Republic of Iraq
Ministry of Higher Education
and Scientific Research
University of Babylon
College of Engineering
Department of Electrical Engineering

Threat Object Detection and Classification of X-Ray Imaging Using Deep Learning and Embedded System.

A Thesis
Submitted to the College of Engineering / University of Babylon in Partial Fulfillment of the Requirements for the Degree of Master in Engineering/ Electrical Engineering / Industrial Electronics

By

Saif Sarmad Hussein Omran

Supervised by

Assistant prof. Dr. Hilal Al-Libawy

2022 A.D. *1444 A.H.*

Examining committee certificate

We certify that we have read this thesis entitled "**Threat Object Detection and Classification of X-Ray Imaging Using Deep Learning and Embedded System**" and as an examining committee, examined the student **Saif Sarmad Hussein** in its content and that in our opinion it meets standard of a thesis for the degree of master in engineering/electrical engineering / Industrial Electronic.

Signature:	Signature:
Name: **Prof. Dr.**	Name: **Prof.Dr**
Osama Qasim Jumah	**Haider Ismael Shahadi**
(chairman)	(member)
Date: / /2022	Date: / /2022

Signature:	Signature:
Name: **Asst.Prof.**	Name: **Asst.Prof.**
Dr. Raed Saleem Hashim	**Dr. Hilal AL_Libawy**
(member)	(supervisor)
Date: / /2022	Date: / /2022

Name: **Asst. Prof.**
Dr. Shamam Fadhil Alwash
(Head of Electrical Department)
Date: / /2022

Signature:
Name: **Prof.Dr**
Hatem Hadi Obeid
(Dean of College of Engineering)
Date: / /2022

Supervisors Certification

I certify that this thesis titled **“Threat object detection and classification of x-ray imaging using deep learning and Embedded system”** was prepared by **Saif Sarmad Hussein** under my supervision at the Electrical Engineering Department, College of Engineering at University of Babylon, in a partial fulfillment of the requirements for the degree of Master of Science in Electrical Engineering/Industrial Electronics.

<u>Supervisor</u>

Signature:

Name*: Asst. prof. Dr. Hilal Al-Libawy*

Date: / / 2022

In view of the available recommendation, we forward this thesis for debate by the examining committee.

<u>Head of the Electrical Engineering Department</u>

Signature:

Name: Asst. Prof .Dr. Shamam Fadhil Alwash

Date: / / 2022

Acknowledgements

Thanks be to God for the insight that inspired me, and praise be to Him for the blessings He has bestowed upon me, and to Him is the credit for achieving what I aspire to in this research.

*It gives me great pleasure to extend my thanks and gratitude to (**Assistant. prof. Dr. Hilal Al-Libawy**), who provided me with the information I needed in my research*

And for the great efforts he made for me since he proposed the subject of the research and his continuous supervision, and for his advice that helped me a lot in overcoming the difficulties I faced, calling from God success and brilliance in the scientific career.

I would like to extend my thanks and gratitude to the Deanship of the College of Engineering and the Presidency of the Electrical Engineering Department for the facilities and administrative procedures they provided me, which effectively contributed to the completion of the requirements of this research.

I would like to extend my thanks to google to the Google site for providing the Kaggle site, which gave me 20 hours for free every week to implement the software on its servers.

Researcher

Dedication

To my wonderful father.... who taught me the meaning of life and how to choose the best.

To my dear mother... the source of love and giving that words cannot describe, and my inspiration and encouragement in my scientific career

To my brothers and sister... my support and I share my joys and sorrows.

To my beloved wife… who supported me and endured many life's difficulties to complete my thesis.

To all my friends and to all those from
whom I received advice and support;

I dedicate to you the summary of my scientific effort.

Abstract

Threat Objects Detection (TOD) is a system used to scan X-ray images of passenger's baggage for the purpose of detection and classification threat objects. The human inspection is a time-consuming task and slow process, in additional many human mistakes happened, especially in rush hours. Recently, new techniques have been used in TOD systems to increase accuracy and pace of inspection process such as deep learning. The most popular deep learning approaches, have achieved high performance and speed in X-ray image processing to detect threat elements during baggage inspection. However, the exist technique still suffering from critical problems such as overfitting and need updating the dataset. This thesis proposes a TOD system based on Convolution Neural Networks (CNNs). Moreover, the proposed system implemented embedded system using Raspberry pi 4 Model B.

The proposed TOD system consists of DenseNet 121 network and the logistic regression algorithm. To achieve high accuracy detection of threat objects, the system uses Test Time Augmentation (TTA) technique. Work execution environment is open servers (Kaggle), From which the DenseNet 121 network and the logistic regression algorithm are downloaded. Train and test the proposed TOD system on a GDX-ray dataset that contains baggage images of four types of (guns, shuriken, blade and others). Recall metric results for images classification is 97.68%. As far as we know, the GDX-ray dataset has not been executed on an embedded device using Raspberry Pi4 model B. The implementation stage achieved high-recall metric results for real-time images classification is 94.25%. The use of data augmentation technology during training and testing is one of the best techniques that lead to improving the performance of the proposed system.

List of Content

List of Content

List of Tables

List of Figures

List of Figures

List of Figures

List of Abbreviations

Abbreviation	Definition
Acc	Accuracy
ANN	Artificial Neural Network
CNN	Convolutional Neural Network
CPU	Central Processing Unit
DL	Deep Learning
DNN	Deep Neural Network
D-W	Depth-Wise
Faster RCNN	Faster Region based Convolution Neural Network
FC	Fully Connected
FPGA	Field Programmable Gate Array
GANs	Generative Adversarial Networks
GB	Giga Byte
GPIO	General Purpose input/output
GPU	Graphics Processing Unit
HDMI	High-Definition Multimedia Interface
KNN	Number of K Nearest Neighbor
LR	Logistic regression
MAP	Mean Accuracy Precision
ML	Machine Learning
MVSR	Mean-Variance-Softmax-Rescale
oBIFs	oriented Basic Images Features
R_FCN	Region_ based Fully Convolutional Networks
RAM	Random-Access Memory
SBCs	Single-Board Computers
SD	Secure Digital
SSD	Single Shot Detector
TOD	Threat Objects Detection
TPR	True Positive Rate

List of Abbreviations

Abbreviation	Definition
TTA	Test Time Augmentation
USP	Universal Serial Bus

List of Symbols

Symbol	Definition
η	Average Recall
η_i	Recall for Each Class
λp	Wavelength of a Photon
μ	Attenuation Coefficient
ρav	Mean Pooling
ρ_m	Maximum Pooling
$\sigma(\Upsilon)$	Sigmoid Function
$\sigma(x_c)$	Softmax Function
C_o	Light In Space
d	Euclidean distance
e_k	Error function
Ep	Energy of Photon
f	Transfer Function
fp	Frequency Photon
h	Planck's Constant
$H(p,y)$	Cross-Entropy Loss Function
I	captured intensity
I_o	Emitted Intensity
nf	Feature Map
P	Precision
T	Threshold Value
w_i	Weight
X_i	Input Sample
X_i	Input
y	Output of Node x_i, and T is the threshold value
Y_i	New Sample
Z	Thickness

Chapter One

Introduction

CHAPTER ONE

INTRODUCTION

1.1 Overview

In recent years, researchers are interested in designing and implementing accelerated systems that detect and classify threat objects in X-ray images of passenger baggage. The X-ray inspection system is considered as one of the effective systems to reduce terrorist attacks. It helps the human inspectors to facilitate their task of seeing the contents of the various baggage within a limited period of time because the X-ray displays the items of the baggage on a screen. The Threat Object Detection TOD system detects threats elements by displaying x-ray images of passengers' baggage, the automatic inspection of the TOD system using machine learning achieves higher speed and accuracy than manual inspection [1-4].

The use of machine learning algorithms on X-ray images has improved the performance of threats detection in stadiums, airports and government buildings such as detection (guns, explosives, knives, etc.) [2]. Its importance derives from its high effectiveness in reducing the risks of terrorism [5] [3, 6]. The researchers found that manual inspection is better for detecting the threat compared to the machine in the case of a high threat rate and Human eyes have a higher ability to detect the threat object inside the plastic bags during the peak [4, 7]. However, many challenges in manual inspection, such as the lack of threatening elements or small threat objects which complicates the process of detecting threat elements [1, 8, 9]. Moreover, the manual inspection process is stressful and boring [2]. (CNNs) is one of the deep learning algorithms that have been used in the field of computer vision and have proven their high performance in detecting threat objects in X-ray images.[5]. One of the most important reasons that prompted the priority of using (CNNs) is the speed of the inspection of baggage compared to human inspection [10].

Performance accuracy based on human inspection is from (80% to 90%)[11] .Although (CNNs) have achieved more than 90% performance in detecting threat objects[2]. Development work is still open for CNNs algorithms because they always need to be trained on a new and big dataset to reduce loss of generality and overfitting problems [4].

1.2 Literature Review

High performance achieved by convolutional neural network algorithms with high accuracy and speed in the field of computer vision. These algorithms encouraged researchers to use them to detect threat elements in travelers' baggage. As shown in the following way:

A. Binary Classification

Binary classification is one of the ways to prevent threat images. The principle of this classification is to divide the elements into two categories, hazardous and non-hazardous as shown in the following related works:

In 2016, S. Akcay et al. [12] The AlexNet algorithm was introduced with the use of transfer learning. The last layers are adjusted and the first layers are frozen because the features extracted from the first layers are rich, which leads to the improvement of the performance of the binary classification process for X-ray images from a DBP2 dataset. The presence of a gun against no gun and the results of the test accuracy are 98%.

In 2016, N. Jaccard et al. [13] presented a comparison between the use of classical machine learning and deep learning, In a binary classification mission between non-empty cargo of small threat metal or cars as in the task of first classifying non-empty or empty cargo containers. In the second classification task, cars are categorized from containers classified as non-empty. Using stream_ of_ commerce Dataset. Containers are verified using oriented Basic Images Features (oBIFs) by limiting the size of features that can be detected. Random forests are used with this method. The researchers used the 19-layer and 9-layer CNN, and the highest detection rate was obtained from the 19-layer CNN, 91%.

In 2017, S. Akcay and T. P. Breckon. [14] presented the Region_ based Fully convolutional networks (R_FCN) model was introduced. They trained the model with using ResNet-50 and ResNet-101 to extract features for optimal performance. A learning transfer approach is used pre-trained networks on ImageNet. They used a DBF6 dataset containing 15,449 samples in the task of classifying firearms against a background The highest accuracy of 96.3 was achieved by applying (R_FCN) model with ResNet-101 network.

In 2019, V. Riffo et al. [15] presented a Multiview single-spectral X-ray by using space carving for a three-dimensional reconstruction, computer vision algorithms were used. During training, a database is built to extract features to know the target object of the handgun. During the test, they performed a 3D reconstruction of the test images to extract images by matching the test descriptors with the training descriptors. 8000 samples taken from GDXray dataset were matched using 3D shape context (3DSC), to get the best performance for classifying images recall and precision 97%.

The researchers worked on machine learning algorithms with good results and used a data set of X-ray images to detect threat elements by the binary classification of images. However; multi-classification is preferred over binary classification to know the type of threat category.

B. Multi- Classification

The task of multi classification is more complicated than the task of binary classification because of the large number of categories to be classified and the level of challenges is higher, as will be explained as follows:

In 2016, D. Mery et al. [2] presented the (CNN) models to extract features from images the best layer of AlexNet. The nearest neighbor algorithm (KNN) has been trained on the features extracted from the AlexNet to get the final classification results, reaching (91.2%). When the previous method was used by means of GoogleNet instead of AlexNet and on the same GDXray dataset, the results reach to (96.3%).

In 2018, I.Aydin et al. [16] introduced Faster Region Based Convolution Neural Network algorithm (faster RCNN). The faster RCNN algorithm detects the threat elements and encloses them in boxes. After the elements are detected, two convolutional neural networks are used to perform the classification task. The researchers used three categories of X-ray images from the GDXray dataset (handgun, shuriken, razor). By using the above algorithm, the researchers were able to classify and detect are threat elements with an accuracy of 98.42%.

In 2019 J.Yang et al. [17] proposed a method for augmenting the data of the prepared elements for X-ray images by using Generative Adversarial Networks (GANs) to improve performance in detecting threats. Initially, a model is used original Deep Convolutional Generative Adversarial Network (DCGAN) to generate images, but the images still need a higher quality because there are many noises, especially in the images of the firearm. Wasserstein used the model GAN with gradient penalty (WGAN-GP), researchers were able to raise the quality of the images. In the case of training the CNN model on real images and the classification accuracy is 97.8%. If the CNN model is trained on the real images and the generated images, the classification accuracy is 98.37. Researchers added 1000 generated images for each category (wrench, fork, handgun, power bank, lighter, pliers, knife, liquid, umbrella and screwdriver).

In 2020, Y.Wei and X.Liu. [18] introduced a deep learning algorithm called a Single Shot Detector (SSD). They used the transfer of learning methodology for the task of classifying and detecting threat images from GDXray dataset. The images contain four categories (Knife, Handgun, Shriken and a Razor blade). The researchers made a comparison after applying the transfer of learning between the SSD300 model and the SSD512 model. The best performance was obtained by using the SSD300 model that achieved a mean accuracy precision (mAP) of 91.5%. If the data is large enough, the SSD512 model will perform better.

In 2020, J. Yong, et al [19] introduced a methodology for data augmentation based on generative adversarial network (GAN) to improve performance based on the faster region convolutional neural network (Faster RCNN) model. The researchers used an unbalanced SIXray dataset (is a set of security x-ray images). The deep convolutional GAN (DCGAN) has been adapted to create new X-ray images. The training set has been augmented with new images. By increasing the data, the researchers were able to treat the problem of data imbalance, and they were able to divide the installation of images and combine the transformation of images and installation of images. The results showed a clear improvement on the model and the accuracy of true positive rate reached (TPR) 96.6% for all Categories Gun, Knife, Wrench, Pliers and Scissors.

In 2021, P.Steno, Priscilla, et al. [20] presented faster region-based on the convolutional neural network (Faster R-CNN) model to make the performance of detecting threats better. The model is based on extracting the objects of the proposals of the threat organism in humans and extracting the degree of the object to determine the position. This system is used modified loss function to improve the performance of the classifier. New binding boxes are designed to increase the object consolidation ability of the system and detect its localization. The system proposed which has reached 96.2%.

In 2021 Y.Wei, et al. [21] presented Model of detection and classification of threat elements in X-ray images of the GDXray dataset. The model depends on depth-wise separable convolution. This model has benefits for improving detection accuracy and speeding up the performance which consists of depth-wise (DW) Convolution and Pointwise (PW) Convolution. The name of the model proposed by researchers is ThreNet model and it is a combination of MobileNet V2 network optimization, deformation layer and three scales. The system was able to achieve Precision 99.07%.

The researchers worked on the multi-classification of X-ray images that contain threatening elements and got good results. However, the results remained below the level of ambition. Some results, though highly accurate, but the number of classification categories is less. Some results have high accuracy and a large number of categories but for a different data set.

C. Classification in Real Time

Few real-time TOD system research has been carried out, and those who classified the threat images in real time managed to get high results. The researchers used the embedded devices to classify the images, but most of the research is concerned with the medical field. As shown in the following:

In 2020, Z.Zhu, Zhanyu, and Feng Xu. [22] Presented a three-dimensional imaging experiment at high frequencies that is displayed with a single dimension of multi inputs and outputs, where the matrix sends 8 and receives 16 to reach an actual imaging along the vertical column. The researchers experimented with people moving by a cart, and the algorithm was implemented on a NVIDIA device. They used a deep neural network algorithm DNN, and the algorithm was trained on a data set of images of people who hide multi elements. The accuracy of the classification of pedestrian safety reached 67%.

