge to design energy efficient data aggregation and transmission approach especially for the multimedia information, for example, digital pictures and recordings [45]. In this work, a novel resource management and energy aware data transmission algorithms is designed to minimize the energy consumption with optimum quality of service (QoS), it is called Dynamic power Allocation and Chain based Routing (DACR) protocol.

This section presents the design details for the proposed DACR routing protocol model for the real time digital sensor-based applications home/building automation using different types of sensors such as body sensor (for health monitoring), temperature sensor, soil sensor, lightning sensors etc. As we discussed above, the DACR it consists of two contributions to achieve the energy minimization of sensor nodes and quality of service QoS. Figure 3.6 shows the architecture of DACR.

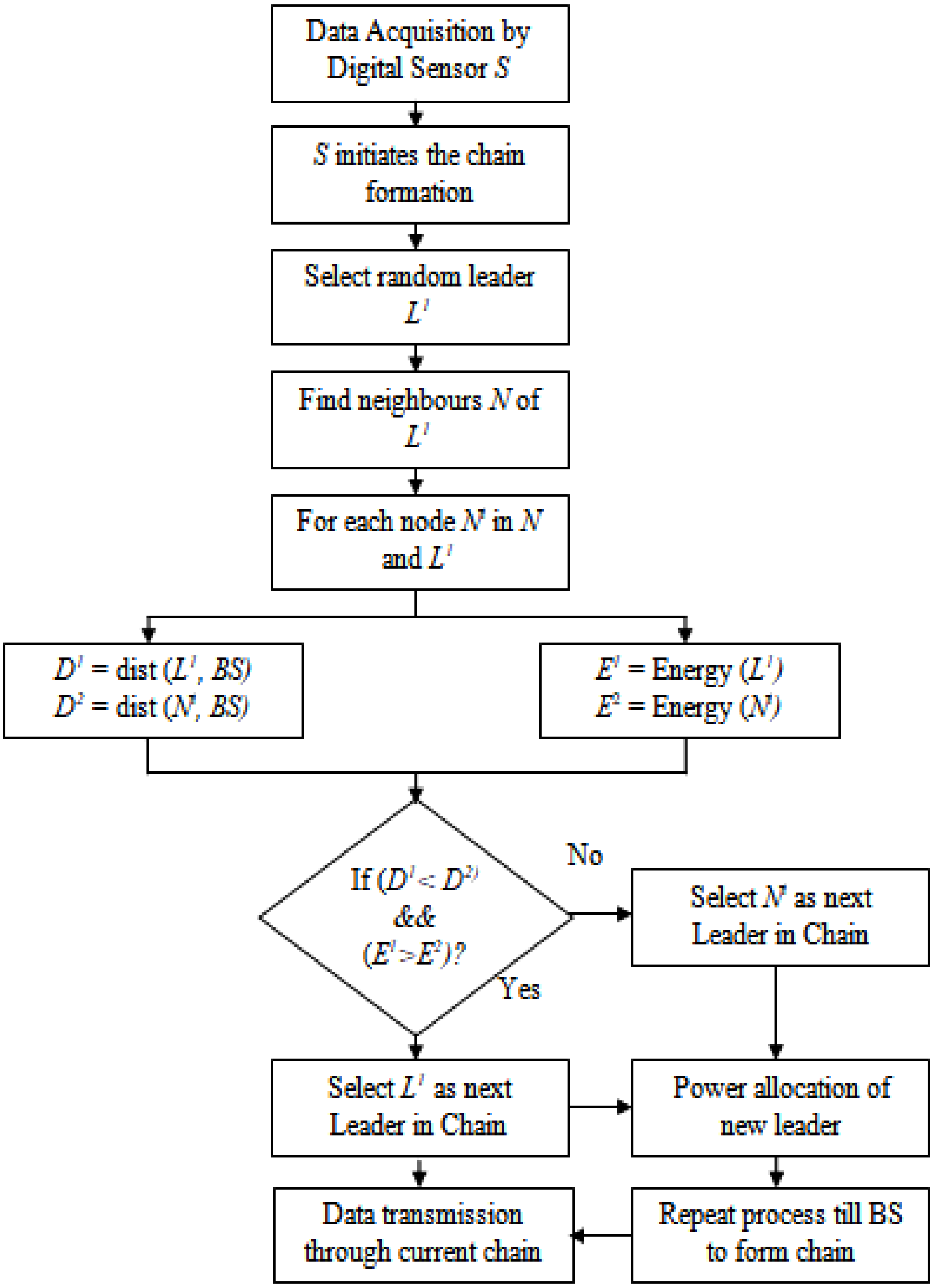


Figure 3.6 System architecture of proposed DACR [45]

As observed in figure 3.6, initially each sensor nodes performs a periodic data acquisition, and then start building the energy chain formation in which the node *S* selects the temporary node as the next hop leader in the chain and then extracts the neighbor. The distance from the base station Bs and remaining energy parameters used to select the next leader from the neighbors set in comparison with current selected leader. After the selection of leader, the dynamic power allocation approach applied to save the energy resources for selected sensor node. The process is repeated till the complete chain from any source sensor node to BS (Sink) node. After the chain formation, data transmission initiated and aggregated at BS node. In below subsections, we first present the methodology designed for the dynamic power allocation of sensor nodes and then algorithm for complete DACR model.

3.6.1 3Dynamic power allocation

Conventionally changing the transmission power manually having worst impacts on QoS and energy performances for the wireless sensor network (WSN) applications. Higher transmission power leads to interference in network and lower transmission power leads to longer delay in communications. Thus, we designed algorithm for dynamic allocation of transmission range to restriction the transmission power of sensor center points using the connection function according to current network topology for home/building automation. Algorithm 1 demonstrates the key steps of the proposed dynamic power allocation approach. The adaptive control of transmission power helps to achieve the efficient energy resource utilization and hence improve the network lifetime.

Algorithm 1: Dynamic power allocation	
Input	
R: current relay/sensor node in chain	
Output	
R^1: node R after newly assigned transmission power	
1.	R computes all its current n number of neighbor nodes in nb set
2.	R broadcasts beacon messages to all nodes innb
3.	For each neighbor nb_i upon receiving beacon message sends
-	ACK packet nb_i^{ack}
-	Number of its neighbor nb_i^{nb}
4.	R computes the mean of its transmission power level using all neighbor outcomes nb_i^{ack} and nb_i^{nb}:
5.	$\mathrm{A^R} = (\sum_{i=1}^{n} \mathrm{nb_i^{nb}})/\,\mathrm{n}$
6.	Node R adjust its current transmission power according the value of $\mathrm{A^R}$
7.	$R^{1=}R$ then connects to number of neighbors' as per the value of $\mathrm{A^R}$
8.	Return R^1

To explain above algorithm, let's suppose the current node is *P* and it communicates a reference point message to its neighbor's hubs (*Q*, *R*, *S* and *T*). Every node replies with ACK and their neighbor node number *P* (*Q*= 2; *R*= 2; S= 2; *T*= 2). In the wake of getting this control data from the neighbor hubs, hub P registers the normal esteem (*2+2+2+2 = 8/4 = 2*). Thus, node *P* allocates is control level to such an extent that it associates just with the mean number of nearest neighbor hubs.

3.6.2 Chain formation

As the standard chain depends bunch course of action method encountering the vitality productivity problem while group improvement and chain advancement, in this study we proposed the chain setup procedure of the regular PEGASIS (routing protocol) as the directing protocol. In chain standard is based on the Ant Colony Optimization (ACO) algorithm by using sensor nodes parameters as separations node from the BS hub. This procedure dependably shapes the ideal chain and consequently ideal pioneer determination for future information transmission. ACO chooses the effective forward subterranean insect hub. After the formation of chain, the backward ant message sent from *BS* to *S*. The process of chain formation and data transmission is showing in algorithm 2.

Algorithm 2: Chain Formation and Transmission

Input

S: source sensor node that collect the periodic data for the transmission

BS: base station node that acts sink to gather the periodic information from S

N: set of sensor nodes

Output

Data transmission and aggregation at *BS*

1. Periodic data acquisition by node *S*
2. *S* broadcast and select the temporary leader node *L*
3. $Ant = 0$
4. N^L = neighbors' ($L \in N$)
5. For each $n \in N^L$ do
6. D^1 = dist (*L, BS*)
7. D^2 = dist (*n, BS*)

```
E1 = L->energy
    E2 = n->energy
    If (D1 < D2 ) && (E1 > E2 )
L->leader
        Apply algorithm 1 on L
        Ant= Ant+1
    Else
n->leader
        Apply algorithm 1 on n
        Ant= Ant+1
    End If
End for
Repeat the process till to the BS node to form the chain.
BS send BAnt to S
After the chain formation, S starts sending data towards BS.
Repeat the process for the next periodic interval.
Stop.
```

In algorithm 2, as described above, every selected leader further analyzed for the dynamic transmission power allocation (Algorithm 1). After the chain formation, the S (broadcast and select the temporary leader node *L*) starts sending the data towards the BS node through chain selected by algorithm 2. For next periodic interval, similar process applied to achieve the vitality balance among the sensor hubs in a network.

CHAPTER FOUR

EXPERIMENTAL RESULTS AND ANALYSIS

4.1 Introduction

In this chapter, the system has been tested in order to investigate the accuracy of the system to track the moving objects and measure the best angle that will produce maximum power. In the experiment, the electrical power generated from the panel that uses proposed tracker which runs in real-time has been monitored. The results have been recorded and compared with systems that have no tracker. Furthermore, the cost of the proposed system can be compared with other commercial trackers that are available in the markets. Figure 4.1 depicts the proposed setup tracker in operation.