In 2021, Y.Wei, , et al. [23] Presented CNN-based classifier network. It contains one input layer, five convolution layers, five batch normalization layers and one output layer. The network classifies x-ray images of bags through the front section of the bag. The data set contains of 15873 images (large-sized, medium-sized and small-sized), each category of bags contains 5,291 images. In the task of classifying the three categories of bags, researchers achieved an accuracy of 96.01%. The researchers ran their experiments on the ubuntu 16.04. operating system, by using one NVIDIA TiTan-V GPU with 12GB memory. They managed to get the real-time classification accuracy of the T-Walt system to 96%.

In 2021 V.Ponnusamy, et al. [24] Introduced the YOLO algorithm and its representation in a programmable gate FPGA classification and X-ray image detection from the GDXray dataset,. The researchers were able to use YOLOV3 on Google Collop and a web download on the PYNQ-Board containing an FPGA that allows Python to be used with its libraries. They were able to binary classification as hazardous or non-hazardous for x-ray images that contain a hazardous object. The researchers were able to get a higher mean precision (mAP) of 98.9%.

In 2022 S.Yaman, et al. [25] Introduced a Mean-Variance-Softmax-Rescale (MVSR) algorithm, They were able to classify the chest x-rays and find out if the person had Covid-19. After applying the algorithms to the X-ray images, they were able to obtain high results in the task of classifying those images. The researchers were able to implement a convolution network in a MATLAP environment. The researchers were able to carry out the work on the FPGA platform, with an accuracy is 96.16%.

Table (1.1): Summary of literature review

References	Algorithm	Dataset	Results	Nots
S. Akcay et al [12] .	AlexNet	DBP2	Accuracy 98%.	Binary Classification
N. Jaccard et al [13].	19-layer CNN,	Stream_ of_ Commerce	Accuracy 91%	Binary Classification
S. Akcay and T. P. Breckon [14].	R_FCN	DBP6	accuracy 96.3%	Binary Classification
V. Riffo et al [15].	Multiview single-spectral	GDXray	Recall and Precision 97%.	Binary Classification
D. Mery et al [2].	GoogleNet	GDXray Baggage	Recall 96.3%	Multi Classification 4-classes
I.Aydin et al [16].	Faster RCNN	GDXray Baggage	Accuracy 98.42%	Multi Classification 3-classes
J.Yang et al [17].	DCGAN	GDXray	Accuracy 98.42%	Multi Classification 11-classes
Y.Wei and X.Liu [18].	SSD	GDXray Baggage	Precision 91.5%	Multi Classification 3-classes
J. Yong, et al [19].	Faster R-CNN	GDXray	Accuracy 96.6%	Multi Classification 5-classes

Table (1.1): (Continued)

References	Algorithm	Dataset	Results	Nots
Y.Wei, et al [21].	ThreNet	GDXray Baggage	Precision 99.07%	Multi Classification 3-classes
Z.Zhu, Zhanyu, and Feng Xu [22].	DNN	Images of people	Accuracy 67%	Multi Classification in real time
Y.Wei, et al [23].	CNN-based classifier	X-ray images of bags	Accuracy 96%	Multi Classification 3-classes in real time
V.Ponnusamy, et al [24].	YOLOV3	GDXray	Precision 98.9%	Binary Classification in real time
S.Yaman, et al [25]	MVSR	Chest X-ray	Accuracy 96%	Binary Classification in real time

1.3 Scope

The GDX ray dataset available exclusively on the Internet for this work was used. in the security field, it is hard obtain to most the dataset that used in publication. Also, these datasets are confidentially agreed with researchers and they are not allowed to distribute them. The implementation of proposed algorithm is focused on embedded devices. The embedded device is suitable for work area environments.

1.4 Problem Statement

Threat elements detection becomes compulsory necessity, especially in airport. The human baggage inspection un accurate and time consuming. The researchers in this area still suffering from one or more of the following challenges:

1. The classifier based deep learning has overfitting problems.
2. Real time implementation of Threat Objects Detection (TOD).

1.5 Thesis Objective

1. Used CNNs algorithms to speed up in the classification task to detect threat elements.
2. The DenseNet121 network use for its high performance, with the use of augmentation and test time augmentation to reduce overfitting problem.
3. Real time implementation using embedded device low cost and simple by using raspberry pi 4 model B.

1.6 Thesis Layout

In additional to chapter one the thesis organized as follows:

Chapter two explains X-ray and all the concepts related to TOD. Algorithms and techniques to perform the task of classifying x-ray images and obtaining the highest accuracy in addition to clarifying the embedded device.

Chapter three presents the proposed system to obtain the highest possible recall, with an explanation of the parts of the proposed system and the techniques that were used to improve the system.

Chapter four explains the implementation of work and showing the results by using the proposed system and implementation of the system using software and Embedded device. The chapter also provides comparisons with previous works.

Chapter five includes conclusions and future work that can add higher accuracy and speed to classify and detect threat objects.

Chapter Two

Theoretical Background

CHAPTER TWO

THEORETICAL BACKGROUND

2.1 Introduction

Threat Object Detection(TOD) is a system used in airports that scans x-ray images of passengers' baggage to detect elements of threat [26]. The X-ray imaging system is considered one of the vital and effective systems for protecting the environments [27]. There are many systems that image objects such as ultrasound imaging and thermal imaging [28, 29]. However, it is preferable to use X-ray imaging in detecting threat objects because the images are high-resolution of the objects inside the bags. X-ray imaging tests reduce the risk of terrorism and criminal cases [5, 30]. X-ray imaging is used in many fields in medicine, industry and security, and it is considered a low-cost X-ray imaging technique [31]. Machine learning algorithms are used to address the problem of recognizing categories in X-ray images. Machine learning is a branch from artificial intelligence. There are computational algorithms designed to simulate human intelligence and the machine is learning by dataset. The application of machine learning in many fields such as computer vision, spacecraft engineering and medicine applications. Classification of X-ray images is one of the tasks provided by machine learning. The importance of machine learning lies in its ability to deal with big data and categories with complex features. Machine learning algorithms such as logistic regression (LR), artificial neural network (ANN) and k-Nearest Neighbor (KNN) are used in the classification mission[32-35]. Deep learning algorithms can be used in the classification task, such as the convolution algorithm, which is rich in weight because it contains many hidden layers. The convolution algorithms showed a high performance in classifying X-ray images by improving the classification accuracy[4, 36].

2.2 Threat Object Detection TOD System

TOD is a system that scans x-ray images of passenger baggage at airports to display threat objects. Pre-recorded threat objects of X-ray images are displayed on X-ray baggage images to improve performance in threat objects classification and detection. The use of the TOD-system has raised the level of vigilance of workers in the field of baggage inspection at airports, so it has become widespread [23, 26, 37].

2.3 X-Ray

X-rays are electromagnetic radiation of high energy and can be represented by a wave model or photons. The wavelength of X-rays ranges from 1nm to 10nm, the voltage ranges from 50KV to 150KV. The speed of X-rays in a vacuum is equal to the speed of light, and its speed is less when it passes through other materials[38, 39]. The wavelength of a photon λp can be calculated by the frequency a photon fp and the speed of light in space C_0 through an equation (2.1) [39].

$$\lambda p = \frac{Co}{fp} \qquad (2.1)$$

The energy of a photon can be found from equation(2.2) [39].

$$Ep = fp\, h \qquad (2.2)$$

Where, Ep: The energy of a photon, h:(Planck’s constant ≈6.626 069 × 10−34 J s) [38].

A. X-Ray Generation:

X-rays are produced by two main components, the X-ray generator and the X-ray tube. The x-ray generator provides the control and power source for the x-ray tube. An x-ray tube contains the components and environment for x-ray extraction. The x-ray tube consists of a glass tube, cathode and anode. The cathode consists of two parts, the filament and the focusing cup. The potential difference applied to the cathode causes the filament to heat up, causing it to generate accelerated electrons. The focusing cup determines the path of the electrons leaving the filament towards the anode. The anode consists of two parts, a tungsten plate and a copper base. When the accelerated electrons coming from the cathode collide with tungsten metal, X-rays are generated. The copper base is used to absorb the heat generated by the collision of accelerated electrons in the tungsten material as shown in figure (2.1) [38].

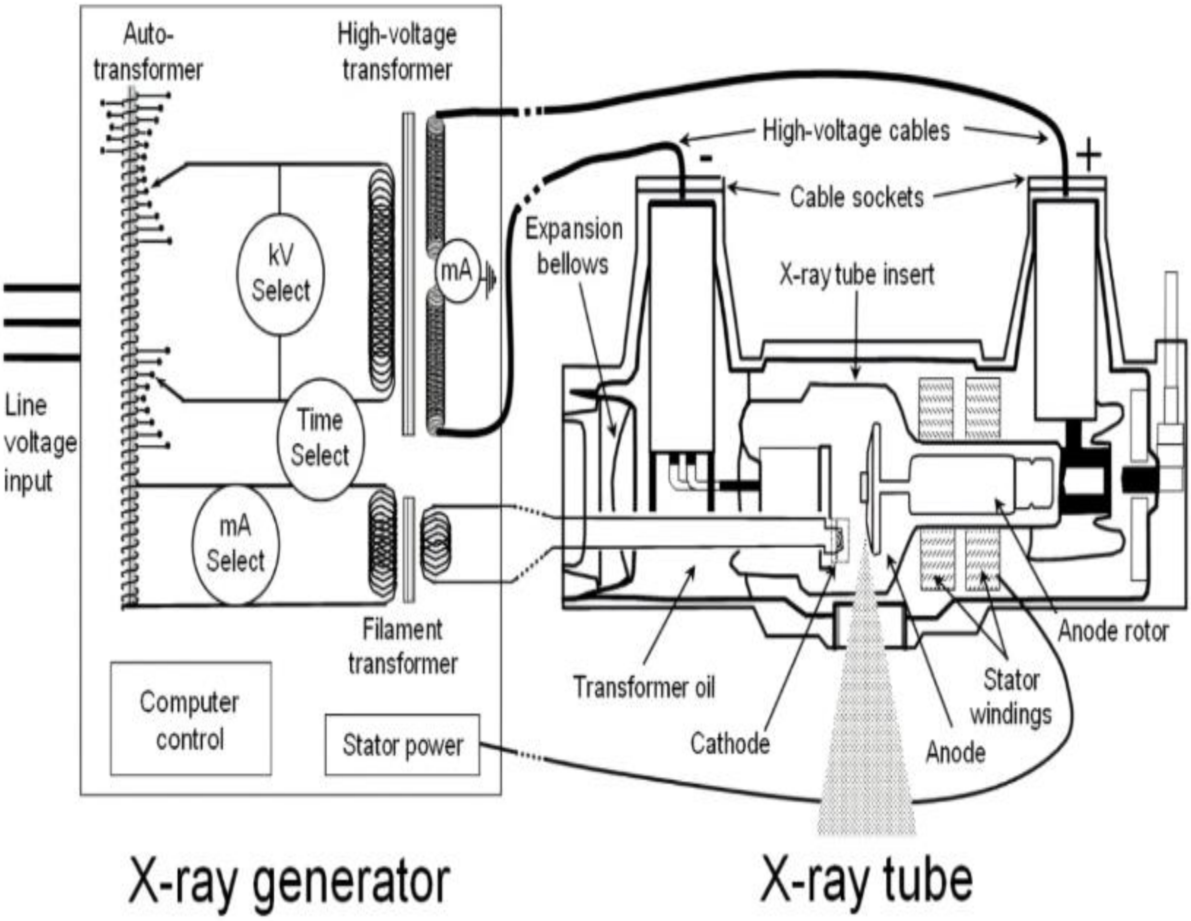


Figure (2.1): Explains the basic components of an x-ray tube and x-ray generator [38].

B. X-Ray Imaging:

In the past, X-ray films were used through the direct effect of X-rays on the chemical properties of X-ray film material, which use X-ray energy beam has the ability to penetrate materials and obtain the image formed by the shadows of those materials. While modern detection systems convert X-rays into light, and then light into electrons. Convert X-ray into visible light (images) by using X-ray image intensifiers, as shown in figure (2.2). Phosphorous is used to convert X-ray photons into light photons. The photoelectric effect inside the photocathode converts the resulting light into electrons. An optoelectronic system is used to accelerate and focus electrons towards the phosphorous outputs. Then the electrons are converted into visible light to be captured on TV camera tubes or film materials [39].

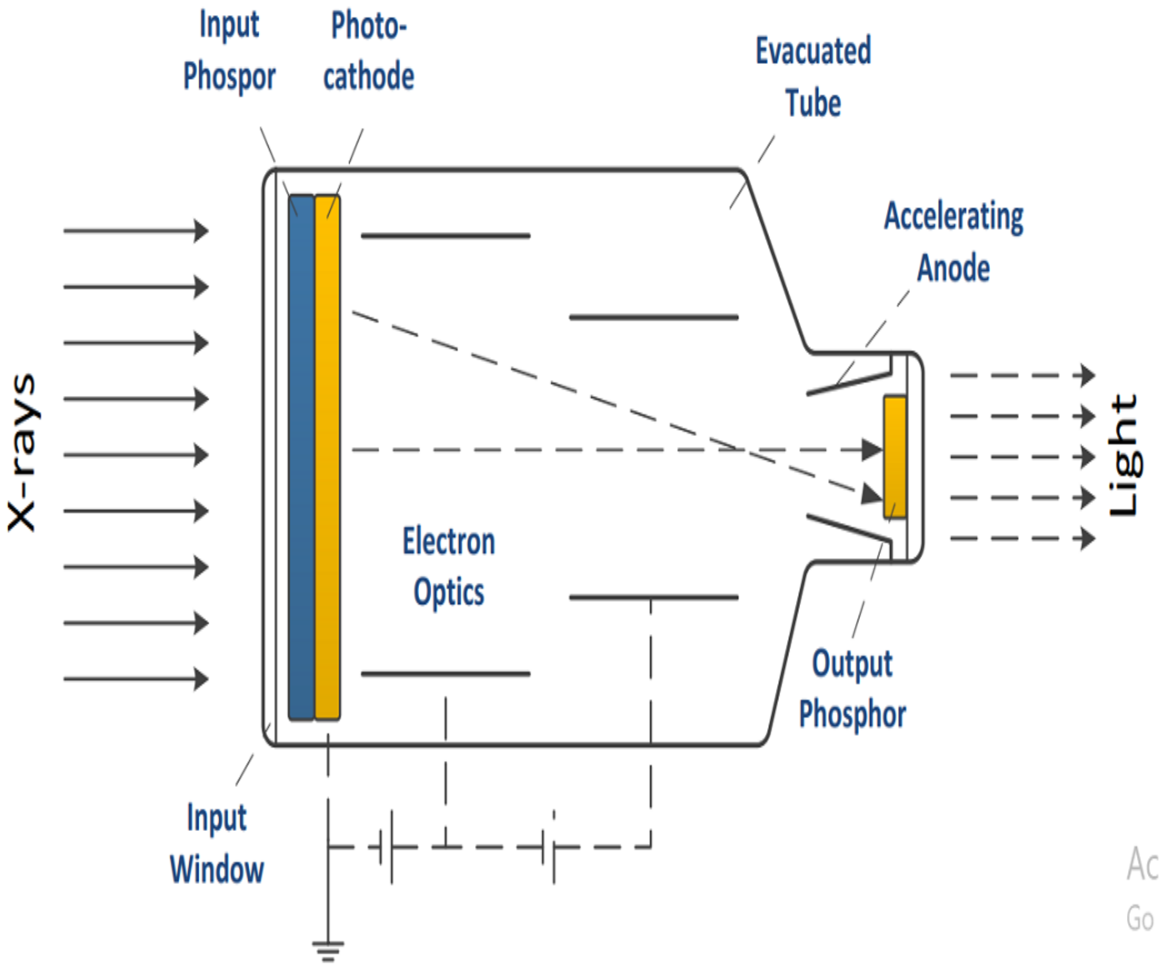


Figure (2.2): X-ray transformations to get the image [39].

X-ray image-formation testing based on energies is either using a single-energy beam and producing grayscale images or using one or more dual-energy beams that produce false color images. Attenuation of the radiant energy of X-rays during the process of image formation according to the law of absorption in equation (2.3).

$$I = I_0 \, exp(-\mu Z) \qquad (2.3)$$

where I is the captured intensity, Z is thickness, I_0 is the emitted intensity, μ is the attenuation coefficient [5].

2.4 Machine Learning

In recent years, machine learning has become one of the most widespread techniques in computer science. Machine learning is divided into several fields according to the nature of data label, such as supervised learning and unsupervised learning as shown in figure (2.3)[32].

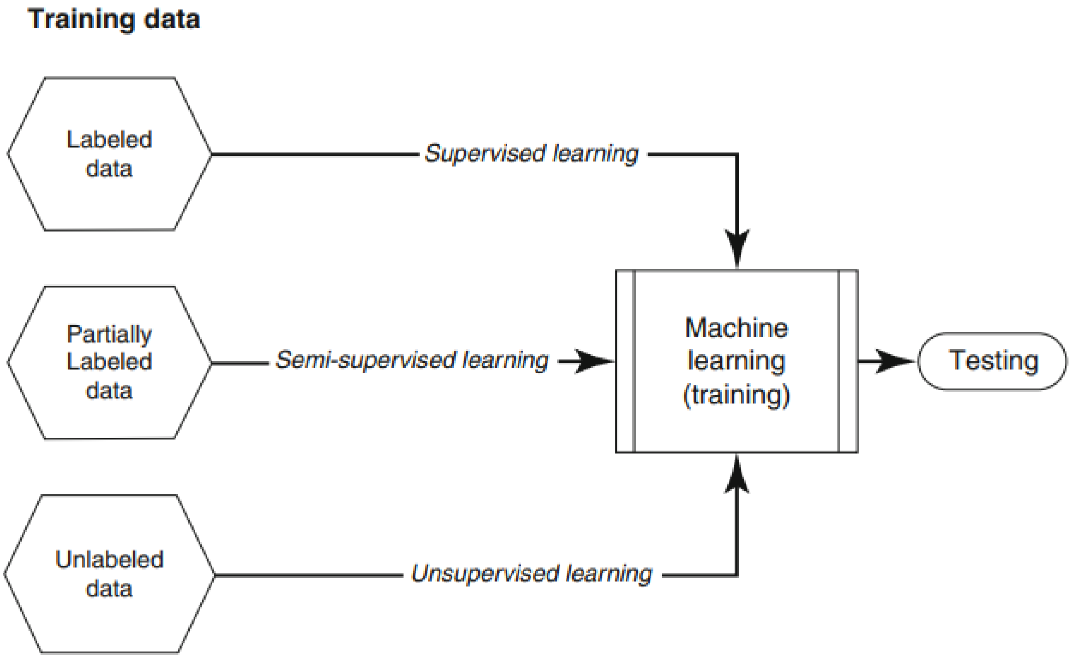


Figure (2.3): Categories of machine learning algorithms according to training data nature[32]

Supervised learning and Semi-Supervised Learning are divided into two parts: classification and regression, while unsupervised learning is only clustering[32, 40, 41]. Figure (2.4) shows three fields of machine learning.

2.4.1 Supervised Learning

Supervised learning is a description that describes the prediction task of classifying an object of interest, such as classifying an object as dangerous or non- dangerous[17]. This means that supervised learning is the design of algorithms that can produce general hypotheses and patterns using an asymptotic status that is provided in advance to predict a future status (all images contain a target). Training machine learning models is done through training datasets. The validation data set is used to valid the performance of the model, to adjust the features and get the best performance that fits the training data. Final evaluation of the performance of machine learning models using the test dataset. An example of classifying images into three categories of (Hundgun, Fruit knive and Wrench) another example of classifying e-mail into spam or not spam [18, 19,[42].

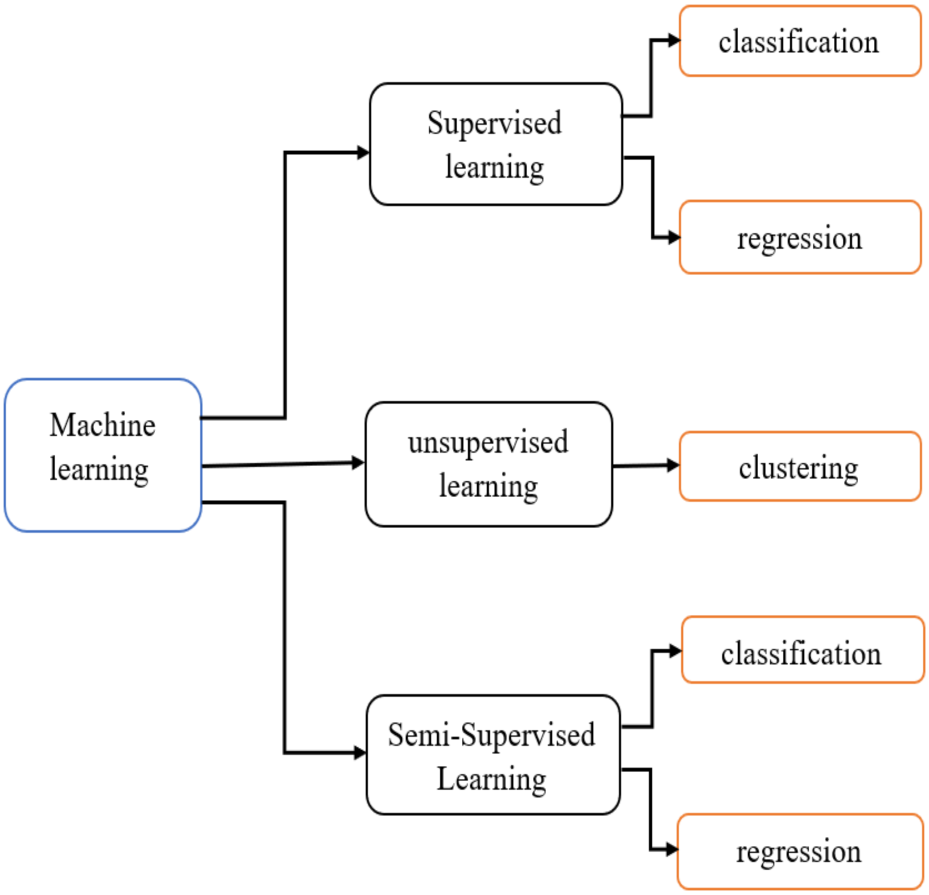


Figure (2.4): Three fields of machine learning

2.4.2 Classification

It is the most used type in machine learning. In this type, the input is classified into two or more types. The goal of the learning process is to produce a model that classifies any new income into one or more of [32]task of classification, such as logistic regression (LR), k-Nearest Neighbor (KNN) and artificial neural network (ANN)[43, 44]. Figure 5 shows a binary classification task using 30 training samples, 15 training samples are categorized as negative and 15 as positive. The data is two-dimensional, each sample contains two values of x1 and x2 using the supervised learning algorithm to learn a rule- the decision boundary represented as a dashed line and to separate the two samples. Each class is classified according to the values of x1 and x2,as shown in figure (2.5) [45].

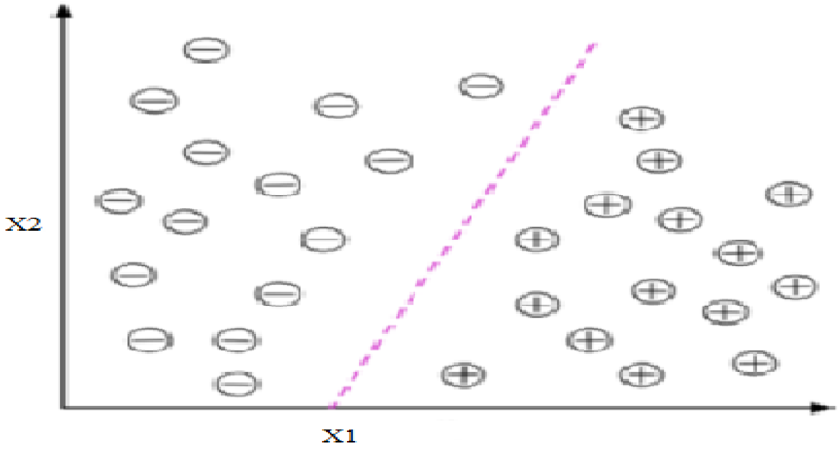


Figure (2.5): Binary classification between two classes [45].

1. Logistic Regression (LR)

Logistic regression is a simple classification model, hence its frequent use. It is based on conditional probability p("x")=p(y=1\X=x). The expected variable output y=1 depends on the input variable X [46]. In multi-classification, the training data set consists of images as a pair [44]. yϵ (1,2…S) where S number of possible classes, xi features of the input images, and y is the target value. Using the Softmax function for multi-classification, which distributes a probability as a complete vector p, will make the probability values within the interval (0,1). The Softmax is shown in equation (2.4) [44].

$$h(z)_s = \frac{e^{Z_s}}{\sum_{i=1}^{S} e^{Zi}} \qquad (2.4)$$

Where Zs values are the elements of the input vector, with sϵ y= (1, 2... S). The outcome of prediction equals 1 if s=y, and 0 otherwise. The learning objective function shown in equation (2.5) [46].

LSR [x,y] = -log py *(2.5)*

Using a parameter and a Coefficient that leads to improved prediction of logistic regression, as shown below:

Coefficient C: Inverse of regularization strength must be a positive float. The reduction in C leads to n increase in regularization [47]. L1_ratio: It is a parameter that controls the effect of each of L1(penalty term) and L2 (penalty term). The value of L1_ratio is between zero and one. If L1_ratio value is zero then only L2 is applied, while if the L1_ratio value is 1 then only L1 is applied. Any number between zero and one indicates the combination of both (L1 & L2) in certain proportions [48].

2. K-Nearest Neighbor (KNN)

KNN is a simple and effective algorithm that stores the available states and depends on calculating the Euclidean distance (d) in classifying new states as in equation (2.6). The algorithm saves similar objects to classify new samples by measuring similarity, and the categories are determined on the basis of the nearest neighbor. The performance of sampling classification depends on the value of K mostly. Increasing the value of K makes the performance better[49-52].

$$d = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \qquad (2.6)$$

Where x_i is input sample, y_i is new sample.

3. Artificial Neural Network (ANN)

ANN is a machine learning algorithm that simulates the idea of neurons in the human brain. It is considered one of the fast algorithms with the possibility of development and has the ability to deal with non-linear relationships. The algorithm consists of three basic components: the nodes, the network and the learning method. Each node is linked to a set of inputs via links that have weights. If the sum of the inputs exceeds the threshold values, the signal will pass to the neighboring nodes by the transfer function. The math process is done in equation (2.7) and as shown in figure (2.6).

$$y = f(\sum_{i=0}^{n} w_i x_i - T) \qquad (2.7)$$

where y is the output of the node, f is the transfer function, w_i is the weight of input x_i, and T is the threshold value [53, 54].

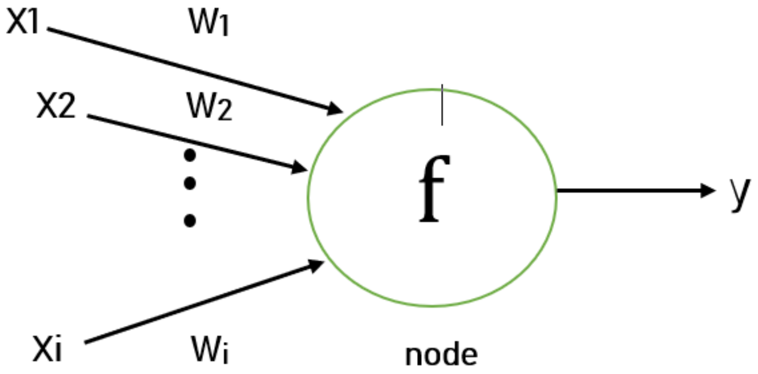


Figure (2.6): single node: x_i = input, w_i = weight, f = transfer function, y = output [54].

The network contains several layers and each layer contains several nodes. Neural networks are divided according to the feed into two parts: one-way forward feed and one-way feedback. The weights are adjusted during the network training on the data and reduce the error between the output and the desired value. To evaluate the network performance, the difference between the output of the node and the target embarrassment is calculated and called the error function as in equation (2.8) [53, 55].

$$e_k = y_{k,n} - y_k^* \tag{2.8}$$

Where e_k error function, $y_{k,n}$ output of the k th output node at iteration n, y_k^* target output for the k th node [54].

2.5 Deep Learning (DL)

DL is one of the expanded and sub-fields of machine learning and it is more in depth by increasing layers and the representation of data in a hierarchical manner at more than one level. DL has the power to learn which features are automatically extracted from the raw data. Deep learning has high speed and good performance in solving complex problems such as reducing logistic regression error and obtaining high classification accuracy if the data is large [56, 57]. Figure (2.7) shows the architecture of a deep neural network (DNN) that contains many hidden layers[58].

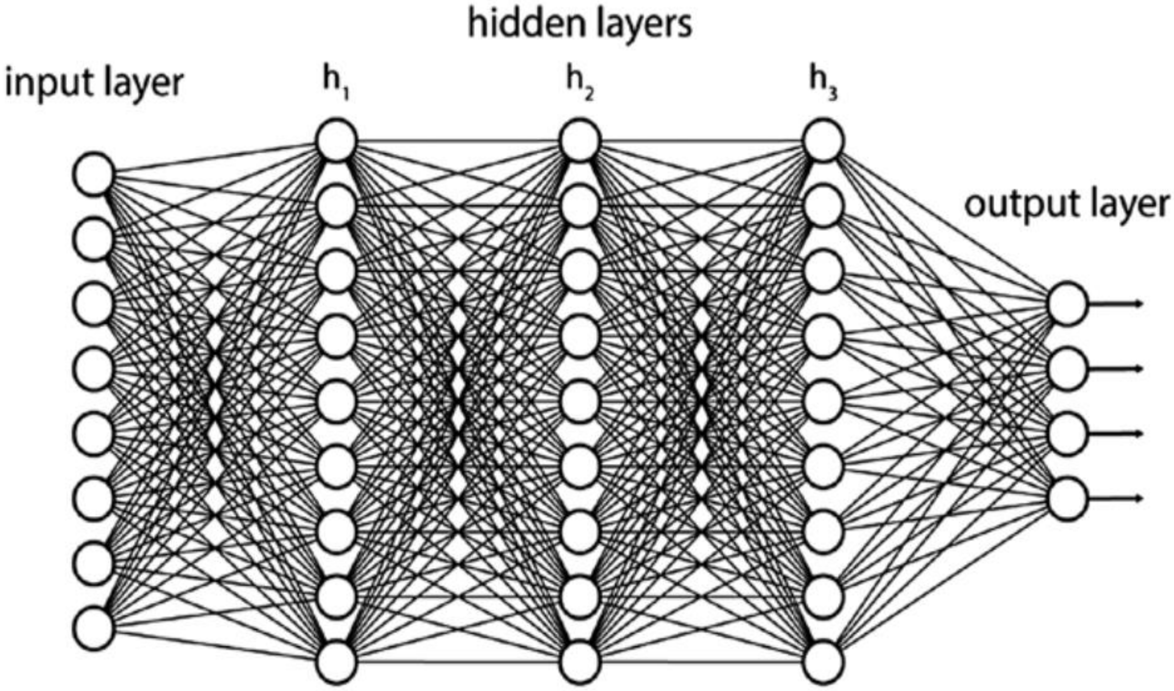


Figure (2.7): Deep neural network [59] .