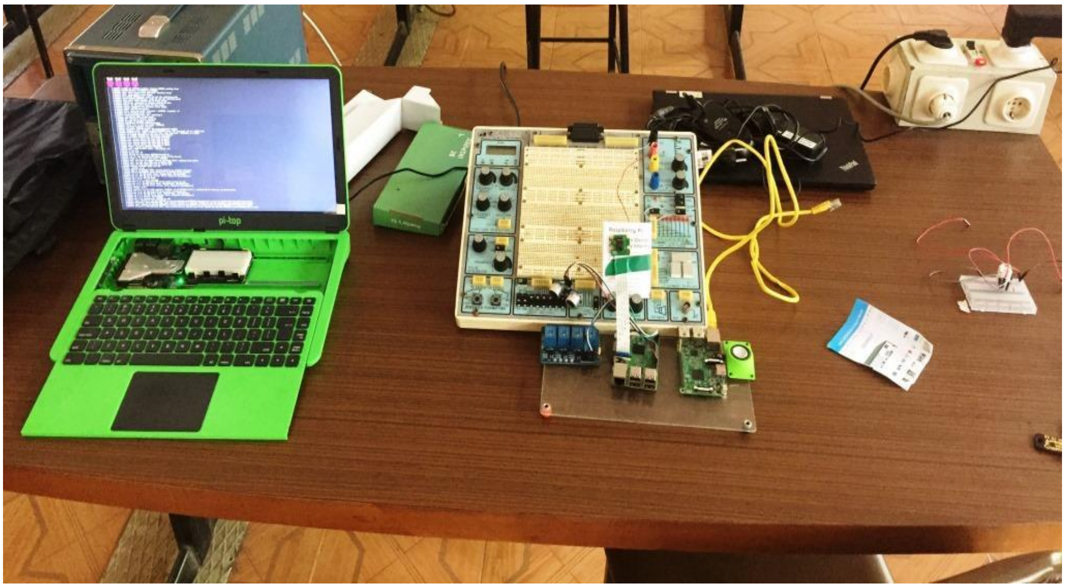


Figure 4.1 The proposed system on operation

4.2 Model Preparation and Training

In this study, two well-known object recognition benchmark datasets are examined and compared with the proposed approach. Pascal Visual Object Classes Challenge 2007 VOC2007) [46, 47] and Common Objects in Context (COCO) [48] .

To train the deep neural network for object detection, a vast amount of data onto an actual scenario is required and true ground-truth annotations. So, for the purpose of object detection problem, the Pascal VOC dataset was used. Images of the dataset were collected from “Flickr "website. It includes 28,952 images with 27,450 annotated regions of interest and contains 20 common class objects and 1 class is used to denote background. The twenty classes are divided between four-sections Person, Animal, Vehicle and Indoor. Although there are many objects in this dataset such as aeroplanes, bicycles, boats, etc.…, we utilised only three objects which are (persons, chairs, and sofas). These objects are concerned with the requirement of our proposed model which needs to classify the person from other environment objects like chair or sofa.

It's proved for classification and detection tasks. The VOC datasets contain some pictures that are taken underneath varied conditions with completely different object size, orientation, pose, lightning conditions, position, and partial occlusion. Each image has annotations XML file giving a bounding box coordination which are placed within the upper-left corner, and object class label for each object inside the image typically more than the object is given in a picture and bounding boxes might overlap. The task of localization can be considered as the problem of identifying the location of a single object among more than an object. To increase the reliability of the Pascal VOC dataset, it contains some challenges such as: some images are of poor quality (e.g. include motion blurring) or of bad illumination (e.g. include objects in silhouette), also some objects are ‘occluded’ by close-fitting things e.g. clothing, mud, snow, etc …, the images allow viewpoints within 10-20 degrees.

The proposed method is also used to train the Common Objects in Context (COCO) data. Images of the COCO dataset were collected from Microsoft. It includes more than 200000 images with 91 objects.

4.2.1 Datasets preparation and details

First, Pascal VOC and COCO datasets are loaded from suitable repository then the images are reduced to size 32×32. After that, we’ve choice randomly 60000 images among the total images with 10 classes only and divided them into training with 50000 samples and testing with 10000 samples. The class label for each class can be identified properly by converting the class vector into binary class via using suitable encoding and then normalization images to float in one scale to reduce training time. The optimizer on a cross entropy loss function after softmax activation with a batch size of 32. The initialization of the weights is done with the Glorot initialization method which is a common initialization scheme for deep neural networks and python environment.

4.2.2 The proposed method training and evaluating

The model has ten classes, it was trained twice times for different filter sizes (3×3) and (5×5) for 10 and 95 epochs. Table 4.1 shows the model accuracy and loss with different iterations (epochs). The best result was with 3×3 window run for 95 epochs.

Table 4.1 Results of learning the model

Filter size	Iteration epochs	Validation accuracy %	Test accuracy %	Error rate %	Training time (min)
5×5	10	48.3	57.3	42.7	21
3×3	10	59.2	58.8	41.2	34
3×3	95	**74.7**	**77.0**	**23**	125

The accuracy and error curves for 5x5 filter map and ten epochs with no overfitting are appeared in figure 4.2. While the accuracy and error curves for 3x3 filter map and 10 epochs are illustrated in figures 4.3. In the same context, the accuracy and error curves for 3x3 filter map and 95 epochs in figure 4.4.

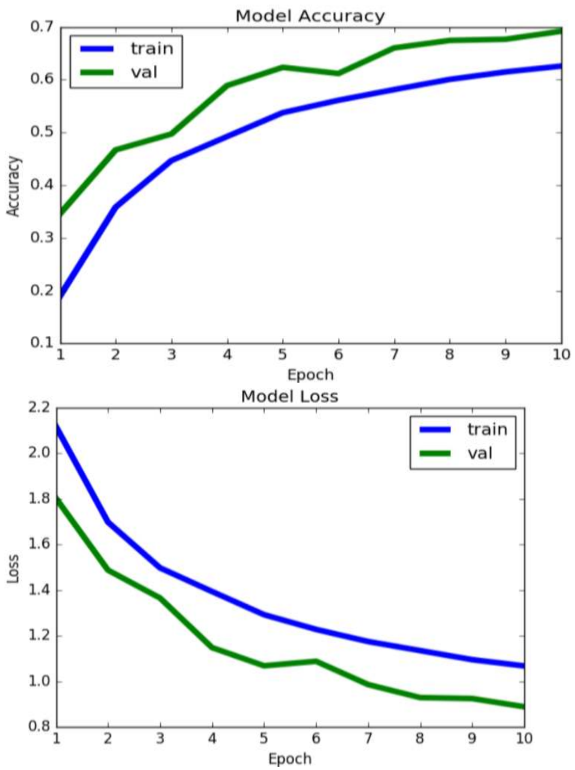


Figure 4.2. The model accuracy and loss with 5x5 filter map and 10 epochs

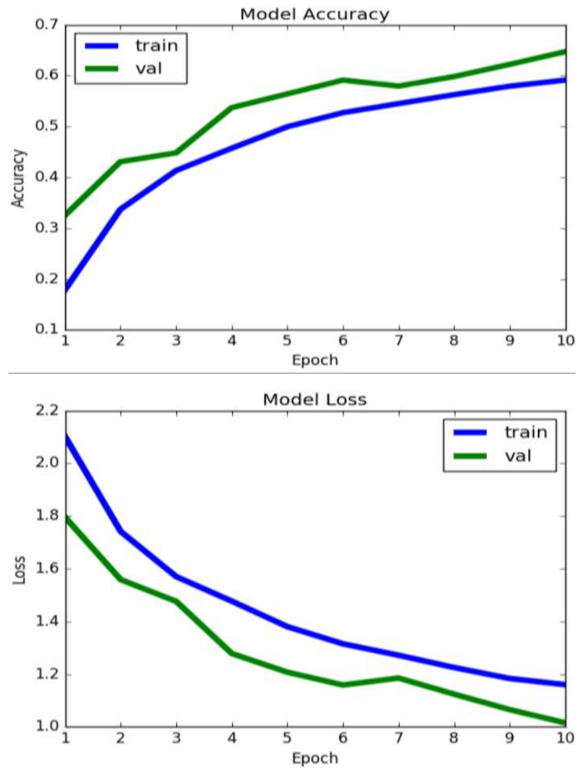


Figure 4.3 The model accuracy and loss with 3x3 filter map and 10 epochs

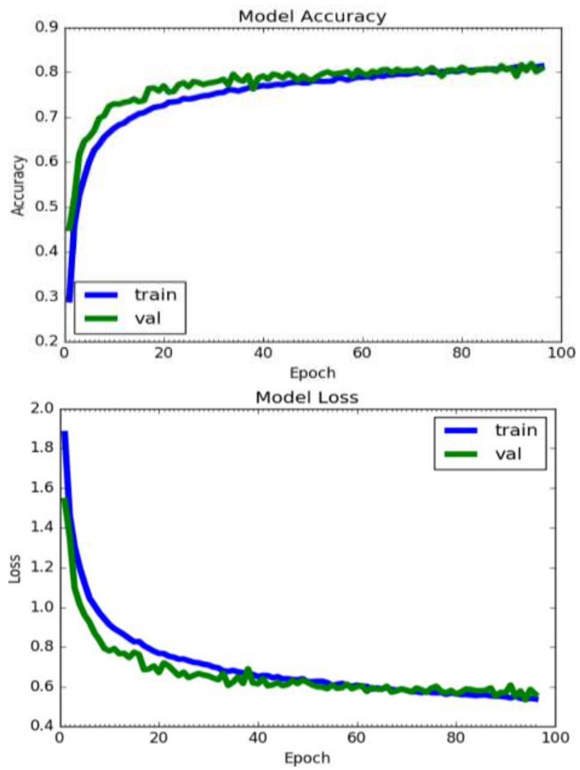


Figure 4.4 The model accuracy and loss with 3x3 filter map and 95 epochs

The proposed CNN algorithm can recognise 10 classes (ten objects). Certainly, increasing classes and decreasing the iterations (epochs) that are necessary to minimize the error lead to decrease the classification precision. Besides that, the real time smart PMS doesn't need to recognize un-useful objects like aero planes, bicycles, boats, etc. in home or building, as a result, it can summarize the substantial objects to the smart PMS with the human and simple chair or sofa.

To minimize the memory footprint and training time, the training set images were pre-processed to create labels with single annotations. Moreover, to increase the classification accuracy, only three classes were used (person, chair, sofa). There should be a save for interested annotations that containing a single object. The Pascal VOC and COCO datasets that are used in the proposed model are explained as in tables 4.2 and 4.3.

Table 4.2 The detail of the training Pascal VOC dataset

Object	Training images	Validating Images
Chair	224	221
Sofa	100	80
Person	1025	750

For the training and validating purpose, the total used images=2400, each image requires 0.9 second, the total VOC training data time=36 minutes.

Table 4.3 The detail of the training COCO dataset

Object	Training images	Validating Images
Chair	300	200
Sofa	200	100
Person	2000	1400

For the training and validating purpose, the total images=4200, each image requires 0.9 second, the total COCO training data time=63 minutes. So, the total training time for both VOC and COCO is approximately 99 minutes = 1.65 hour.