2.5. 1 Deep Learning Performance Enhancement

This section explains the problems that deep learning encounters, and the methods that can be used to reduce these problems and improve the performance of deep learning in image processing.

2.5.1.1 Overfitting Reduction

Overfitting is a problem in which the classifier has a high accuracy on the training set while the accuracy is low for the classifier on the test set [60, 61]. One way to reduce overfitting is to use data augmentation. Data augmentation is one of the simplest and most widely used methods to reduce the problem of overfitting by increasing the training images [62]. Several techniques can be applied during this process, such as changing the lighting, zooming, horizontal flipping, vertical flipping. rotation and random cropping [63]. The augmentation in the image data demonstrated a higher performance during the training of the convolutional neural network (CNN) algorithm, especially when the data is limited [64].

Data augmentation is used to obtain a larger data set during the training phase of powerful neural networks. Data augmentation is one of the solutions to the problem of lack of data, as data augmentation is based on enhancing the quality and size of training data images and thus improving the performance of deep learning networks. One of the methods of processing mixed-content image data is to crop images: a central part of each image is crop to change the dimensions, width and height of the images. Cropping reduces the size and enhances the image. Another way to augmentation the image dataset during training is to rotate the angles between 1 and 359 degrees [63]. For the example of anomaly detection, Figure (2.8) shows the effect of the augmentation on the training data to improve accuracy [65].

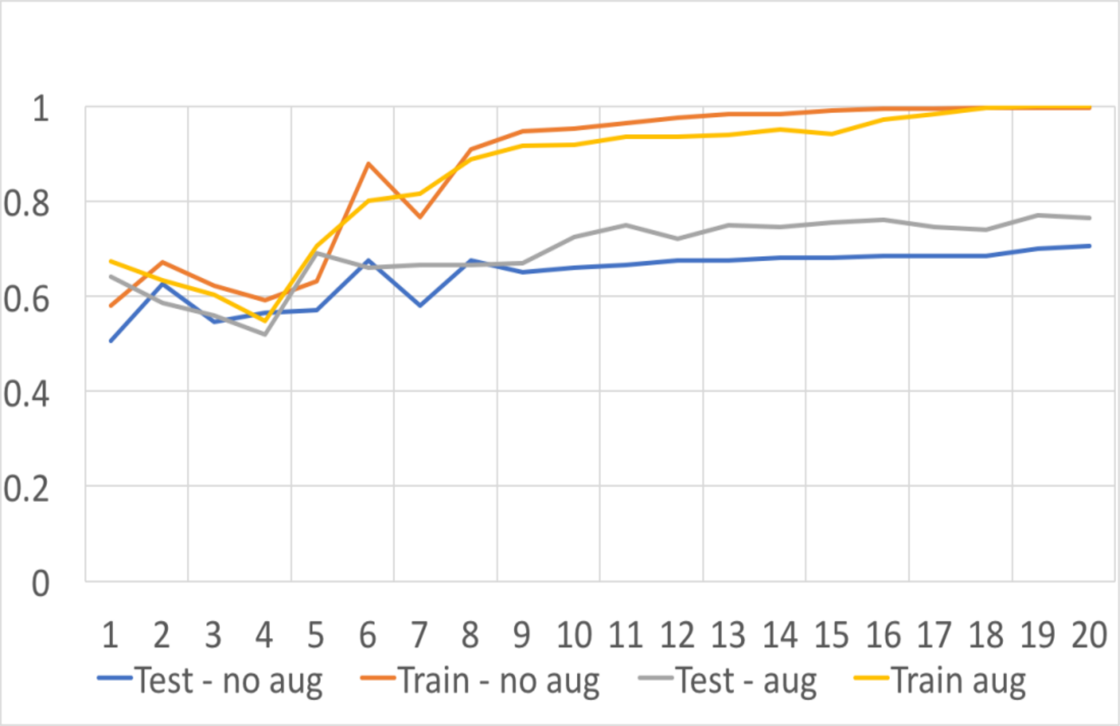


Figure (2.8): Accuracy plots [65].

2.5.1.2 Test Time Augmentation (TTA)

(TTA) is the process of selecting random parts of the original images (N) several times and these are collected and extracted the average feature and the maximum feature [66]. TTA is used to increase the data set during testing. These test images are augmented in a similar way to augmentation the data set during training such as rotation and translation to get better prediction. The importance of the test appears clearly on the data set that contains more than 500 images. The improvement in model performance is less when only augmentation data is used during training. When using data augmentation during training and testing, the improvement in performance is better for classification algorithms [63, 67].

Deep learning model is more effective by using data augmentation during testing. The DenseNet algorithm achieved the highest accuracy using TTA. The use of TTA in medical image segmentation performed better than the dropout-based multi prediction technique and the single-prediction baseline technique. TTA achieved a better performance for the problem of high confidence in the distribution of outliers, that is, it reduces the predictions that are classified as true even though they are false predictions because TTA extracts many image variables during the test of many predictions of the same image. When using deep learning algorithms, the use of image segmentation by TTA leads to model optimization. [63, 68].

The effect of TTA is very high in improving the performance of poorly efficient classification models, while it is less effective in improving the performance of models with high accuracy. The use of TTA does not affect the cost of running the model because the use of TTA is after the training process, but it requires additional time [69]. The following example in figure (2.9) shows the effect of data augmentation and test time augmentation in improving accuracy for dataset of medical images (skin lesion) [63].

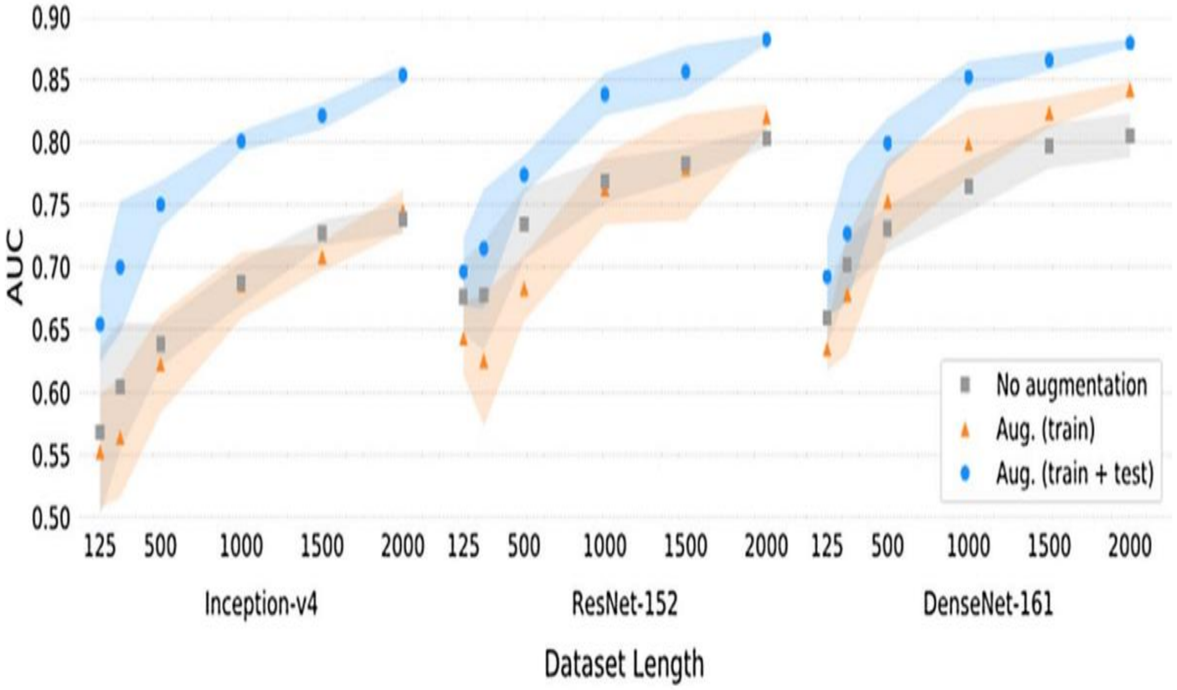


Figure (2.9): Impact of test-time data augmentation for skin lesion classification [63].

2.5.1.3 Transfer Learning

The purpose of its use is to improve the performance of the model. Transfer Learning can be useful in solving the problem of lack of data by taking advantage of a previously trained neural network. Convolutional neural networks containing huge data that were previously trained to classify images such as AlexNet are used to benefit from it during training of the new network. The AlexNet network was trained on a huge data set called an Image Net, which contains approximately 1,000,000 images and 1,000 classification categories. The learning transfer approach adjusts the upper layers of the network to take advantage of the properties related to secondary features, while the first layers remain constant because they provide the general features. Pre-training of CNN networks is used to reduce computational cost because training from zero takes a lot of time [12, 70].

2.6 Convolutional Neural Networks (CNNs)

(CNNs) are a modern version of multi-layer neural networks, in recent years, it has become widespread in many fields such as computer vision, speech recognition and the healthcare field [71-73]. It has achieved remarkable results in many computers vision tasks such as object detection and classification. CNNs consists of four main layers, namely the convolution layer, pooling layer, activation layer and fully connected layer [10]. Figure (2.10) shows the feature extraction and classification layers of a convolutional neural network [74].

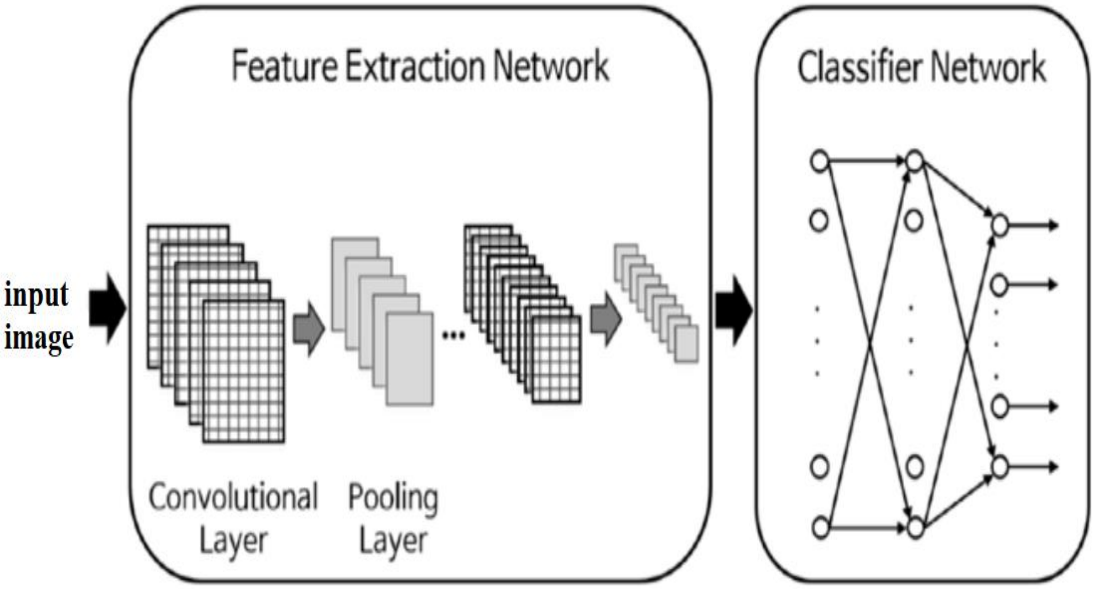


Figure (2.10): Typical architecture of CNN [74].

2.6.1 Convolution Layer

It is the first layer through which features are extracted by the convolution process. Convolution layers contain a large number of digital filters. Using the convolution process turns the input image into new images called feature maps. Feature maps show the salient features of the original image. The convolution layer is distinguished from the rest of the CNN layers by its use of filters instead of weights, each convolution layer filter creates a single feature map. Convolution filters are two-dimensional or more arrays. Through training, filter matrix values are determined in a similar way to updating weights in simple neural networks, the convolution filter values are constantly updated throughout the training phase [75].

Assume that the original image is A, the filter (kernel) is K, A of dimensions $n_A \times n_{A,}$ kernel K of dimensions $n_K \times n_K$ and (i,j) position. By using the above parameters, the map features (B) can be calculated using equation (2.9) [76].

$$B_{ij} = (A * k)_{ij} = \sum_{f=0}^{nk-1} \sum_{h=0}^{nk-1} A_{i+f,j+h}\ K_{i+f,j+h} \qquad (2.9)$$

The dimensions of the matrix($n_{B,}$ n_B) after the convolution can be calculated from equation (2.10).

$$n_{B=}\left[\frac{n_A - n_k}{s} + 1\right] \qquad (2.10)$$

where(s) is stride (stride is the number of steps the kernel moves over the original image matrix)[76]. Figure (2.12) shows the calculations of the convolution layer to convert the original image to the map features and the dimensions of the new array. The first pixel is calculated by using the equation 10 $(7*1+2*0+3*^{-}1+4*1+5*0+3*^{-}1+3*1\ 3*0+2*^{-}1 = 6)$. In the same way, with the kernel moving one step, because stride is one, the values of the rest of the pixels can be calculated. The dimensions of the resulting matrix are calculated using the equation (3.11), The number of rows is two and the number of columns is three [77].

Padding is used to standardize the sizes of images between the input layers and the output layers. The use of padding improves the performance of deep network models because it makes training better, but it increases training time because it increases the input sizes of the convolutional layers. The use of padding raised the classification accuracy to more than 2% [78]. The number of output layers (feature map) after applying the padding can be calculated by applying equation (2.11) [60].

$$n_{f}=\frac{n_A+2P-F}{P}+1 \qquad (2.11)$$

Where n_f is feature map, p is padding.

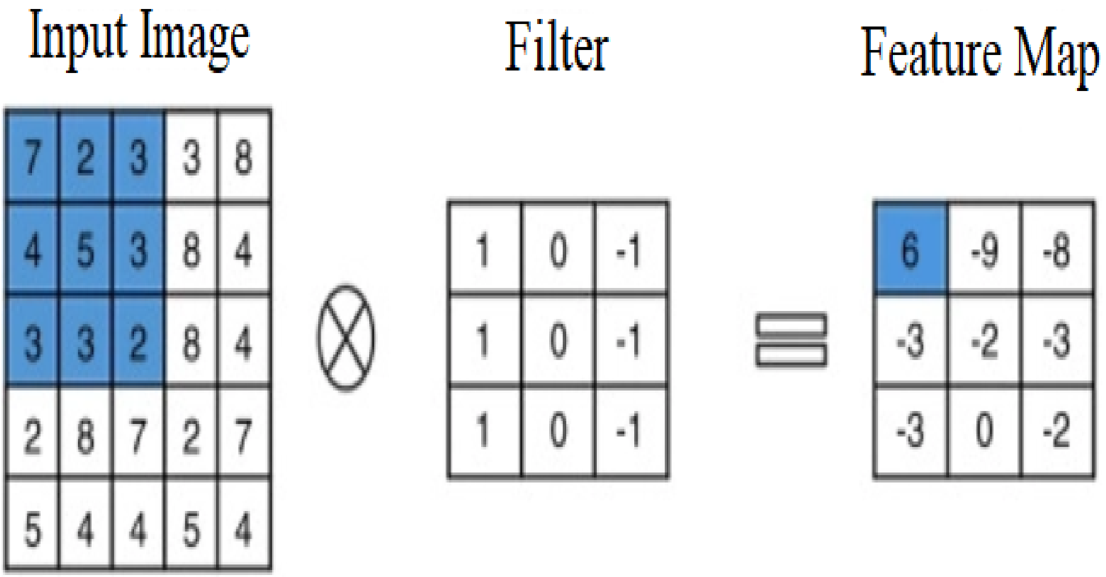


Figure (2.11): Convolution Layer Operation [79].

2.6.2 Pooling Layer

It is the second layer of CNN that extracts features and it is a 2D layer. The pooling layer down samples adjacent pixels and converts them to a single pixel, this reduces the image dimensions. Through aggregation classes, representative values are defined as maximum or mean. To illustrate the assembly process, 4 * 4 pixel input images are selected with the use of a 2 * 2 pixel grouping layer, as shown in figure (2.12) [74]. To calculate the maximum pooling (ρ_m), equation (2.12) is used.

$$\rho_m = \max_{i=1\cdots k} a_i \qquad (2.12)$$

To calculate the Mean pooling (ρ_{av}), the equation (2.13) is used.

$$\rho_{av} = \frac{\sum a_i}{N} \qquad (2.13)$$

Where N is number of elements, a_i the values of the elements in the input array.

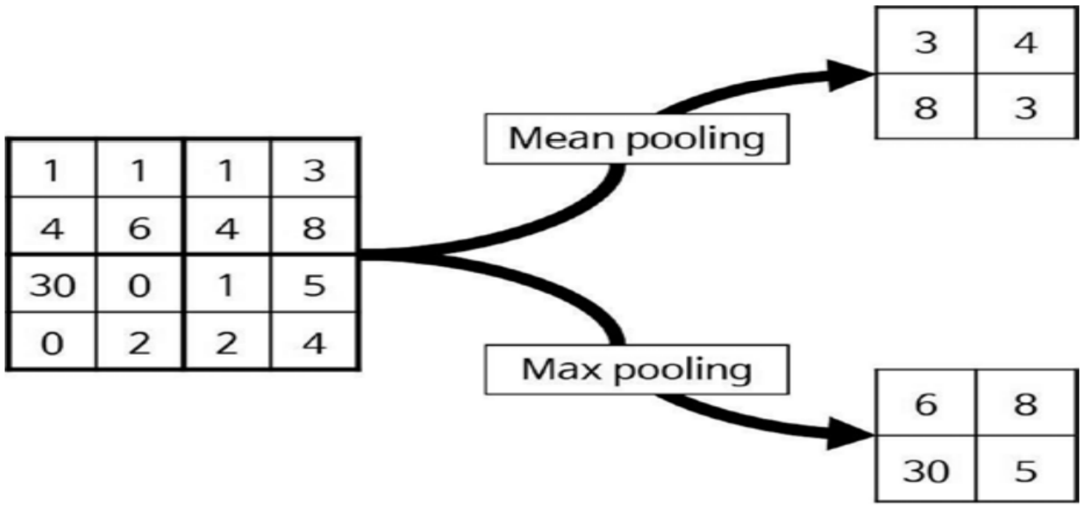


Figure (2.12): The resultant cases of pooling using two different methods (Mean and max) [74].

2.6.3 Activation Layer

After the convolution layer, the features are entered into the activation layer. There are many different kinds of activation functions and they all pass complex features. Activation functions allow the information to be transmitted to the next neuron, and the activation layer is always between two layers of CNN. Figure (2.13) shows the general structure of the activation function.

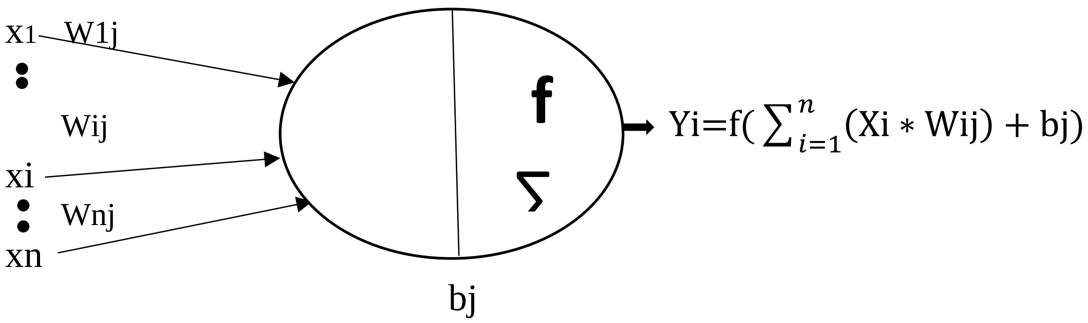


Figure (2.13): General activation function structure [80]

Where: xi represents the input feature, j at the same time, n features are input to the neuron, bij represents the internal state of the neuron j, wi j represents the weight value of the connection between the input feature xi and the neuron j, and yij is the output of the neuron j. f (·) is the activation function [80].There are many kinds of activation functions such as rectified linear unit (ReLU), sigmoid function, and Softmax Function [80, 81].

2.6.3.1 ReLU Function

ReLU is a standard and powerful feature of deep learning networks . The output of the ReLU is a linear function, the principle of action of the ReLU is shown in the mathematical expression in equation (2.14). The f(x) of the ReLU will be zero for all x-values less than zero and the f(x) will be x for all x-values greater than zero. Figure (2.14) shown the ReLU function [82, 83].

$$\text{ReLU(x)=max(0,x)} = \begin{cases} x, & if \;\; x > 0 \; Active \\ 0, & if \;\; x < 0 \;\; Inactive \end{cases} \qquad (2.14)$$

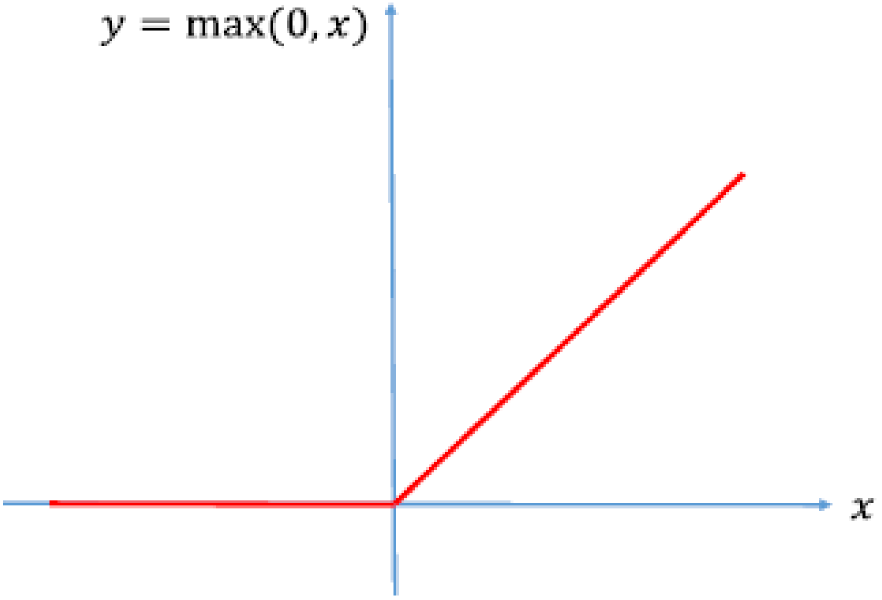


Figure (2.14): The ReLU Function[83]

2.6.3.2 The Sigmoid Function

It is one of the most commonly used activation functions that gives values in the period between zero and one by applying equation (2.15).

Frequently used in prediction models where the output file is probability dependent, given that the probability assumes the values within the specified period between zero and one. Figure (2.15) shows the sigmoid activation function, which is in the form of the letter S-shape, starting with zero and ending with one. The sigmoid function is a non-linear function and the prediction accuracy is high in binary classification [76, 84].

$$\sigma('\Upsilon) = \frac{1}{1+e^{-\Upsilon}} \qquad (2.15)$$

Where $\sigma('\Upsilon)$ is sigmoid function, $'\Upsilon$ is total input to the hidden neuron.

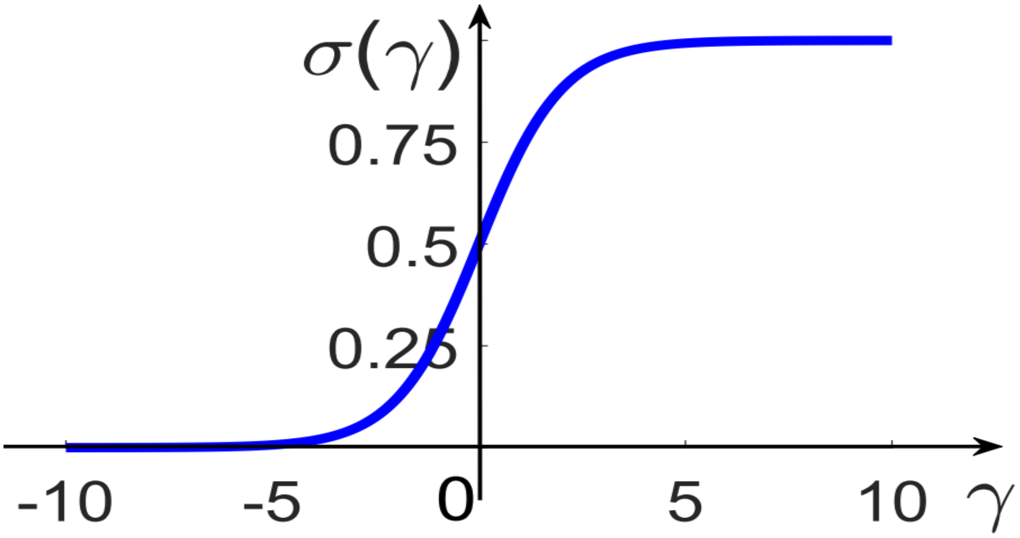


Figure (2.15): Sigmoid function [84].

2.6.3.3 The Softmax Function

It is an activation function frequently used in deep neural networks. It is a generalization of the sigmoid function and is nonlinear [85]. The Softmax function is widely used in multi classification, so it is used in the outer layers of deep neural networks, All output values of the Softmax function are between zero and one, so it expresses the probability. It is used in the output layer to convert the product of linear values into a probability product between zero and one, by applying equation (2.16) [86, 87].

$$\sigma(x_c) = \frac{e^{x_c}}{\sum_{j=1}^{C} e^{x_j}} \qquad (2.16)$$

Where C represents the number of classes, x is a C-dimensional input vector with arbitrary real-values.

Figure (2. 16)shows the softmax function and how to classify when the output is categorical [81].

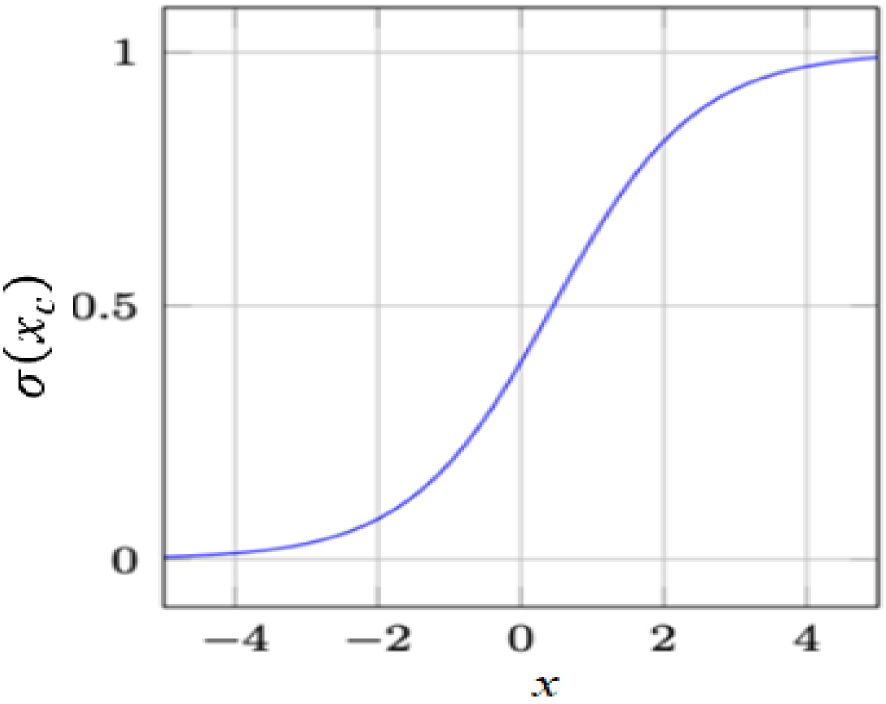


Figure (2.16): Softmax function [81]

2.6.4 Fully Connected Layers FC

It is necessary to use fully connected layers because they constitute the bulk of network parameters. This huge number of parameters is trained inside fully connected layers, in order to fit complex nonlinear functions within the feature space in which the input data elements are defined. For a better performance of the CNN network, the number of neurons in the full connection layers must be increased. Figure (2.17) shows the effect of fully connected layers on the depth of a CNN network. [88].

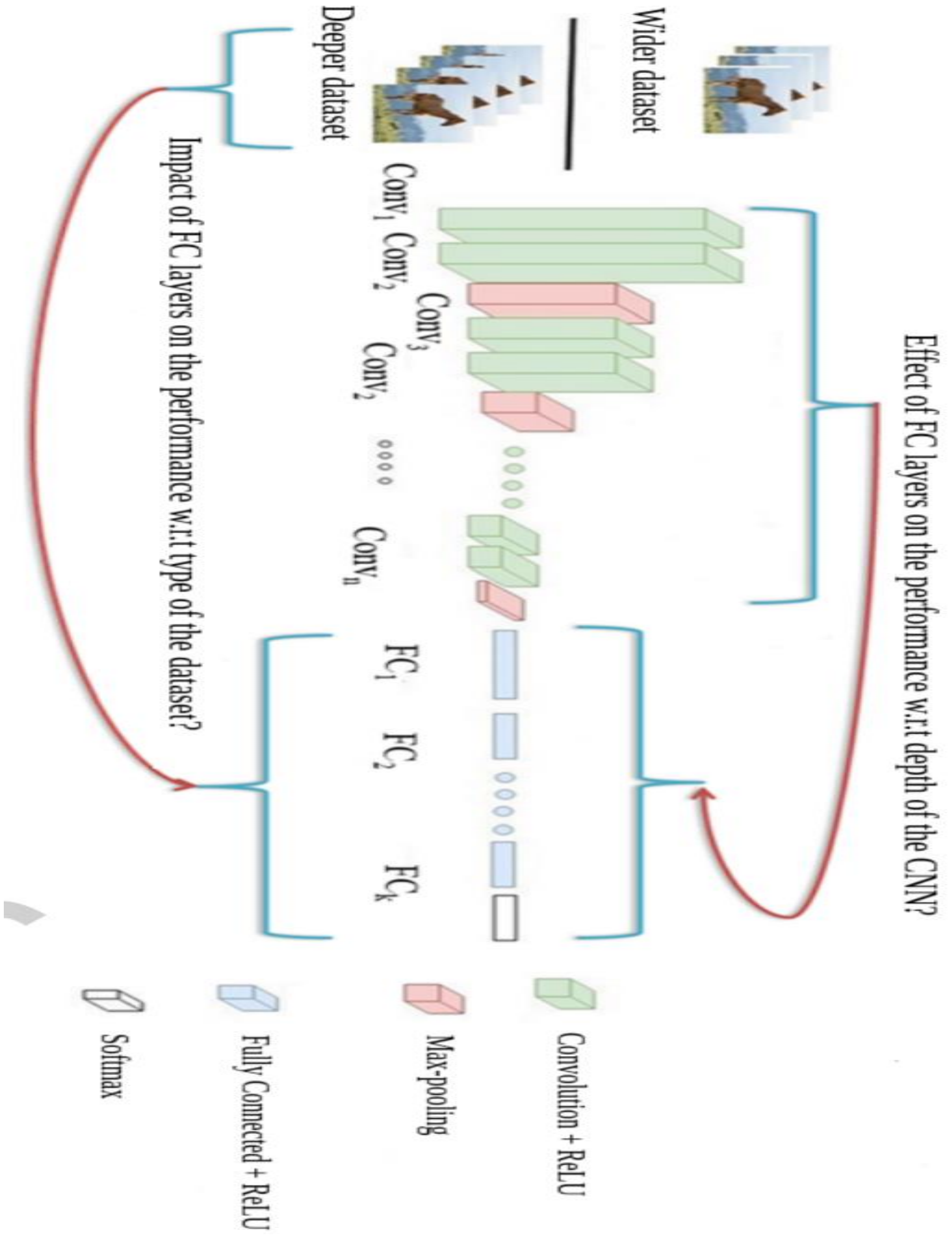


Figure (2.17): The illustration of the effect of deeper/wider datasets and depth of CNN [88].