The proposed model can achieve 73.2% mAP for Pascal VOC 2007 validation set using Pascal VOC 2007 training set. On the harder MS COCO dataset, they provide the result of 68.9% mAP for COCO validation set using COCO training set where the mAP metric means (Mean Average Precision).

4.3 Measurements and Performance Metrics

The total classification performance can be calculated by comparing expected target labels with predicted target class labels. The detailed results can be observed from the confusion matrices whose each row represents the samples in an expected class and each column represents the samples in predicted classes. To express classification performances of each class in dataset, the precision and recall criteria will be examined for proposed methods. Precision can be defined as the fraction of positive results that are correctly identified among the positive predictions while recall is fraction of positive results that are correctly identified among the real positive values.

$$\begin{aligned} Recall &= \frac{TP}{TP+FN} \\ Precision &= \frac{TP}{TP+FP} \end{aligned} \tag{4.1}$$

where TP and FP stand for correctly labeled positive samples, and wrongly labeled positive samples respectively. FN symbolizes wrongly labeled negative observations. Besides the classification performances of the proposed and benchmark methods, the computational time costs will be analyzed in this thesis. Also, The F-measurement (F-measure) is described as a harmonic mean of two other classification metrics; precision and recall as:

$$F - measure = \frac{Recall \times Precision}{Recall+Precision} \tag{4.2}$$

The overall accuracy and error are the measure to show the classifier's ability of correctly predicting the given samples in overall the data and defined as:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \qquad (4.3)$$
$$Error = 1 - Accuracy$$

In order to exactly evaluate the accuracy and other performance metrics of an object recognition algorithm on a dataset, we exploit the k-fold validation process, i.e., apply input data into the system with different and random permutations taken without replacement and then compute the average response. Firstly, from the Pascal VOC and COCO datasets, we choose randomly 920 images, 340 images for the person and 580 images for chair & sofa. Secondly, these images are applied to the proposed system with 10 random permutations to calculate the accuracy of each time. Finally, we compute the average predication of the persons and chair & sofa as in table 4.4 below:

Table 4.4 A confusion matrix and other performance metrics for the classification results

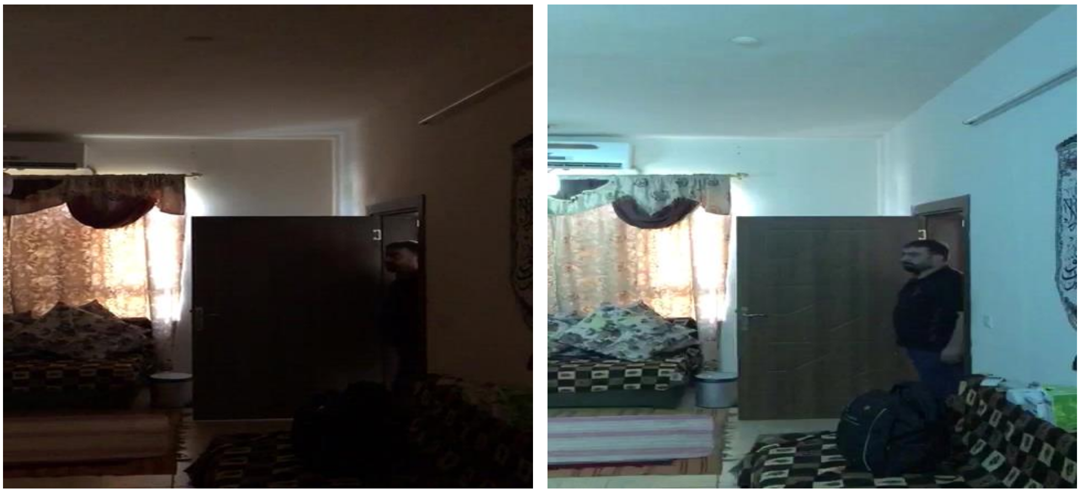
	Person predicated	chair & sofa predicated	**Recall**	**F-measure**
1 (Actual)	317	54	0.854	0.891
0 (Actual)	23	526	0.958	0.931
Precision	0.932	0.906		

Overall Accuracy	**Overall Error**	**Correctly Classified**	**Incorrectly Classified**
0.916	0.083	843	77

There should be a determination of how well the model predicted the location of the object typically.

4.4 Testing of the Proposed Algorithm

The function of the power management system is to manage and measure all operations in real-time conditions. For this purpose, we instill the smart power management system in the building (apartment) that has 2 rooms, 1 hall, 1 bathroom. The sensors (light, temperature/humidity, odors and ultrasonic) were totally separated into two isolate lighting groups A and B, then placed in the desired locations within buildings area, where each room and hall have three sensors (light, temperature/humidity and ultrasonic) sensors, the bathroom has three sensors (light, ultrasonic and odors) sensors. And we placed a camera and processing unit in the hall. In addition, we connected the TV to a switch that is connected to the system via relay as well as a vent fan to control switching process. At first, all body left the apartment from the main door, then the power management system successfully turns off the lights and all connected devices. When the door opens the system detects motion and switches on lights in the corridors. When any one entering the first room, the system detects the motion and switches on the lights. In the first room the temperature sensor starts measuring the room’s temperature and then switches on the air-conditioner unit. When the second person enters the second room, the system switches on the lights and air-conditioner unit too. Figure 4.5 shows the operation process when a person enters and left the room.



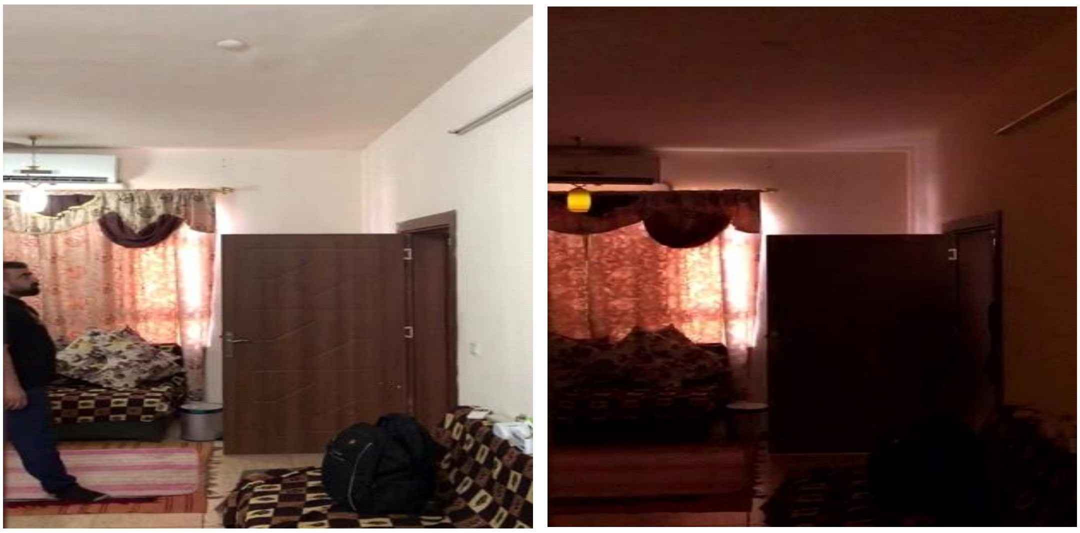


Figure 4.5 Light on/off operation based on human detection and distance measurement

In the hall, same as in room when a person enters the hall, the system detects motion via ultrasonic sensor and directly switches on lights of groups A and B. The system also starts gathering temperature from sensor and switches on air conditioning unit. In the same time the system powers on the camera and recognizes the person at the same time the ultrasonic sensor starts computing the person's location in the room. When a person sits on the sofa that is in front of the TV, the person becomes region A, and the system switches off group B lights, in the same time the system analyzes the hall object and when it successfully detects the person and the sofa, it switches the TV on. Figures 4.6 and 4.7 shows the detection process, recognition accuracy and switching operation.

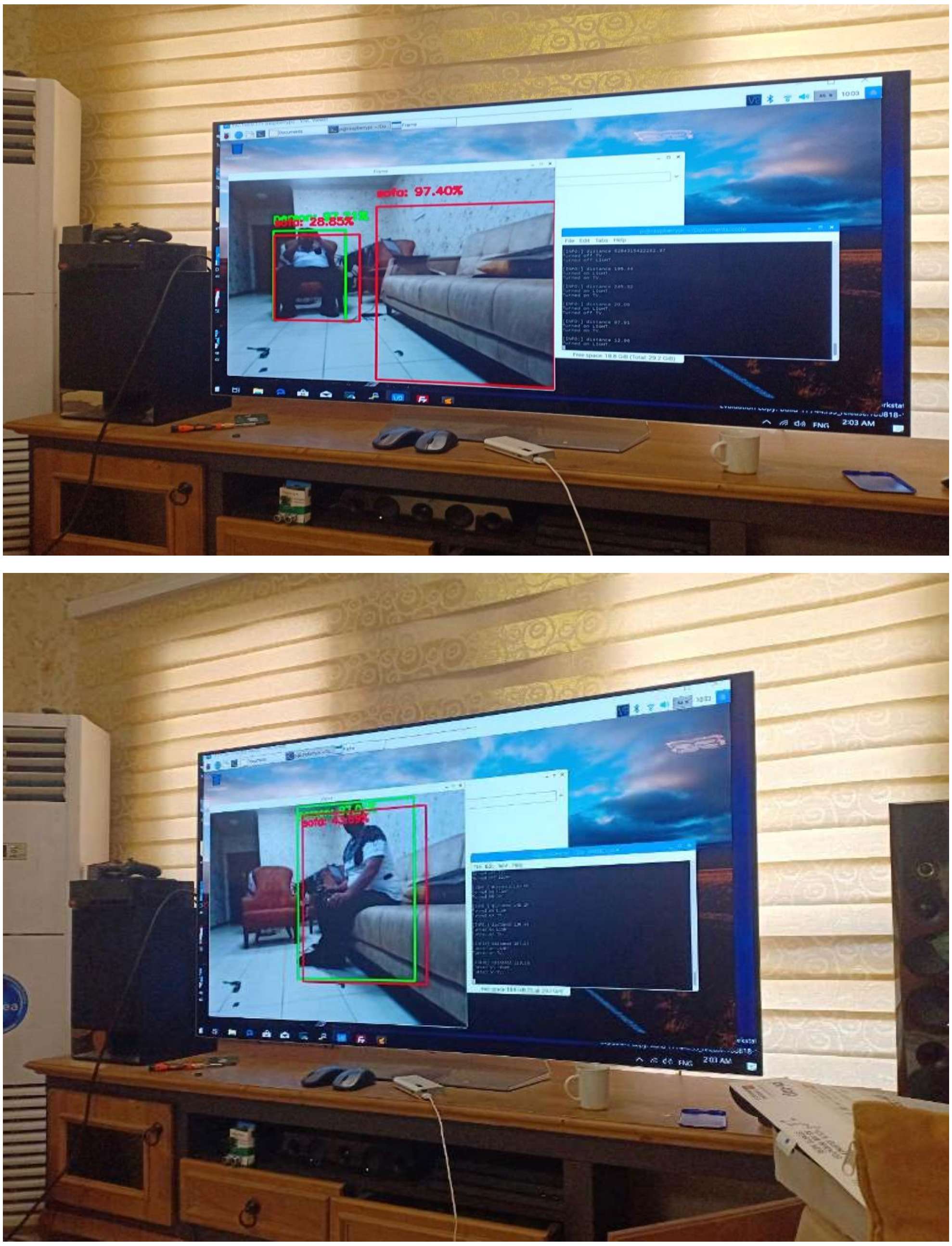


Figure 4.6 Human and chair/sofa recognition when human set on chair and sofa

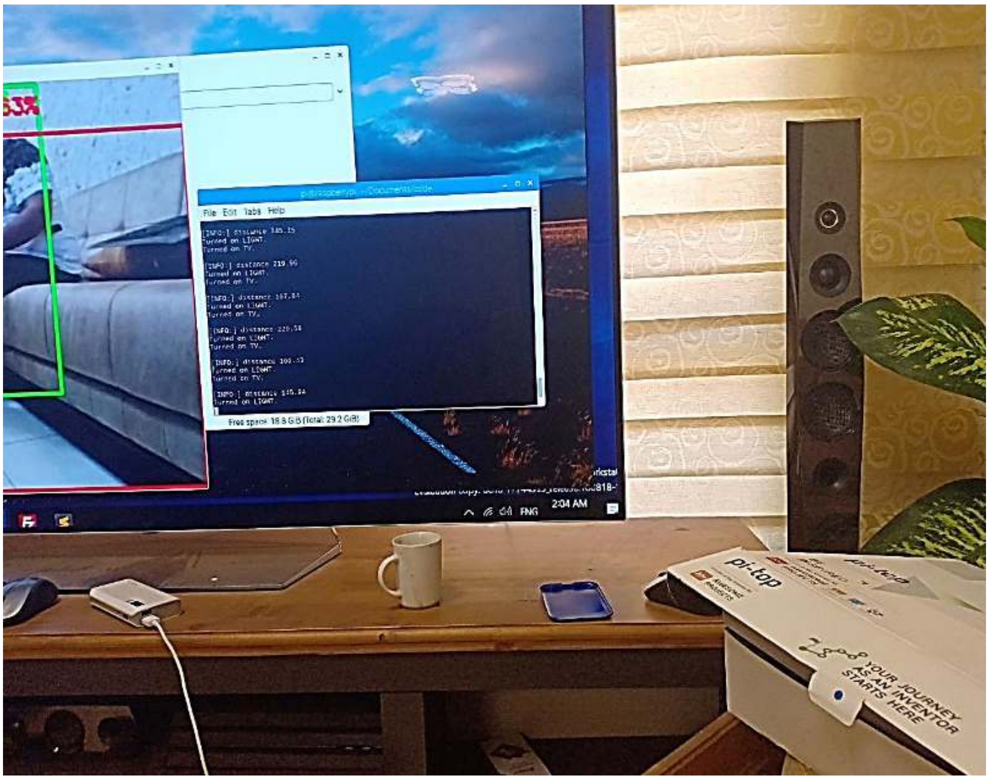


Figure 4.7 Human distance detection when moving in the room and switching operation

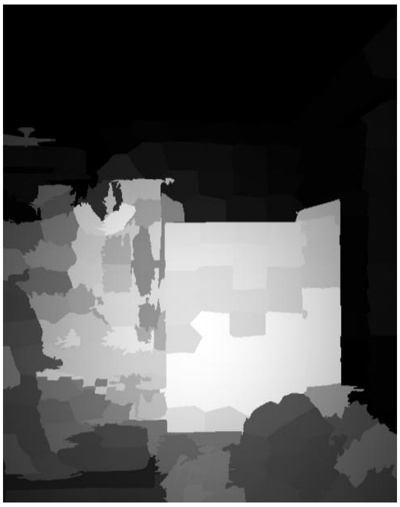
As shown in figure 4.5, we have tested the operation when the human enters room. The system is successfully detecting the human entering the room and isolates it from another object in the room. The system computes the distance between the human and sensor, switches on the light. Figure 4.6 shows the test of TV switching on/off process. At first, the system detects the chair in the room and compute a distance between chair and TV, and if there is no human presence, then the system stays switch TV off. When the human is within the room it detects the human activity, if the human sits on the chair or sofa then it will switch the TV on, and if he stands up, within one minute, the system will switch-off the TV. Figure 4.7 shows the human distance calculation and switching operation as a human move over the room and sits in a chair or sofa. The results show that the precision of revelation is very high and it detects the human and the chair/sofa with 98% maximum accuracy from the other room objects. In addition, the accuracy of recognition of the separated object reaches up to 97% and in relatively good response time about 0.9 seconds. However, in some cases the recognition accuracy drops to 27% especially when the simulation mixed between human and chair (e.g., human is not clear and is sitting on the chair). The ultrasonic also works well and detects distance but some it produces value with some errors. The operation of switching light and TV works well and it runs in fast and smoothens.

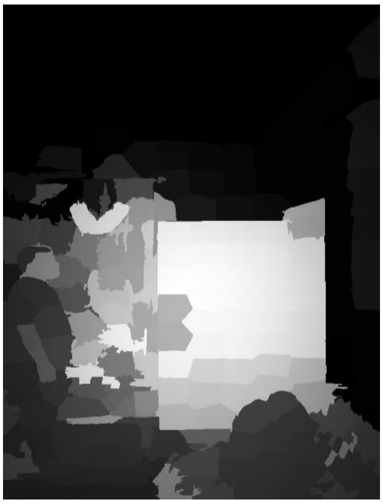
The high accuracy of recognition as well as fast processing speed is related to utilized the saliency algorithm with CNN. The main benefit from employing the saliency algorithm is the robust extracting of the most significant features of the objects from sense. It successfully segments these objects with good details that progress the presentation of the neural network in addition to reduce the needing of high quality and as a result it can reduce computational requirements and increase training and recognition speeds. Figures 4.8 (a, b, &c) show the feature extracted from high resolution input images that have many objects where each input image is colored with 300×300×3. The results showed that this method efficiently removes foreground object from other objects in the sense with high details that can be detected easily and accurately with many recognition techniques.


(a)


(b)


(c)


(a)


(b)

(c)

Figure 4.8 Feature extractions and segmentation using saliency objects for many cases

This part shows the accuracy gain when applied saliency method to generate saliency map that used as input for CNN to detect related objects.

Figures 4.9, 4.10 and 4.11 show the saliency map generated from a live streaming video with different distance (far, near, and very near), the video stream is with 25 fps (frame per second), instead of operating with the whole video, we always process the mid frame (frame 13) as an image comparable to the whole video.

Figure 4.9 Isolation results after applied saliency map at real-time streaming (Human in deep of view)

Figure 4.10 Isolation results after applied saliency map at real-time streaming (Human in near distance of view)

Figure 4.11 Isolation results after applied saliency map at real-time streaming (Some part of Human appeared in view)

As shown from figures (4.9-4.11), the human body is effectively detached from the environment even with different situations. As a result, it can be easily processed and detected by the CNN. For recognition the saliency map features are then passed to the CNN to recognize the human body, the whole recognition steps are shown in figure 4.12.

The recognition results show that CNN has successfully recognized people existing in real time with high accuracy reach of 97% tracking success in good time reaches 0.9 second. This high accuracy with 98% detection is related to best feature isolation of human body by used saliency map.

Figure 4.12 CNN recognition results

4.5 DACR Performance Evaluation

This section described the test examination and evaluations of the DACR protocol with the state-of-art WSN protocols [49-56]. The performance of DACR has been evaluated with recent clustering-based routing protocols QoS-aware and heterogeneously clustered routing (QHCR) [57], Energy-Efficient Heterogeneous Ring Clustering Routing Protocol (E2HRC) [58] and conventional PEGASIS protocol. As the DACR is mainly designed for the applications related to building automation, I designed the different WSNs with varying the number of sensor nodes (30 to 180) to check the reliability on small to large buildings. I constructed 30 sensor hubs and placed them in a single building but placed in random locations of the same building that is, rooms, corridors, and holes. I implemented and evaluated above routing methods using a network virtual reality program called (NS2). The sensor nodes are deployed into 100×100 square area of building. Table 4.5 illustrates the Parameters of the designed network or the specifications that can affect the WSN performance for all routing methods.

Table 4.5 Parameters of network design

Number of sensor nodes	30-180
Number of CBR Traffic	5
Simulation Time	200 second
Packet size (Kbps)	512
Routing Protocols	DACR, QHCR, E2HRC, PEGASIS
MAC	IEEE 802.11
Propagation Model	Two-Ray Ground
Antenna	Omni Antenna
Initial energy (J)	30 nj
Tx Energy consumption	1 nj/s
Rx Energy consumption	1 nj/s

The four vital key parameters for the WSN are: average energy consumption, network lifetime, average throughput, and average delay which are calculated using the concepts and equations below.

The average energy (E) consumption is measured as:

$$\mathrm{E} = \frac{\left(\sum_{k=0}^{N} I^k - R^k\right)}{N} \tag{4.4}$$

Where, N is total number of sensor nodes in network, I^k is the initial energy of k^{th} node, and R^k is the remaining energy of k^{th} node.

The network lifetime (NL) can be calculated depending on the total measured remaining energy as:

$$\mathrm{NL} = \frac{\left(\sum_{k=0}^{N} R^k\right)}{N} \tag{4.5}$$

Average throughput (T) is computed as:

$$T = S^n . S^s . 8 - \mathrm{A} \tag{4.6}$$

Where, S^n and S^s are sequence number and segment size respectively and A represents the current active duration to compute the average throughout.