2.6.5 Loss Functions

After the final prediction of the classification task which is the output layer of the convolutional neural network structure, errors must be known to evaluate the model. The loss function calculates the difference between the actual output and the predict output to find out the value of the error. One of the loss functions is the Cross-Entropy function.

Cross-Entropy function is a function by which the performance of a convolutional neural network model is evaluated. Used in multi-category classification in the outer layer Softmax, its value is the probability p ∈ {0,1}[89]. Using Equation (2.17), the result is calculated within probability.

$$P_i = \frac{e^{a_i}}{\sum_{k=1}^{N} e_k^a} \qquad (2.17)$$

Where N represents the number of neurons in the output layer, e^{a_i} represents the non-normalized output from the preceding layer.

Equation (2.18) represents the mathematical formula for the Cross-Entropy loss function [89] .

$$H(p, y) = -\sum_i yi \log{(pi)} \qquad (2.18)$$

where i ∈ [1, N]

2.7 Convolutional Neural Network Architectures

There are many of architectures of (CNN) such as AlexNet [60]. VGG16 [90] GoogleNet [91]. and DenseNet121 [47].

1) AlexNet

The Alex network contains eight layers, from Layer 1 to Layer 5, which are convolutional layers. The last three layers are the fully connected layers. The size of the image entered into the network is (227 * 227 * 3) pixels. In the first convolution layer, 96 filters of size (11*11) pixels are used with (stride=4). After applying this processing, we get 96 feature maps, that is, they are equal to the number of layer filters. The size of the second layer is (55 * 55) pixels, the max pooling is applied 96 filters in size (3 * 3) pixels and (stride=2). The size of the third layer is (27 * 27) pixels, another convolution layer is used with 256 filters with a size of (5 * 5) pixels with padding=2. Processing continues by passing the features of the maps to the other layers of the network, and the last layer is responsible for the final predictions of the classification process, as shown in figure (2.18).

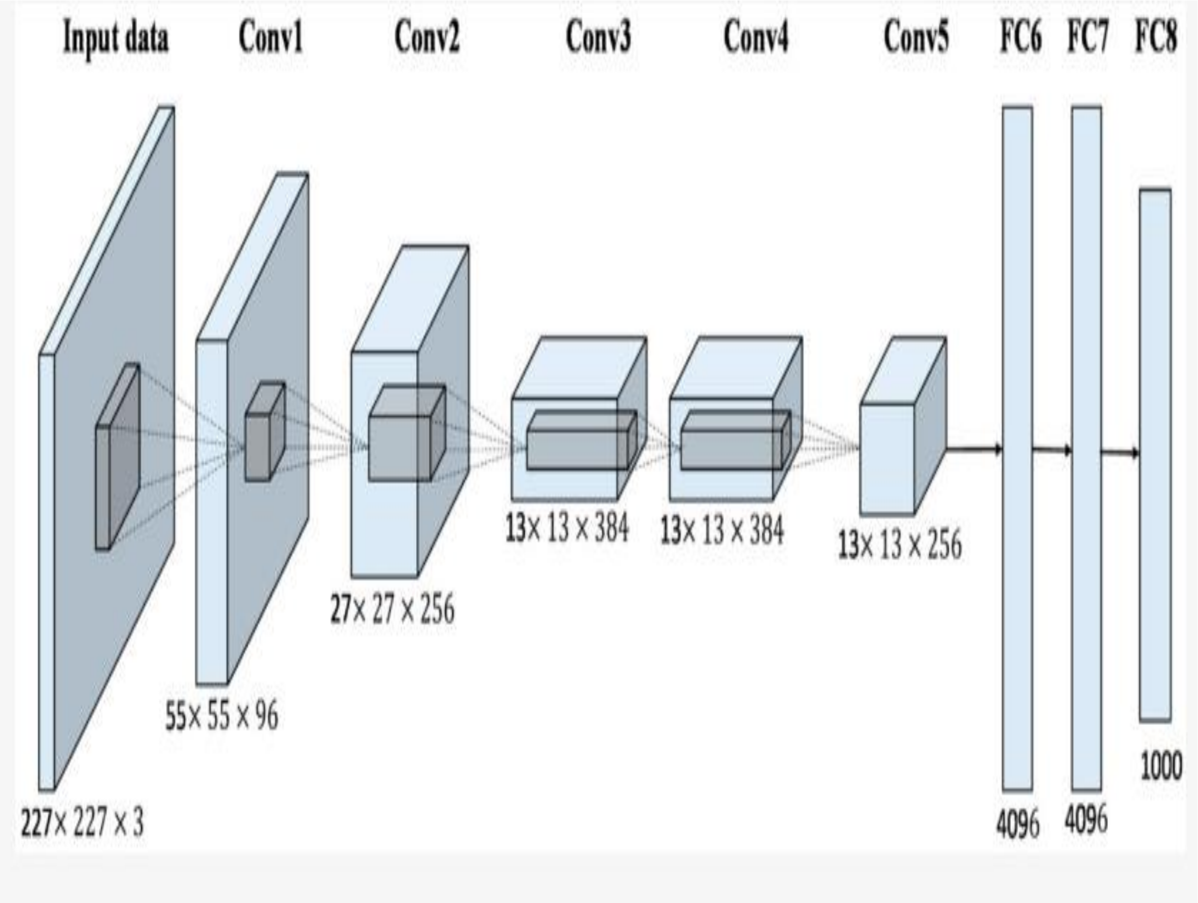


Figure (2.18): AlexNet architecture [92]

2) VGG16

A model of a Convolutional neural network is considered one of the simple and flexible networks when dealing with it because it uses small filters (3 * 3) pixels along the network and (stride=1). It consists of 16 layers, the first thirteen are Convolutional layers and the last three are fully connected layers. It contains the max pooling size (2*2) pixels and (stride=2) and activation function ReLU. In the end, the total is 41 layers. The last fully-communication layer contains the softmax function to output the final prediction of the classification process. The network was trained on more than 1 million images from the ImageNet dataset and 1,000 categories [93]. Figure (2.19) shows the types of layers that make up a VGG16network[90].

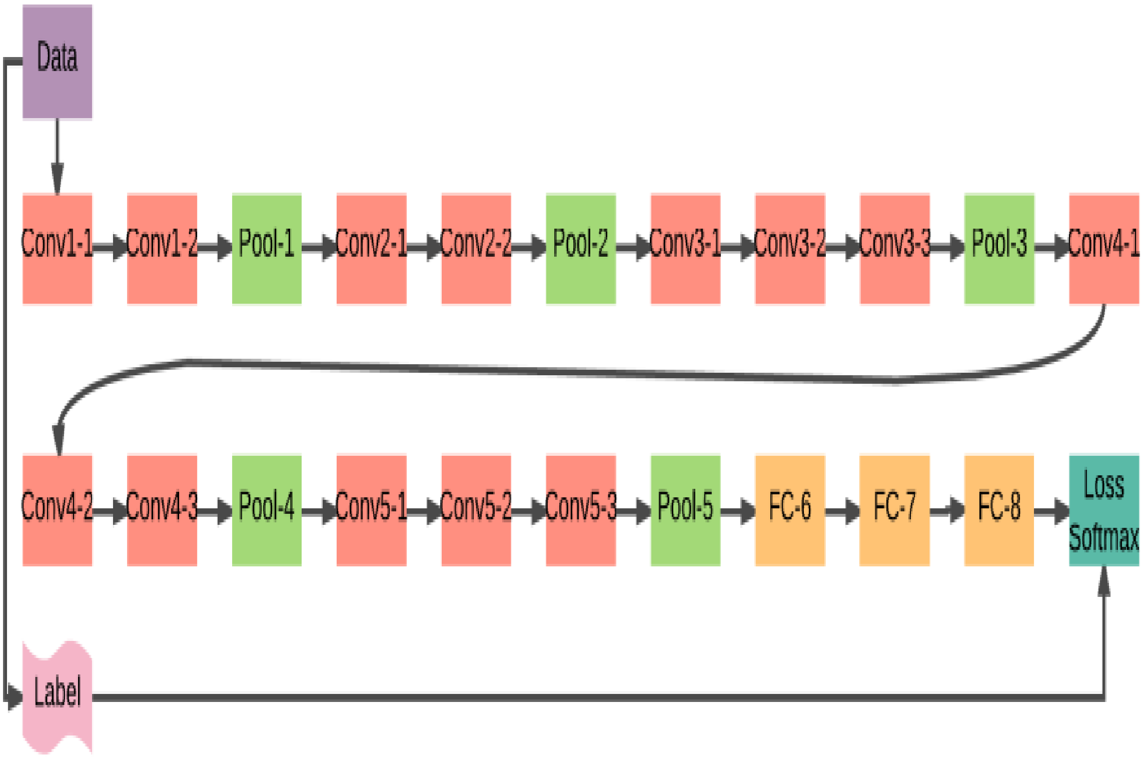


Figure (2.19): The architecture of VGG16 [90].

3) GoogleNet

The GoogleNet network consists of 27 layers divided into 22 convolution layers and 5 pooling layers. The network proposed a Google unit called inception (module). The network consists of 9 inceptions: the first inception consists of three different sized convolution layers and pooling layer, as shown in figure (2.20). The second inception consists of six convolution layers and a pooling layer, as shown in figure (2.21) [89, 91].

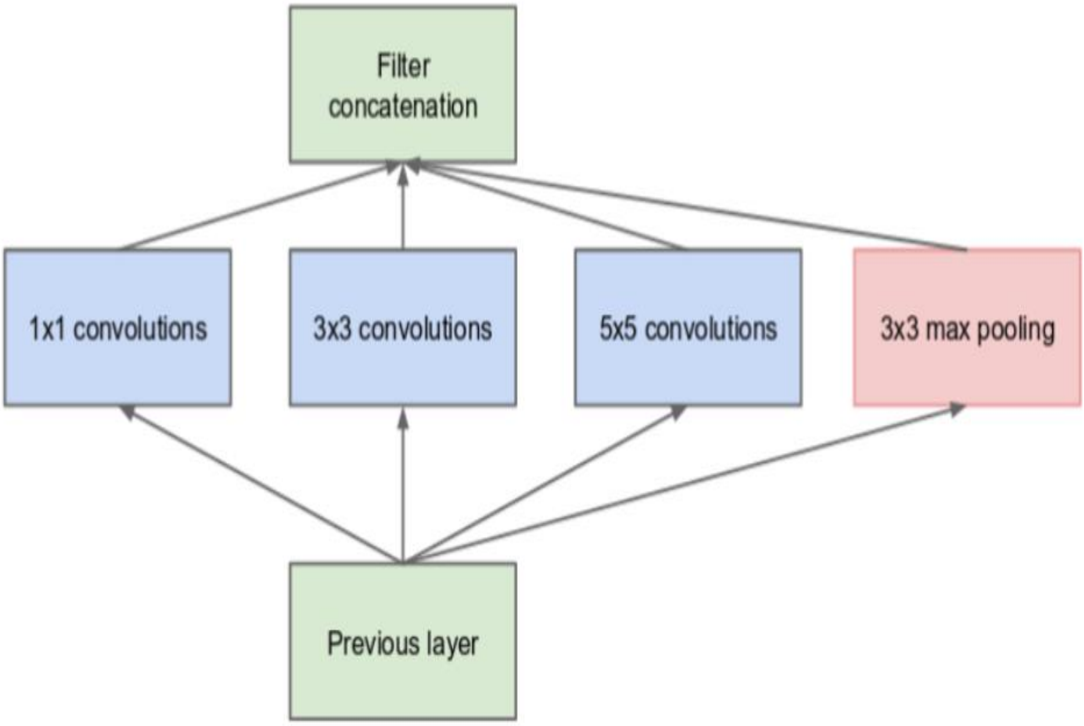


Figure (2.20): First inception module [91].

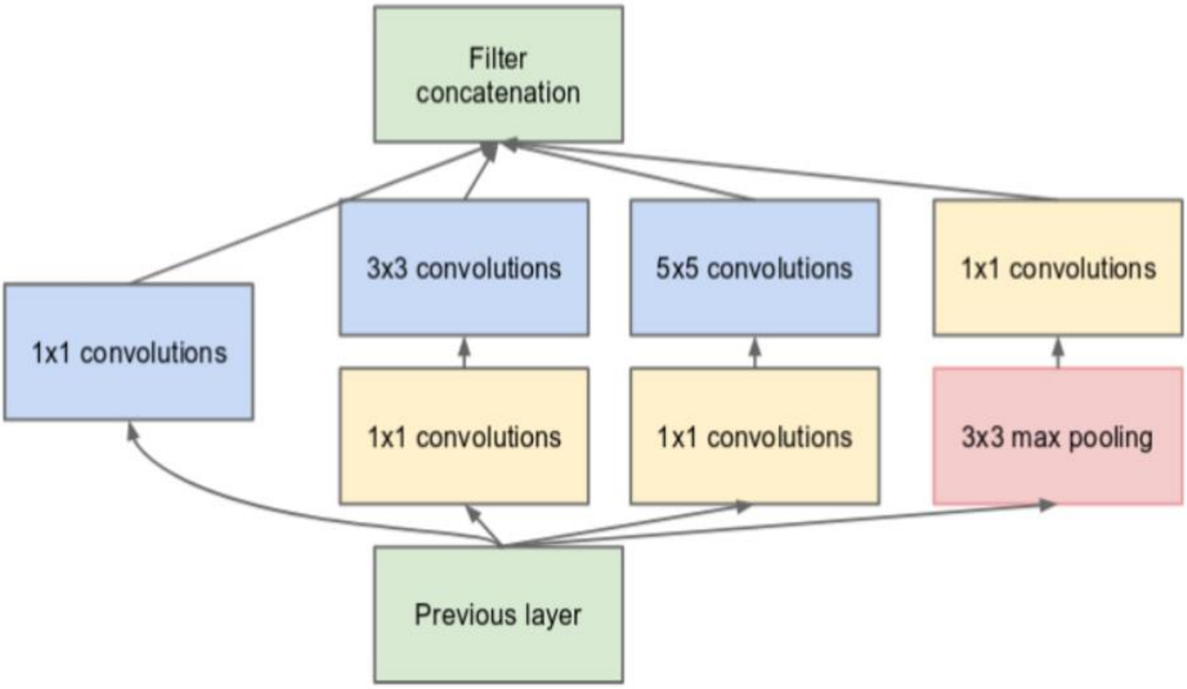


Figure (2.21): Second inception module [91].