Finally, the average delay (D) is manipulated as:

$$D = N.[D^T + D^P + D^C + D^Q] \tag{4.7}$$

Where D^T is transmission delays, D^P is propagation delay, D^C processing delay, D^Q is Queuing delay, and N is number of links (Number of routers - 1).

We measured the above parameters for WSNs using different routing methods. Based on the measurements, we can represent the comparative results as described below.

The average energy consumption analysis presented in figure 4.13 with readings in table 4.6. The proposed DACR protocol shows the significant reduction in energy consumption compared to state-of-art methods for every size of building application. The energy consumption minimized in DACR because of methods designed for dynamic power allocation which uses the resources efficiently to save the excessive energy consumption as compared to other protocols. Further the chain formation process is very simple and depends on remaining energy parameter and distance factor. The distance parameter cut down the transmission distance and hence minimizes the energy resources as well as reduced the number of the message exchanged. The PEGASIS routing protocol shows the highest energy consumption and less network lifetime performance.

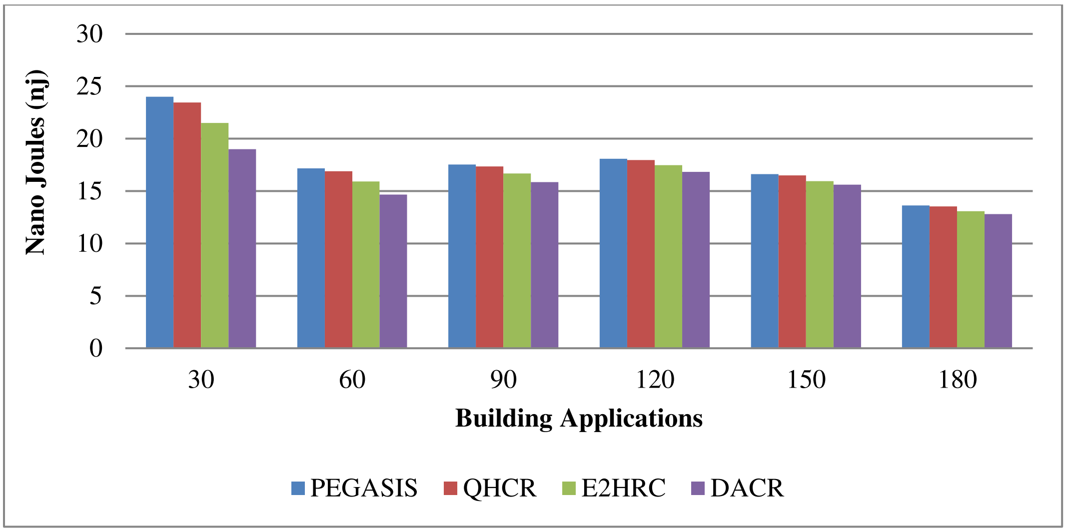


Figure 4.13 Performance investigation of average energy consumption

Table 4.6 Average Energy Consumption Analysis

Density	**PEGASIS**	**QHCR**	**E2HRC**	**DACR**
30	23.99	23.46	21.49	**18.99**
60	17.1645	16.89	15.91	**14.66**
90	17.52	17.34	16.69	**15.85**
120	18.09	17.96	17.46	**16.84**
150	16.62	16.51	15.95	**15.62**
180	13.63	13.54	13.08	**12.8**

The results of average energy consumption reflect the overall network lifetime performance as well with obvious reasons described above.

Figure 4.14 and table 4.7 demonstrate the network lifetime performance results for which the average energy consumption and residual parameters utilized to estimate the number rounds still the first node dies. Among the E2HRC and QHCR, the former shows the better energy efficiency compared to later. The proposed DACR improves the network lifetime due to the lightweight process and optimized resource allocation strategies designed in this work.

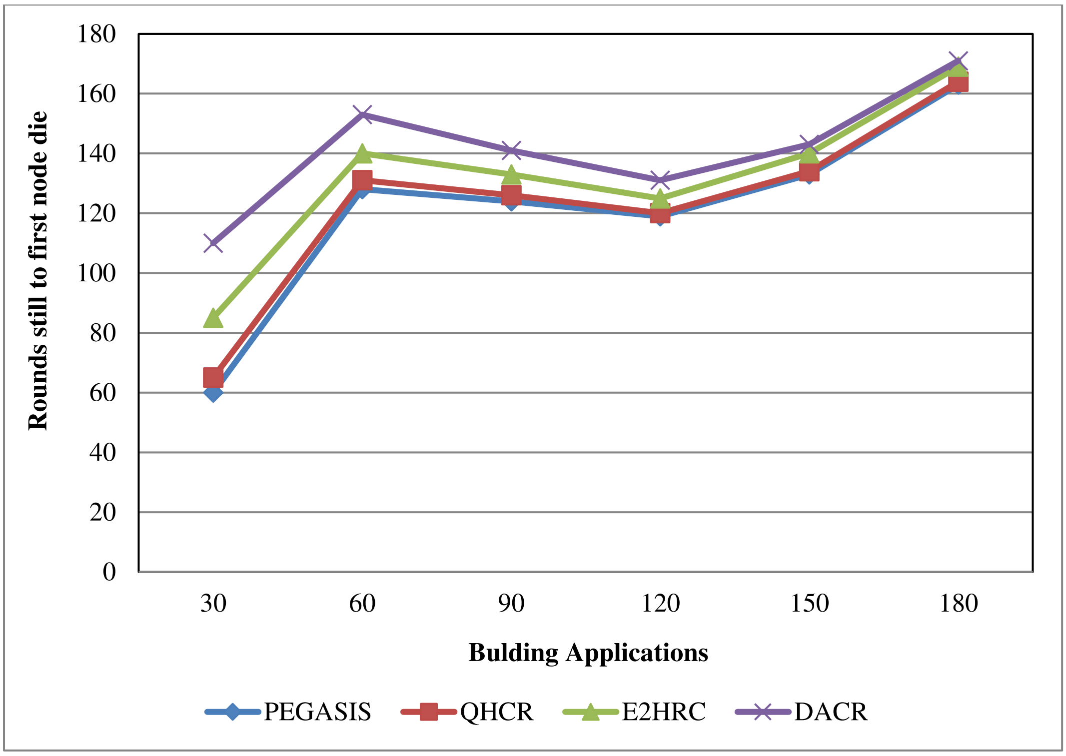


Figure 4.14 Performance investigation of network lifetime

Table 4.7 Network Lifetime Analysis

Density	PEGASIS	QHCR	E2HRC	DACR
30	60	65	85	**110**
60	128	131	140	**153**
90	124	126	133	**141**
120	119	120	125	**131**
150	133	134	140	**143**
180	163	164	169	**171**

Table 4.7 demonstrates the network lifetime performance results for which the average energy consumption and residual parameters utilized to estimate the number rounds still the first node dies. Among the E2HRC and QHCR, the proposed protocol achieved the energy efficiency; the next challenge is to accomplish the exchange off between energy efficiency and QoS execution. The QoS execution is supported utilizing the two core parameters for the most part, for example, normal throughput and normal start to finish delay.

Figure 4.15 (table 4.8) demonstrates the average throughput results for each size of networks. The throughput is measured in KBPS. From the results its shows that DACR protocol enhanced the throughput performance compared to other investigated protocols. PEGSIS showing the poor throughput performance and E3HRC shows the second-best performance. The DACR achieved the highest throughput due to geographical short chain formation to transmit data. The leaders are selected using distance and remaining energy parameters which leads to performance improvement in DACR protocol.

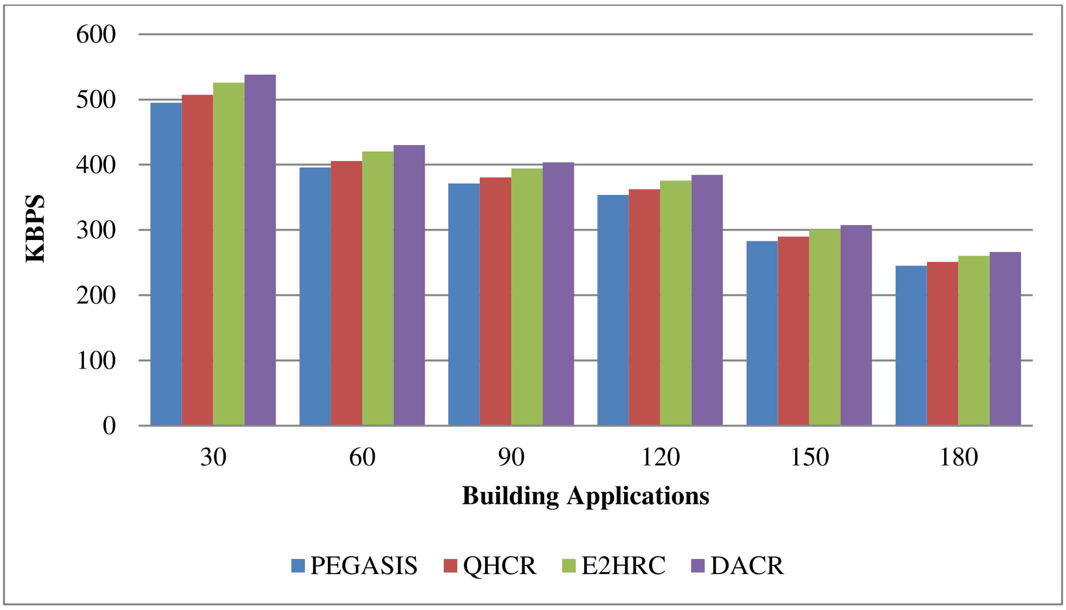


Figure 4.15 Performance investigation of average throughput performance

Table 4.8 Average Throughput Analysis

Node Density	PEGASIS	QHCR	E2HRC	DACR
30	494.93	507.3	525.86	**538.23**
60	395.72	405.61	420.45	**430.35**
90	371.1	380.38	394.29	**403.57**
120	353.46	362.29	375.55	**384.38**
150	282.9	289.97	300.58	**307.65**
180	244.99	251.11	260.3	**266.42**

The next leaders selected based on remaining energy and distance parameters, the delay parameter also minimized for DACR protocol. The distance parameter is not utilized in existing PEGASIS and recent clustering protocols for the data transmission process. The delay performance shows that all the data transmission takes less time due to cut down in total transmission distance in DACR routing protocol.

Figure 4.16 (table 4.9) demonstrates the outcome of average delay performance.

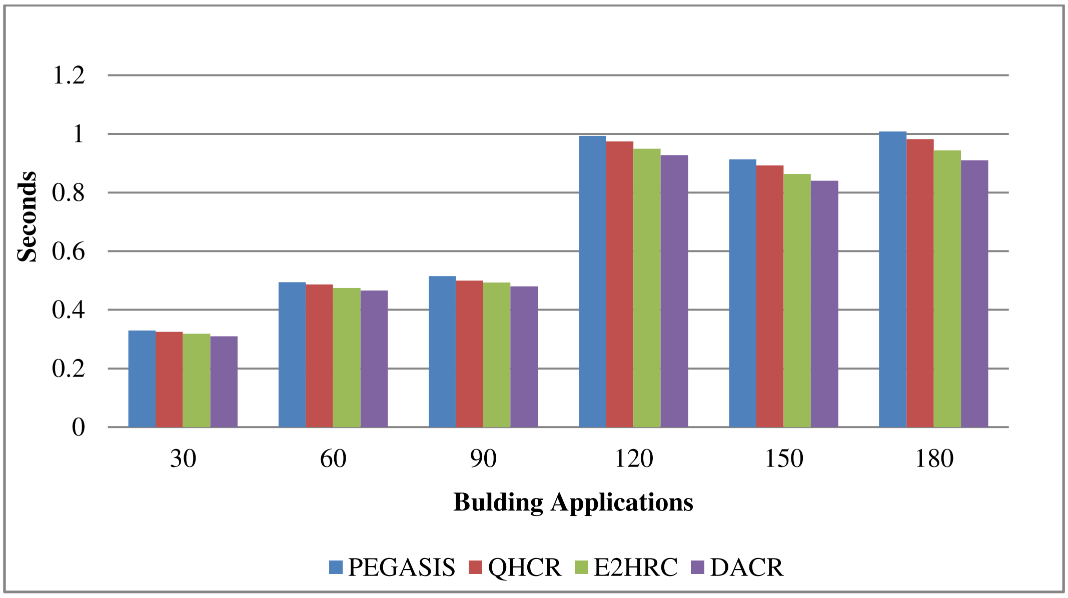


Figure 4.16 Performance investigation of average delay performance

Table 4.9 Average delay Analysis

Node Density	PEGASIS	QHCR	E2HRC	DACR
30	0.3296	0.3253	0.319	**0.31**
60	0.4944	0.4863	0.4747	**0.4655**
90	0.515	0.5	0.4932	**0.48**
120	0.9929	0.9748	0.9489	**0.928**
150	0.9133	0.8926	0.8633	**0.84**
180	1.0081	0.9818	0.944	**0.91**

Table 4.10 illustrates the average performance computed based on above parameters so as to legitimize the general examination of the proposed protocol. From the overall results, the DACR reduces the start to finish postponement and vitality utilization, improve the throughput and system lifetime performances. This also means that DACR operates with minimum steering overhead as thought about all other researched directing methods for different size of building/home networks. The achieved simulation results also justified that DACR achieved the efficient space and time complexities over the state-of-art methods.

Table 4.10 Average performances investigations

Parameters	**PEGASIS**	**QHCR**	**E2HRC**	**DACR**
Avg. Delay	0.6594	0.6446	0.626	0.609
Avg. Throughput	357.18	366.11	379.5	388.43
Avg. Energy Consumption	17.83	17.61	16.76	15.79
Network Lifetime	121	123	132	141

CHAPTER FIVE

CONCLUSION AND FUTURE WORK

5.1 Conclusion

In this work, we have developed a smart power management system that utilised low-cost microcomputer unit with neural network, image processing, and deep learning in order to control power consumption in the buildings and household. Low- power and cost components have been employed to design such system, in which, a single board computer that is Raspberry Pi has been used as a control and processing unit. To serve the power management strategies that include multiple operations like motion detection, recognition, device switching, and controlling, localized multiple sensors have been also adopted, such as lighting sensor to detect lighting intensity, temperature/humidity sensor to detect room temperature, odors sensor to detect bad odors, ultrasonic sensor to detect motion as well as determine person distance.

The real operation testing results have been proved that the entire system can be worked smoothly and efficiently and it can successfully control different tasks at the same time without any failure. In the other hand, the recognition process has been running in a fast and accurate way in order to manage the power system such as switching the device on-off.

The artificial intelligence part of the proposed system which including deep learning with the convolutional neural network is more efficient than other traditional approaches in accuracy and taking the decision. The main difference between both approaches is based on the tracking moving object that can switch the device on-off even with any moving object such as animals or flying objects. The proposed system can detect human and some other objects such as chair and sofa, also it can recognize some of human activities specially when the human is sitting on the chair. The other significant improvements of the proposed methods are:

Firstly, even we have been used a low computational computer unit (Raspberry Pi), it can distinguish a human with high accuracy reach to 97% maximum and fast processing time within 0.9 second. Secondly, human detection and recognition approaches are supported by yielding two effective methods: first is the feature extraction method that depends on the

theory of object detection-based attention (saliency process) and computes foreground of segments for an image. This model can automatically detect the distribution of foreground objects that can be found in the connected regions. Compared, our method doesn't depend on any learning; therefore, it doesn't need image retrieval and annotation. Second, we wholly depend on a shifted set of default bouncing boxes, of various scales and angle proportions to guarantee most items could be caught.

The experimental results have been showed that the system can run the power management system robustly, control and switch all sensors in real time and it is relatively got a fast processing time which gives the ability to use the Raspberry Pi as a defecated processing unit that makes the power management system low cost and practically useful.

5.2 Suggestion for Future Works

Although the current work was attempted to entirely cover and implement different improvements to the PMS system including less human intervention, high computational scalability, low error in learning rate, and a good generalization ability at an extremely fast learning speed, further investigations and improvements on several algorithms (including hardware and software), frameworks and directions are recommended as follows:

1. Investigating the ability of using high performance types of microcontroller techniques like ARM cortex MCUs with single board such as STM MCU, beagle bone boards, etc.
2. Building own embedded real-time system and application software with stacks multi-source manufacturing and worldwide support.
3. Studying the ability to connect and unify multiple single board systems to work as

one system in supervised by main unit to reduce the limitations of using separate systems for managing the power.

4. Nowadays everything is linked to artificial intelligence that is why our study may benefit in several fields. In our future studies, it may try to focus on 3D images segmentation using deep learning techniques, or on real-time image segmentation using deep learning techniques.
5. Try employing efficient feature extraction approaches that are robust and fast like HOG, SURF, SIFT, BRIEF, LBP, etc.
6. Modify the static saliency map approach to a dynamic type that can handle sophisticated situations rapidly.
7. Exploring the diverse profound learning and neural system classification techniques.

REFERENCES

[1] Daniilidis, K; Maragos, P ; Paragios, N. (2010) Computer Vision -- ECCV, 11th European Conference on Computer Vision, Heraklion, Crete, Greece, Proceedings, Part V., September 5-11.

[2] Rahtu, E., Kannala, J., Salo, M. and Heikkilä, J., 2010, September. Segmenting salient objects from images and videos. In European conference on computer vision (pp. 366-379).

[3] Marchesotti, L., Cifarelli, C., & Csurka, G. (2009, September). A framework for visual saliency detection with applications to image thumbnailing. In 2009 IEEE 12th International Conference on Computer Vision (pp. 2232-2239). IEEE.

[4] Zhaoyu, P., Pingping, L., & Changjiu, L. (2008, December). Automatic Detection of Salient Object Based on Multi-Features. In 2008 Second International Symposium on Intelligent Information Technology Application (Vol. **2,** pp. 437-441).

[5] Cheng, M. M., Mitra, N. J., Huang, X., Torr, P. H., & Hu, S. M. (2014). Global Contrast Based Salient Region Detection. IEEE transactions on pattern analysis and machine intelligence, **37(3),** 569-582.

[6] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet Classification with Deep Convolutional Neural Networks. In Advances in neural information processing systems (pp. 1097-1105).

[7] Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E., & Darrell, T. (2014, January). Decaf: A Deep Convolutional Activation Feature for GenericVisual Recognition. In International conference on machine learning (pp. 647-655).

[8] Yanulevskaya, V., Uijlings, J., & Geusebroek, J. M. (2013). Salient Object Detection: From Pixels to Segments. Image and Vision Computing, **31(1),** 31-42.

[9] Van de Sande, K. E., Uijlings, J. R., Gevers, T., & Smeulders, A. W. (2011, November). Segmentation as Selective Search for Object Recognition. In 2011 International Conference on Computer Vision (pp. 1879-1886).

[10] Achanta, R.; ShajiA., SmithK.; Lucchi, A.; FuaP., Susstrunk, S.,(2010) "SLIC Superpixels," EPFL Technical, Report 149300.

[11] Zujovic, J., (2012) Perceptual Texture Similarity Metrics, Ph.D. Thesis, Northwestern University.

[12] Li, W., Logenthiran, T., Phan, V. T., & Woo, W. L. (2018). Implemented IoT-Based Self-Learning Home Management System (SHMS) for Singapore. IEEE Internet of Things Journal, **5(3),** 2212-2219.

[13] Lee, S. H., & Yang, C. S. (2017). An Intelligent Power Monitoring and Analysis System for Distributed Smart Plugs Sensor Networks. International Journal of Distributed Sensor Networks, **13(7),** 1550147717718462.

[14] Rajasekaran, R. G., Manikandaraj, S., & Kamaleshwar, R. (2017, February). Implementation of Machine Learning Algorithm for Predicting User Behavior and Smart Energy Management. In 2017 International Conference on Data Management, Analytics, and Innovation (ICDMAI) (pp. 24-30).

[15] Hsieh, Y. Z., & Jeng, Y. L. (2017). Development of Home Intelligent Fall Detection IoT System Based on Feedback Optical Flow Convolutional Neural Network. IEEE Access, 6, 6048-6057.

[16] Kumar, N. M., Atluri, K., & Palaparthi, S. (2018, March). *Internet of Things (IoT) in Photovoltaic Systems*. In 2018 National Power Engineering Conference (NPEC) (pp. 1-4).

[17] Chavali, P., Yang, P., & Nehorai, A. (2014). A Distributed Algorithm of Appliance Scheduling for Home Energy Management System. IEEE Transactions on Smart Grid, **5(1),** 282-290.