4) DenseNet121

It is a network architecture of (CNN) that is deeper and more efficient in training using short connections between layers. The first layer is linked with all subsequent layers, as well as the second layer and so on. This enables the greatest amount of information to pass between the network layers. The network begins with two basic layers, which are the convolution layer and pooling layer, followed by four dense block and three transition layers. The DenseNet network was introduced in 2017 by Huang and colleagues [47].

The flow of feature gradients from the previous layer to the next layer using short connections between layers increases the information flow between the layers of the denseNet121 network. The layers of the network are formed to be three basic blocks. The first block is the convolution block, it is a basic. The second block is called the dense block, where the convolution blocks are sequential and connected at a high density, and it is the main component of the denseNet network. The third block is called the transition block, that connects adjacent dense blocks. Figure (2.22) shows the contents of each block such as the use of digital filters, their number and size, the use of grouping, number and size, and the use of activation functions [94].

The DenseNet121 network, during its training on the data set, succeeded in obtaining a prediction with high accuracy after applying the technique of data augmentation. While the AlexNet network and the GoogleNet network, the features extracted from them were used with the use of the KNN algorithm to get the prediction.

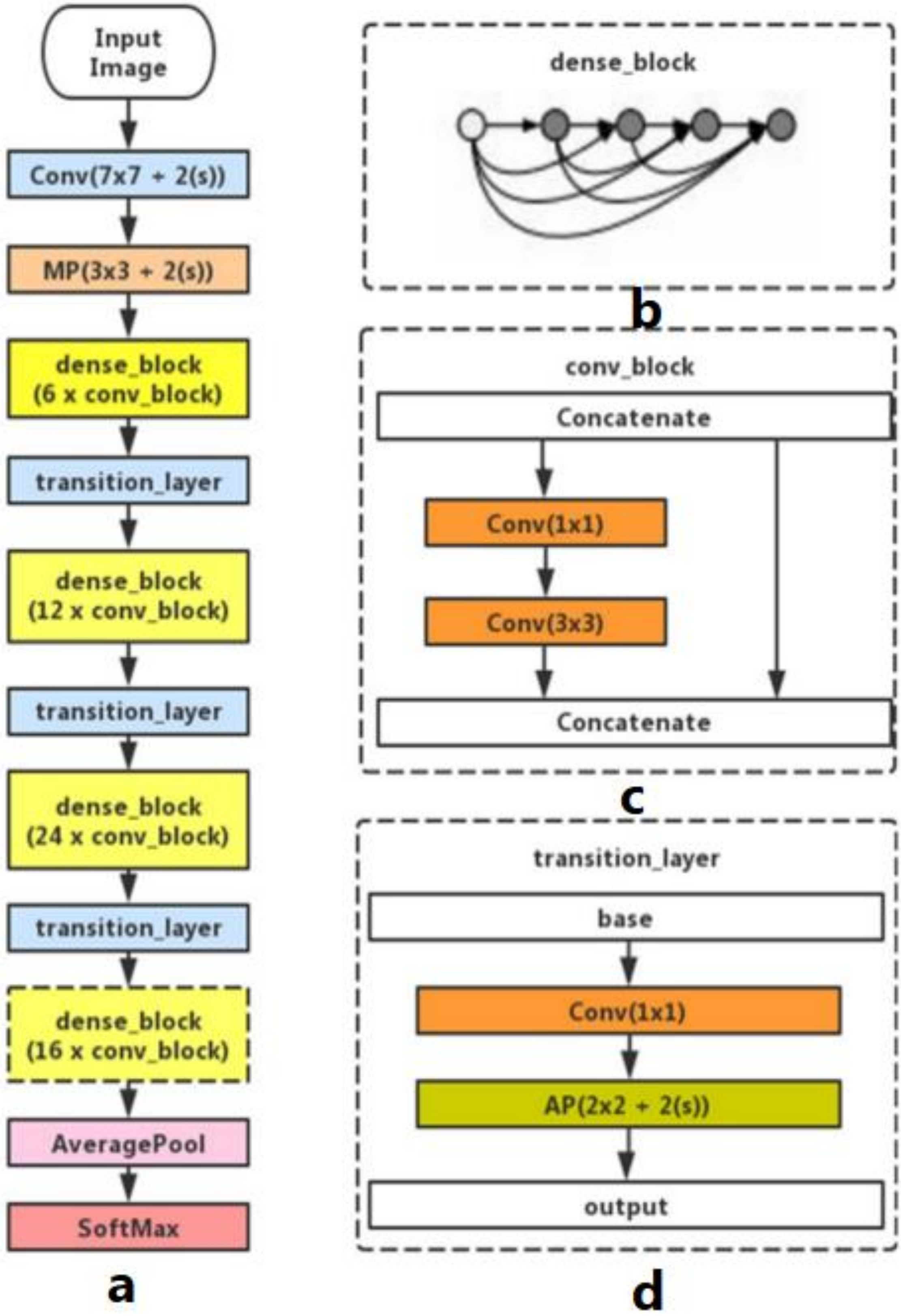


Figure (2.22): (a) DenseNet121 architecture. (**b)** Dense block. (**c**)convolution _block. (**d**) transition layer [94].

2.8 GDXray Dataset

Most of the datasets related to X-ray images baggage inspection are generally not available, as it is considered to be private. Therefore, securing authorities are not allowed to distribute them. However, the GDXray is a publicly available dataset for educational or research purposes [95]. It contains 19,407 X-ray images. It is organized and divided into five types: welds, casting, natural object, baggage, and settings. In this paper, the set of baggage images will be used which contains 8150 images. The color of the images is grey (mono-energy X-ray) [96]. Figure (2.23) shows some samples of X-ray images taken from GDXray dataset [5].

Figure (2.23): Image examples of GDXray dataset [5].

2.9 Performance Metrics of Classification

Performance metrics are used to determine the effectiveness of the machine learning model in the classification task to reach the desired outputs for the categories. There are many different performance metrics that improve the performance of machine learning models, As shown below [97].

2.9.1 Accuracy (acc)

The accuracy metric measures the ratio of right predictions over the total number of cases rated the formula to calculate from equation (2.19).

$$acc = \frac{Total\ correct\ predictions\ for\ images}{total\ of\ images} * 100\% \qquad (2.19)$$

If the dataset is an unequal sample (unbalanced), it is not preferable to use the metric of accuracy in evaluating the performance of the model.

In the case of dataset is unbalanced, the accuracy value of the model is high, but the performance of the model may not be good if it is exposed to real samples. Therefore; it is better to use other metrics such as Precision or Recall [97, 98]

2.9.2 Precision (P)

Precision is used to measure the positive patterns that are correctly predicted from the total predicted patterns in a positive class'. The formula is to calculate of equation (2.20).

$$P= \frac{tp}{tp+fp} \qquad (2.20)$$

Where tp is true positive, fp is false positive.

2.9.3 Recall (R)

The number of true positive samples over the sum of the true positive samples and the false negative samples. We can calculate by using equation (2.21).

$$R = \eta_i = \frac{tp}{tp+f_N} \qquad (2.21)$$

Where fn is false negative, η_i is represent recall for each class.

The average recall (Ƞ) calculated using equation (2.22) [2].

$$Ƞ=\frac{1}{m}\sum_{i=1}^{m}\eta_i \qquad (2.22)$$

Where m is the number of classes.

Most models depend on probability in predicting, and most of the time we use a threshold of 0,5, but this threshold is not always better because the recall value and precision change when the threshold value is changed. Figure (2.24) shows the values of recalls and precisions [97].

2.9.4 F1 Score

It is the metric that combines both recall and precision. Calculate the F1 score using Equation (2. 23). where R is recall; P is precision [97] .

$$F1 = \frac{2PR}{P+R} \qquad (2.23)$$

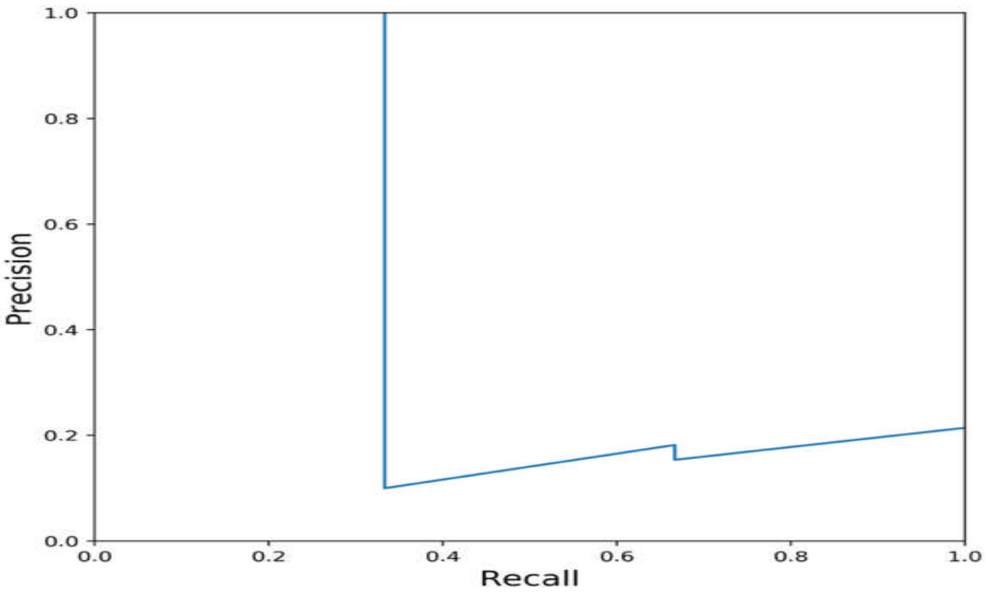


Figure (2.24): precision-recall curve [97].

2.9.5 Confusion Matrix

The confusion matrix is a table containing the results of the classified categories. The sum of the matrix diameter represents correctly classified samples. The rest of the samples are incorrectly classified. The confusion matrix contains information about the expected and actual classification. The confusion matrix enables calculating metrics such as precision, accuracy and recall to evaluate the model's performance [98]. Each category in the confusion matrix contains four types; true positive, true negative, false positive and false negative. Table (2.2) shows the distribution of sample types on the confusion matrix with respect to class A.

Table (2.2): Types of samples for class A

Class A	True positive	False negative
Class B	False positive	True negative

2.10 Python Language and Online Server

It is a high-level programming language that is simple, easy to learn and used for general purposes. Python was invented by Guido van Rossum and the first version of the language was in 1991. Most of the commands of Python are similar to the language spoken by humans, so it will be easy to learn. The development of the Python language continued and many features were added in each new version, until it became the most important programming language in recent years [99].

Python is an essential language for data analysis and machine learning. Python works on many operating systems such as Unix, Linux and Windows.

The Python language provides the ability to transfer the program written in it from one platform to another. Python is open source and there are no restrictions on use and distribution, although the Python Institute retains the rights to the program. Python is extensible. It is possible to use with Python code parts of other languages such as C/C++. Python is characterized as a high-level interpreted language because the code automatically converts the input code to the language of the computer [100].

Online server is an online community for data science researchers and deep learning practitioners. The site contains five main sections: data sets, codes, discussions, competitions and courses. The first section provides a different and varied data set, and there is an evaluation for each data set with the goal of creating this data. The second section provides many codes that have been implemented by beginners and professionals, and it is possible to copy and add any code and modify it. When the code is run, the site's GPU can be used for the purpose of processing and showing the results. The site allows each user to run the site's GPU for 20 hours a week. The third section provides discussions and you can ask about a specific topic and most of the time the answer comes. The fourth section contains competitions that provide financial prizes to the winning

contestant. The last section contains many learning courses such as Python, deep learning and natural language processing [101].

2.11 Embedded System

It is an electronic system that contains a programmable chip and is considered a computer system with specific tasks. Embedded devices are combined with peripheral devices to perform specialized functions to provide solutions for specific projects. The embedded system is often part of a larger system such as automatic brake control in cars. Computers also contain embedded systems such as keyboard and mouse. The use of embedded devices is constantly increasing and there are many programming languages such as C, Python and C++ that are used to program embedded devices. Embedded devices contain a program and a processor to perform the required function and often perform a single task. It contains a reading memory to embed the program and peripheral devices to connect the inputs and outputs. It depends on the microprocessor or the microcomputer. Figure (2.25) shows general example of an embedded system. [102-104].

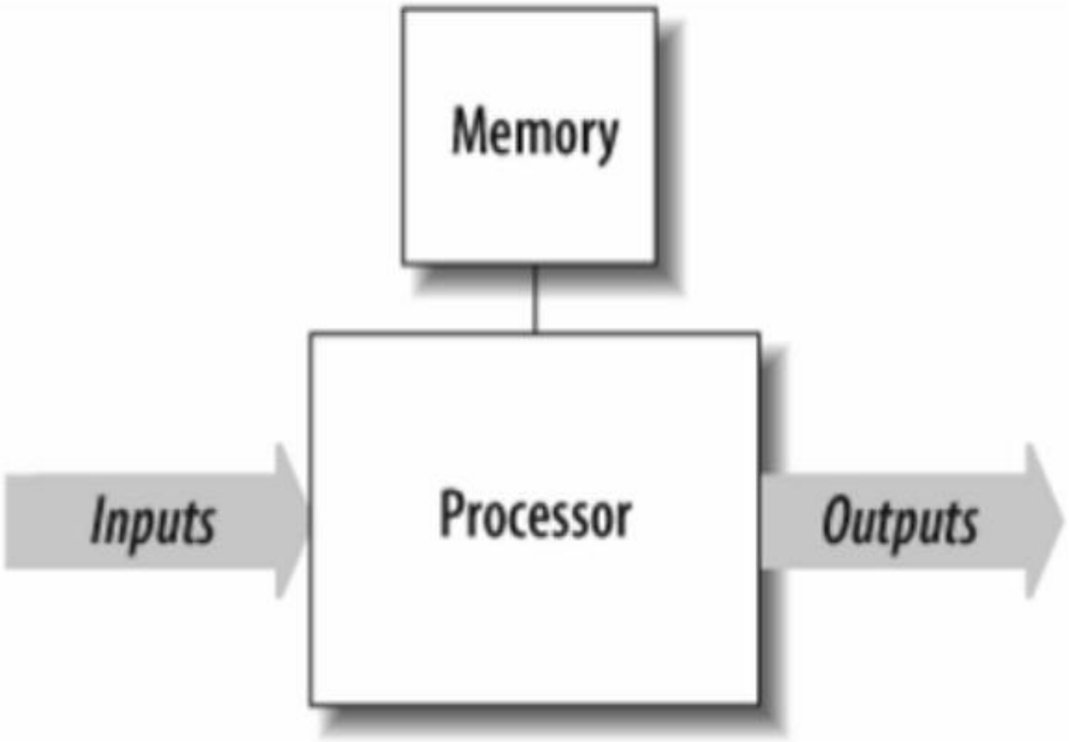


Figure (2.25): A generic embedded system [104].

Embedded devices are characterized by their low cost, small size, low energy consumption and strict operation. Their functions are limited to a larger electrical or mechanical system. Supervising the application programs by the real operating system, and the system contains software and Embedded applications Embedded device manufacturers provide a board that contains all electronic Embedded. The electrical connections between the components of the board are made through a copper path, and the boards are classified into five classes: Memory, central processing unit (CPU), Input device, Data pathway(s)/bus(es) connecting other components to quickly transfer data between the components of the board and output device(s). Figure (2.26) represents the five categories based on the Van Neumann model, this model is a tool used to know and understand the structure of any electronic device, including knowledge of embedded systems devices [105, 106].