[18] Hu, Q., & Li, F. (2013). Hardware Design of Smart Home Energy Management System with Dynamic Price Response. IEEE Transactions on Smart grid, **4(4)**, 1878-1887.

[19] Ozturk, Y., Senthilkumar, D., Kumar, S., & Lee, G. (2013). An Intelligent Home Energy Management System to Improve Demand Response. IEEE Transactions on smart Grid, **4(2)**, 694-701.

[20] Fortino, G., Guerrieri, A., O'Hare, G. M., & Ruzzelli, A. (2012). A Flexible Building Management Framework Based on Wireless Sensor and Actuator Networks. Journal of Network and Computer Applications, **35(6)**, 1934-1952.

[21] Shoham, Y., Perrault, R., Brynjolfsson, E., Clark, J. and LeGassick, C. Artificial Intelligence Index, pp. 1-101. Available at: https://aiindex.org/. Accessed at November 2017.

[22] Arel, I., Rose, D. C., & Karnowski, T. P. (2010). Deep Machine Learning-a New Frontier in Artificial Intelligence Research [research frontier]. IEEE computational intelligence magazine, **5(4)**, 13-18.

[23] Buduma, N. and Locascio, N. (2017) “Deep Learning”.

[24] Goertzel, B. (2015, July). Are there Deep Reasons Underlying the Pathologies of Today’s Deep Learning Algorithms. *In International Conference on Artificial General Intelligence* (pp. 70-79).

[25] Zurada, J. M. (1992). Introduction to Artificial Neural Systems (Vol. 8). St. Paul: West.

[26] Srivastava N., Hinton G., Krizhevsky A., Sutskever I., and Salakhutdinov R. “Dropout: A Simple Way to Prevent Neural Networks from Overfitting”, Journal of Machine Learning Research, pp. 1929-1958, 2014.

[27] Li, F. F., Karpathy, A., Johnson, J., Yeung, S., & TAs, A. (2016). Cs231n: Convolutional neural networks for visual recognition 2016. URL http://cs231n. stanford. edu, 37.

[28] Nielsen, M. A. (2015). Neural networks and deep learning (Vol. 2018). San Francisco, CA.

[29] Ng A., Bensouda Y., and Katanforoosh k. “Deep Learning Specialization”, COURSE 1: Neural Networks and Deep Learning, Lecture 1: Introduction to deep learning, https://www.coursera.org/specializations/deep-learning, deeplearning.ai, Coursera, Accessed 20/6/2018.

[30] Wang, X. (2016). Deep Learning in Object Recognition, Detection, and Segmentation. Foundations and Trends in Signal Processing, 8(4), 217-382.

[31] D. Graupe,(2007) Principle of Artificial Neural Networks, 2nd Edition, USAK,TURKEY.

[32] Nagi, J. (2009). An Intelligent System for Detection of non-Technical Losses in Tenaga Nasional Berhad (tnb) Malaysia low voltage distribution network. Vasa.

[33] Prijono B., "Convolutional Neural Networks (CNN) Introduction", https://indoml.com/, Accessed 2018.

[34] Li F., Karpathy A. and Johnson J. “CS231n: Convolutional Neural Networks for Visual Recognition” http://cs231n.stanford.edu/, Accessed 2016.

[35] Ross G.,(2015) “Fast R-CNN”. Computer Vision (ICCV), *International Conference on IEEE*, pp. 1440 – 1448.

[36] Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2015). Region-based convolutional networks for accurate object detection and segmentation. IEEE transactions on pattern analysis and machine intelligence, **38(1)**, 142-158.

[37] LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. Proceedings of the IEEE, **86(11)**, 2278-2324.

[38] Bhatt N., Bhatt N., and Prajapati P, (2007) Deep Learning: A New Perspective International Journal of Latest Technology in Engineering Management & Applied Science (IJLTEMAS), vol VI, pp.2278-2540.

[39] Bengio Y.(2009) Learning Deep Architectures for AI Foundations and Trends in Machine Learning, vol 2, no 1, pp 1–127.

[40] He, K., Zhang, X., Ren, S., & Sun, J. (2015). Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. IEEE transactions on pattern analysis and machine intelligence, 37(9), 1904-1916.

[41] Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

[42] Simonyan, K., & Zisserman, A. (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition. arXiv preprint arXiv:1409.1556.

[43] Moujahid, A. Figure of the Training and Prediction. Adilmoujahid.com Post, http://adilmoujahid.com/posts/2016/06/introductiondeep-learning-python caffe/?utm_source=top.caibaojian.com/111120. Accessed in 26 June 2016.

[44] Ashaj, S. J., & Erçelebi, E. (2019). Reduce Cost Smart Power Management System by Utilize Single Board Computer Artificial Neural Networks for Smart Systems. International Journal of Computational Intelligence Systems, **12(2)**, 1113-1120.

[45] Ashaj, S. J., & Erçelebi, E. (2020). Energy Saving Data Aggregation Algorithms in Building Automation for Health and Security Monitoring and Privacy in Medical Internet of Things. Journal of Medical Imaging and Health Informatics, 10(1), 204-210.

[46] Everingham, M., Van Gool, L., Williams, C. K., Winn, J., & Zisserman, A. (2010). The Pascal Visual Object Classes (voc) Challenge. International journal of computer vision**, 88(2)**, 303-338.

[47] Oxford Robotics Institute “The PASCAL Visual Object Classes”. Available at: http://host.robots.ox.ac.uk/pascal/VOC/ (Accessed: 29 April 2018)

[48] Lin, T. Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., ... & Zitnick, C. L. (2014, September). Microsoft coco: Common objects in context. In European conference on computer vision (pp. 740-755). Springer, Cham.

[49] Lobaccaro, G., Carlucci, S., & Löfström, E. (2016). A review of systems and technologies for smart homes and smart grids. Energies, **9(5)**, 348.

[50] Kochar G. S., Venkatesan N. and Mahato A. 2017. “PIR Sensors with Drones”. International Journal of Electronics, Electrical and Computational System, Vol. 6, Issue 8, pp.575-577.

[51] Ogollah O. F. 2010. Room Occupancy Monitoring and Counting Device. Project Index: 76, University of Nairobi.

[52] D. Toth and T. 2003. Aach. Detection and Recognition of Moving Objects Using Statistical Motion Detection and Fourier Descriptors. Proc. 12th Int’l Conf. Image Analysis and Processing (ICIAP 03), pp.430-435.

[53] Wang X., Cui L., and Guo Z. 2014. “Advanced Technologies in Ad Hoc and Sensor Networks”: Proceedings of the 7th China Conference on Wireless Sensor Networks. Book, Springer, ISBN: 9783642541742.

[54] Ramli N. L., N. Yamin M., Ghani S. Ab, Saad N. Md. and Sharif S. A. Md. 2015. “Implementation of Passive Infrared Sensor in Street Lighting Automation System”. ARPN Journal of Engineering and Applied Sciences, Asian Research Publishing Network (ARPN), Vol.10, No. 22, pp.17120- 17126.

[55] Patel P. B., Choksi V. M., Jadhav S. and Potdar M. B. 2016. “International Journal of Applied Information Systems”, Vol.10, No.5, pp.37-40.

[56] Twumasi, C., Dotche, K. A., Banuenumah, W., & Sekyere, F. (2017, June). Energy saving system using a PIR sensor for classroom monitoring. In 2017 IEEE PES PowerAfrica (pp. 347-351).

[57] Khan, N. M., Khalid, Z., Ahmed, G., & Yasin, M. (2007, April). A robust routing strategy for wireless sensor networks. In 2007 International Conference on Electrical Engineering (pp. 1-5). IEEE.

[58] Alippi, C., Anastasi, G., Di Francesco, M., & Roveri, M. (2009). Energy Management in Wireless Sensor Networks with Energy-Hungry Sensors. IEEE Instrumentation & Measurement Magazine, **12(2)**, 16-23.

[59] Čagalj, M., Hubaux, J. P., & Enz, C. C. (2005). Energy-Efficient Broadcasting in all-Wireless Networks. Wireless Networks, **11(1-2)**, 177-188.

[60] Polastre, J., Szewczyk, R., & Culler, D. (2005, April). Telos: Enabling Ultra-Low Power Wireless Research. In IPSN 2005. Fourth International Symposium on Information Processing in Sensor Networks, 2005. (pp. 364-369). IEEE.

[61] Dong, Q., Yu, L., Lu, H., Hong, Z., & Chen, Y. (2010). Design of Building Monitoring Systems Based on Wireless Sensor Networks. Wireless Sensor Netowrks, **2(9)**.

[62] Percival Magpantayz, Igor Paprotnyy, Rafael Send, Qiliang Xu, and Chris Sherman, (2014) , Energy Monitoring in Smart Buildings Using Wireless Sensor Networks, The Third International Conference on Smart Systems, Devices and Technologies, IARIA, Paris (France) .

[63] Jesús Blanco, Andrés García, and Javier de las Morenas,(2018), Design and Implementation of a Wireless Sensor and Actuator Network to Support the Intelligent Control of Efficient Energy Usage, Sensors (Basel), **18(6)**, pp: 1892.

[64] Adam B. Noel, AbderrazakAbdaoui, Tarek Elfouly,Mohamed Hossam Ahmed, Ahmed Badawy, and Mohamed S. Shehata,(2017) Structural Health Monitoring using Wireless Sensor Networks: A Comprehensive Survey, IEEE Communications Surveys & Tutorials, Volume: 19 , Issue: 3, pp: 1403 – 1423.

[65] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan,(2000) Energy efficient communication protocol for wireless microsensor networks, in Proc. 33rd Annu. Hawaii Int. Conf. Syst. Sci., Maui (USA),

[66] A. Manjeshwar, and D. P. Agrawal, (2001), TEEN: A Routing Protocol for Enhanced Efficiency in Wireless Sensor Networks, in Proceeding of 15th IPDPS, IEEE, San Francisco (USA), volume: 1, pp: 189.

[67] G. Smaragdakis, I. Matta, and A. Bestavros,(2004) ,SEP: A Stable Election Protocol for Clustered Heterogeneous Wireless Sensor Networks, Dept. Comput. Sci., Boston Univ., Boston, MA, USA, Tech. Rep. BUCS-TR-2004-022.

[68] Hemant Mahajan, Abrosimov Mikhail, and Iehab Abdul jabbar Kamil,(2017) Increasing SCADA System Availability by Fault Tolerance Techniques, International Conference on Computing, Communication, Control and Automation (ICCUBEA).

[69] Hemant Mahajan, Abrosimov Mikhail, and Hayder Hussein Kareem, (2017) Fault Tolerance to Balance for Messaging Layers in Communication Society, IEEE International Conference on Computing, Communication, Control and Automation (ICCUBEA).

[70] Lin, S., Miao, F., Zhang, J., Zhou, G., Gu, L., He, T., ... & Pappas, G. J. (2016). ATPC: Adaptive Transmission Power Control for Wireless Sensor Networks. ACM Transactions on Sensor Networks (TOSN), 12(1), 1-31.

[71] Chipara, O., He, Z., Xing, G., Chen, Q., Wang, X., Lu, C., ... & Abdelzaher, T. (2005). Real-time power aware routing in wireless sensor networks.

[72] Ghufran Ahmed, Noor M Khan, and Mirza M Yasir Masood,(2011) A Dynamic Transmission Power Control Routing Protocol to Avoid Network Partitioning in Wireless Sensor Networks, International Conference on Information and Communication Technologies.

[73] Ciullo, D., Celik, G. D., & Modiano, E. (2010, May). Minimizing transmission energy in sensor networks via trajectory control. In 8th International

Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (pp. 132-141).

[74] Chincoli, M., & Liotta, A. (2018). Self-learning Power Control in Wireless Sensor Networks. Sensors, **18(2)**, 375.

[75] Le T.T.T., and Moh, S,(2017),An Energy-Efficient Topology Control Algorithm Based on Reinforcement Learning for Wireless Sensor Networks, International Journal of Control Automation, 10, 233–244.

[76] Yau, K.L.A., Goh, H.G., Chieng, D., and Kwong, K.H. (2015), Application of reinforcement learning to wireless sensor networks: Models and algorithms, Computing, 97, 1045–1075.

[77] Chincoli M., Syed A.A., Exarchakos G., and Liotta, A., (2016) Power Control in Wireless Sensor Networks with Variable Interference, Mobile Information Systems. 1–10.

[78] Xiaoping Yang, Xueying Chen, Riting Xia, and Zhihong Qian,(2018) Wireless Sensor Network Congestion Control Basedon Standard Particle Swarm Optimization and Single Neuron PID, Sensors (Basel), 19;18(4).

[79] Joseph, A. J., & Hari, U. (2013). Hexagonally Sectored Routing Protocol for Wireless Sensor Networks. Int. J. Eng., **2(5)**, 1249-1252.

[80] Wang, S. S., & Chen, Z. P. (2012). LCM: A Link-Aware Clustering Mechanism for Energy-Efficient Routing in Wireless Sensor Networks. IEEE sensors journal, **13(2)**, 728-736.

[81] Ahmad, A., Javaid, N., Khan, Z. A., & Qasim, U. (2014). and Turki Ali Alghamdi. IEEE SENSORS JOURNAL, **14(10)**.

[82] Hyunjo Lee, Miyoung Jang, and Jae-Woo Chang,(2014), A New Energy-Efficient Cluster-Based Routing Protocol Using a Representative Path in Wireless Sensor Networks, International Journal of Distributed Sensor Networks, Volume 2014.

[83] Srinivas Kanakala, Venugopal Reddy Ananthula, and PrashanthiVempaty,(2014) Energy-Efficient Cluster Based Routing Protocol in Mobile Ad Hoc Networks Using Network Coding, Journal of Computer Networks and Communications, Volume 2014.

[84] Gupta, A., Thakur, A., Saini, H. S., Kumar, R., & Kumar, N. (2016, July). H-IECBR: HBO Based-Improved Energy Efficient Chain Based Routing Protocol in WSN. In 2016 IEEE 1st International Conference on Power Electronics, Intelligent Control and Energy Systems (ICPEICES) (pp. 1-4). IEEE.

[85] Amjad, M., Afzal, M. K., Umer, T., & Kim, B. S. (2017). QoS-Aware and Heterogeneously Clustered Routing Protocol for Wireless Sensor Networks. IEEE Access, **5,** 10250-10262.

[86] Zhang, W., Li, L., Han, G., & Zhang, L. (2017). E2HRC: an Energy-Efficient Heterogeneous Ring Clustering Routing Protocol for Wireless Sensor Networks. IEEE Access, **5**, 1702-1713.

APPENDİX A

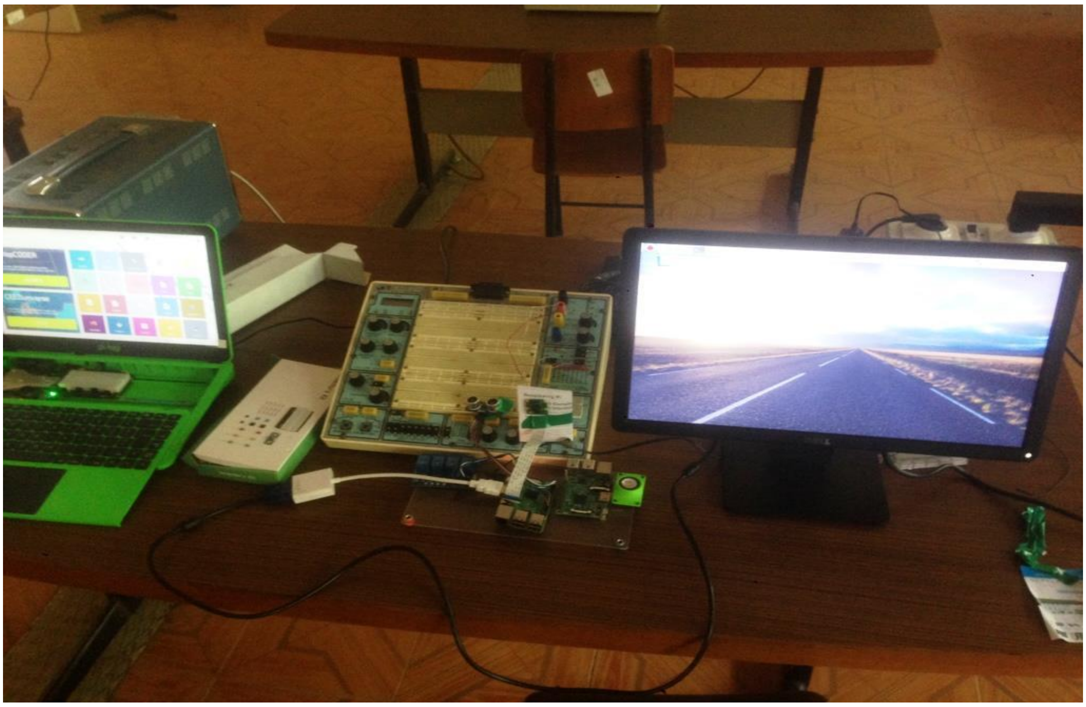


Figure A1

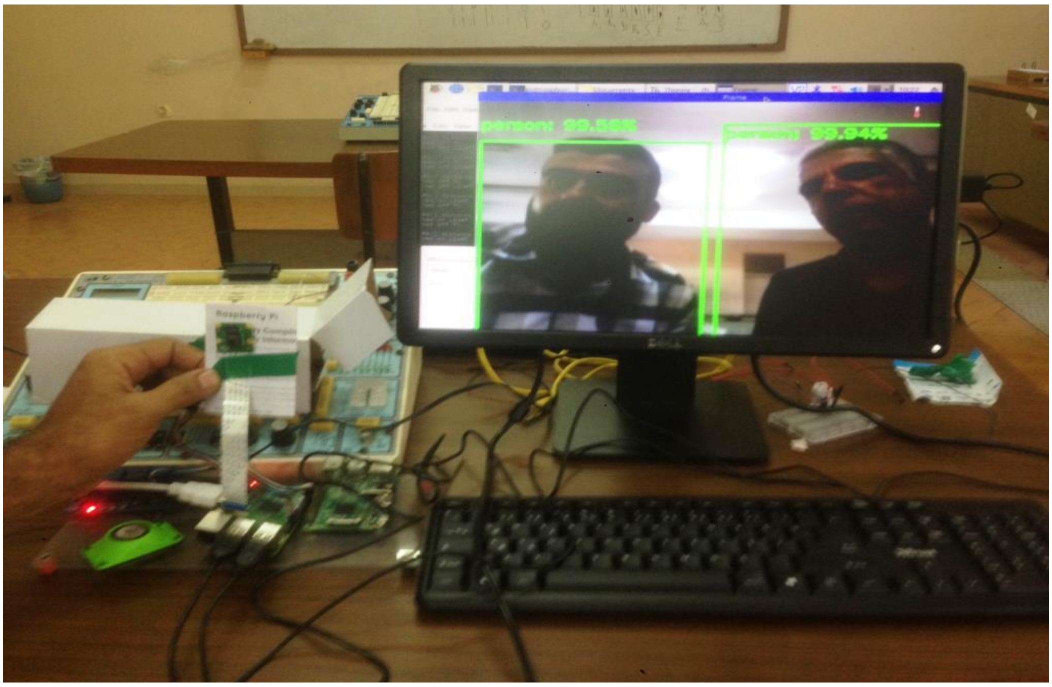


Figure A2

CURRICULUM VITAE

PERSONAL INFORMATION

Name and Surname: SUDAD J. ASHAJ

Birthplace and date: BAGHDAD – IRAQ 08-10-1980

Mariel status: MARRIED

Phone number: +905349790697 - +9647704090604

Email: sudad_80@yahoo.com

EDUCATION

Graduate school	Year
Ph.D. Gaziantep University	2020
Master UTM University	2013
Bachelor Technology University	2005

PUBLICATIONS

International Conference:

1. Ashaj, S. J., & Erçelebi, E. (2018). International conference Design and Implementation of Energy Management System for Buildings. Proceedings of the Fifth International Conference on Technological Advances in Electrical, Electronics and Computer Engineering (TAEECE2018), Turkey.

International Journals:

2. Ashaj, S. J., & Erçelebi, E. (2019). Reduce Cost Smart Power Management System by Utilize Single Board Computer Artificial Neural Networks for Smart Systems. International Journal of Computational Intelligence Systems, 12(2), 1113-1120.
3. Ashaj, S. J., & Erçelebi, E. (2020). Energy Saving Data Aggregation Algorithms in Building Automation for Health and Security Monitoring and Privacy in Medical Internet of Things. Journal of Medical Imaging and Health Informatics, 10(1), 204-210.