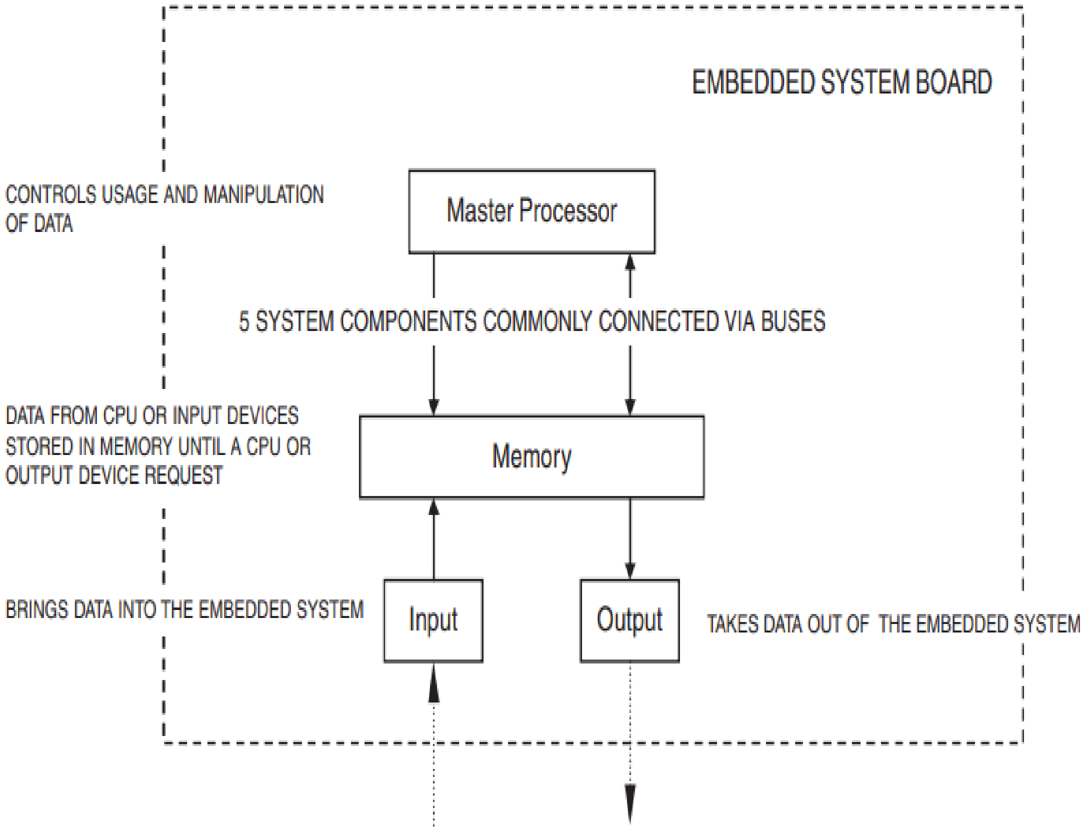

EMBEDDED SYSTEM BOARD
CONTROLS USAGE AND MANIPULATION OF DATA
Master Processor
5 SYSTEM COMPONENTS COMMONLY CONNECTED VIA BUSES
DATA FROM CPU OR INPUT DEVICES STORED IN MEMORY UNTIL A CPU OR OUTPUT DEVICE REQUEST
Memory
BRINGS DATA INTO THE EMBEDDED SYSTEM
Input
Output
TAKES DATA OUT OF THE EMBEDDED SYSTEM

Figure (2.26): Embedded system board organization [106].

Embedded System Applications [106-108].

- Mobile Phones
- Mobile and fixed internet devices
- In laboratories and factories in robotics control circuits
- In the medical and security fields
- Self-driving cars

Table (2.3) shows the difference between computer and embedded systems. Through the table, the most important characteristics and components of embedded systems will be identified [104-106].

Table (2.3): Comparison of computer and embedded systems

Computer	Embedded System
It consists of two parts: Embedded and software.	It consists of three parts: Embedded, firmware and software.
It can perform many tasks.	Performs limited tasks.
Computers can be reprogrammed for a new purpose	Included Embedded is made for a specific set of purposes only.
It requires high operational power, larger size and higher cost	Requires little operational power, smaller size and lower cost
It is not timed	Embedded devices are time-limited.
Requires large memory size	Requires smaller memory size

Embedded devices are interactive, i.e, they interact through the surrounding environment, such as Arduino and Raspberry pi. The Arduino is an open-source single board that uses 8 bits or 32 bits that interact with the environment through boards equipped with sensors. Shields provide users to connect external connections to the Arduino board, The use of Arduino provides many functions such as wireless communication and motor control [102].

2.12 Raspberry Pi

It is a computer belonging to the single-board computers (SBCs) family, it is as small as a credit card. Raspberry Pi was developed in the UK for the purpose of learning the basics of computer science in schools. Raspberry Pi is characterized by its high speed compared to Arduino (see appendix A), up to 40 times, it has a very large random-access memory (RAM) compared to the Arduino. Raspberry Pi is a standalone computer that works on several systems such as Windows and Linux. performs multi tasks that supports a USB port and wireless connection [102]. The manufacturer provides many Raspberry Pi, many versions, and how much the company provides two models, (A, B) in each version. Figure (2.27) shows a copy of the Raspberry Pi 4 model B and the contents on the board [109].

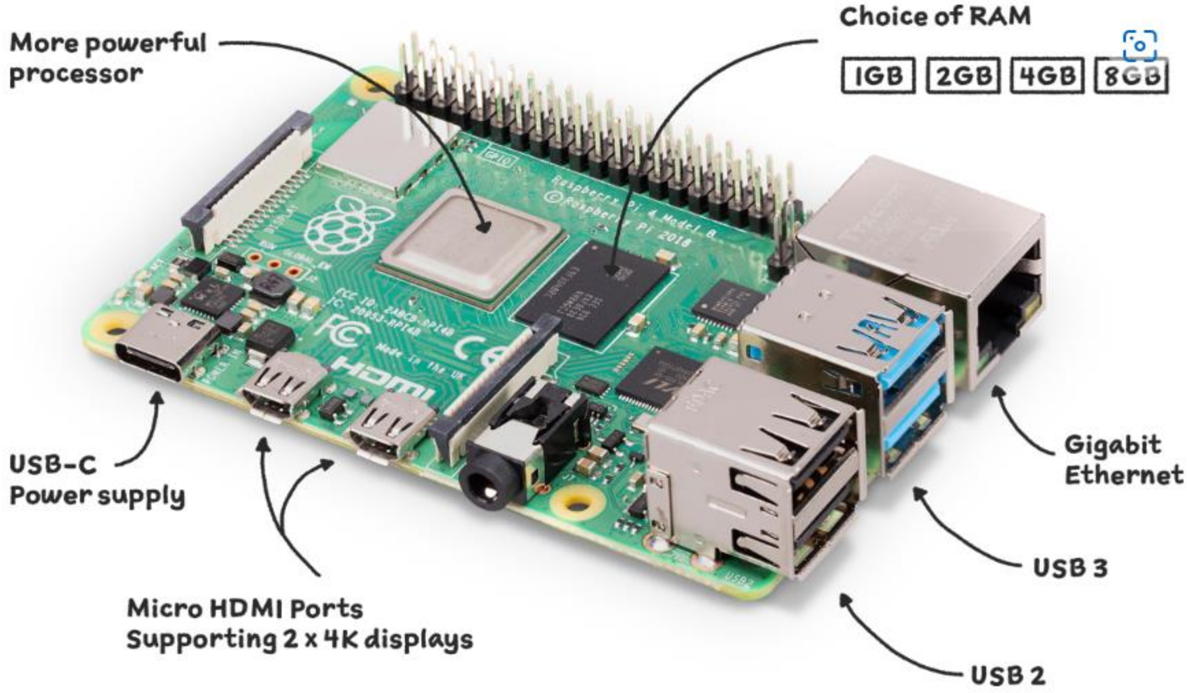


Figure (2.27): Raspberry Pi 4 model B [109].

The Raspberry Pi 4 model B, the latest version of the raspberry foundation supports dual 4K display with new CPU and GPU with (1, 2, 4, 8) GB RAM.

Raspberry Pi4 has a quad-core processor and contains video core 6 (vc6) GPU is used for 2D and 3D graphics processing, as well as image processing. The following points explain the parts and specifications of the Raspberry Pi 4 Model B [109].

- SoC: Broadcom BCM2711B0 quad-core A72 (ARMv8-A) 64-bit 1.5GHz
- GPU: Broadcom Video Core VI
- Networking: 2.4 GHz and 5 GHz 802.11b/g/n/ac wireless LAN
- RAM: 1GB, 2GB, or 4GB LPDDR4 SDRAM

- Bluetooth: Bluetooth 5.0, Bluetooth Low Energy (BLE)
- GPIO: 40-pin GPIO header, populated
- Storage: microSD
- Ports: 2 × micro-HDMI 2.0, 3.5 mm analogue audio-video jack, 2 × USB 2.0, 2 × USB 3.0, Gigabit Ethernet, Camera Serial Interface (CSI), Display Serial Interface (DSI)

CHAPTER THREE

Proposed TOD System

CHAPTER THREE

Proposed TOD System

3.1 Introduction

This chapter explains the proposed TOD system for classifying X-ray images to classification and detect threat objects. The objective of using the proposed TOD system raises the level of security and safety at airports and border checkpoints. The proposed system is to classify and detect X-ray images to know what is threatened and what type. Figure (3.1) Shows the TOD proposed system implemented in Python using online servers.

3.2 Proposed TOD System

There are several TOD systems suggested and implement to improve security level speedup the detection mission. However, more accuracy real time detection evaluation is required.

1. In this work, a new architecture of TOD system is proposed to improve accuracy.
2. Evaluate system using embedded devise implantation.

Building this proposed system has been carried out over two phases. The first phase was simulation and building a deep learning network while the next phase was implementing the trained network using raspberry Pi embedded device. The two phases are detailed below.

3.3 Simulation, Building and Training Deep Learning Network

The available dataset needed to train the proposed system is scarce because it is related to the security aspect. GDXray dataset is publicly available so it was used in our chosen convolutional neural network.

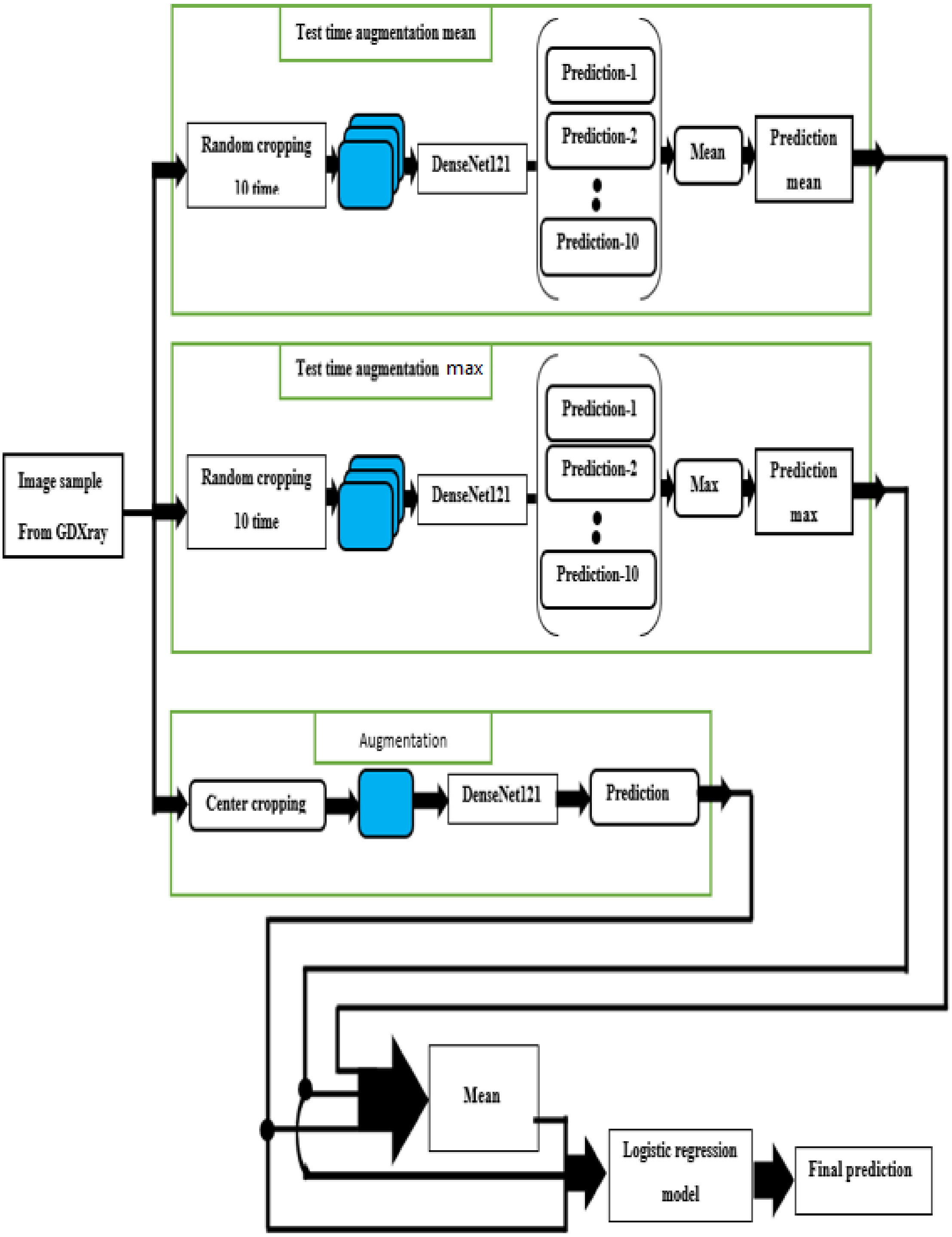


Figure (3.1): Blockdiagram of the proposed TOD system

The proposed TOD system consists of a DenseNet121network and a logistic regression algorithm to get the final prediction. Using techniques such as augmentation and test time augmentation on the dataset improves the performance of the proposed TOD system.

3.3.1 DenseNet121

After loading and organizing the GDXray dataset in table (4.1), the DenseNet convolutional neural network algorithms are loaded and this algorithm is pre-trained on the ImageNet dataset (transfer learning). The importance of using transfer learning comes for the following reasons:

1. Reduces the time to train the algorithm on the required data set.
2. It saves us the effort of entering large data into the network.
3. Algorithm performance improvement.

Network layers are unfrozen. The last layer of the network contains four classes instead of the 1,000 classes of the ImageNet data set. The dataset is passed through all blocks of the DenseNet121 network as detailed in Section (2.7.4) for network training and performance evaluation. After passing the data set in the form of patch, each patch contains 32 images, the network performance was not at the required level. Therefore; the error function is high between the training dataset and the test dataset (high overfitting).

3.3.2 Augmentation Process

The application of data augmentation is considered one of the most important applications to reduce the overfitting in convolutional neural networks, and it is the first part of the proposed system. Figure (3.2) shows the augmentation process. The importance of applying the data augmentation comes from the need of deep neural networks for big dataset during the training phase. This technology deals with image data to augment it, by flipping the images=1, vertical flipping of images=1, taking the zoom=1and max zoom=2 of the images, the maximum rotation the lighting=75% and image warping 20%. figure (3.3) shows the effect of the augmentation on the images. After applying the data augmentation to the GDXray dataset to train the DenseNet121 network, we extract the map of features of the images and get the prediction for the four classes by the softmax activation function in the last layer.

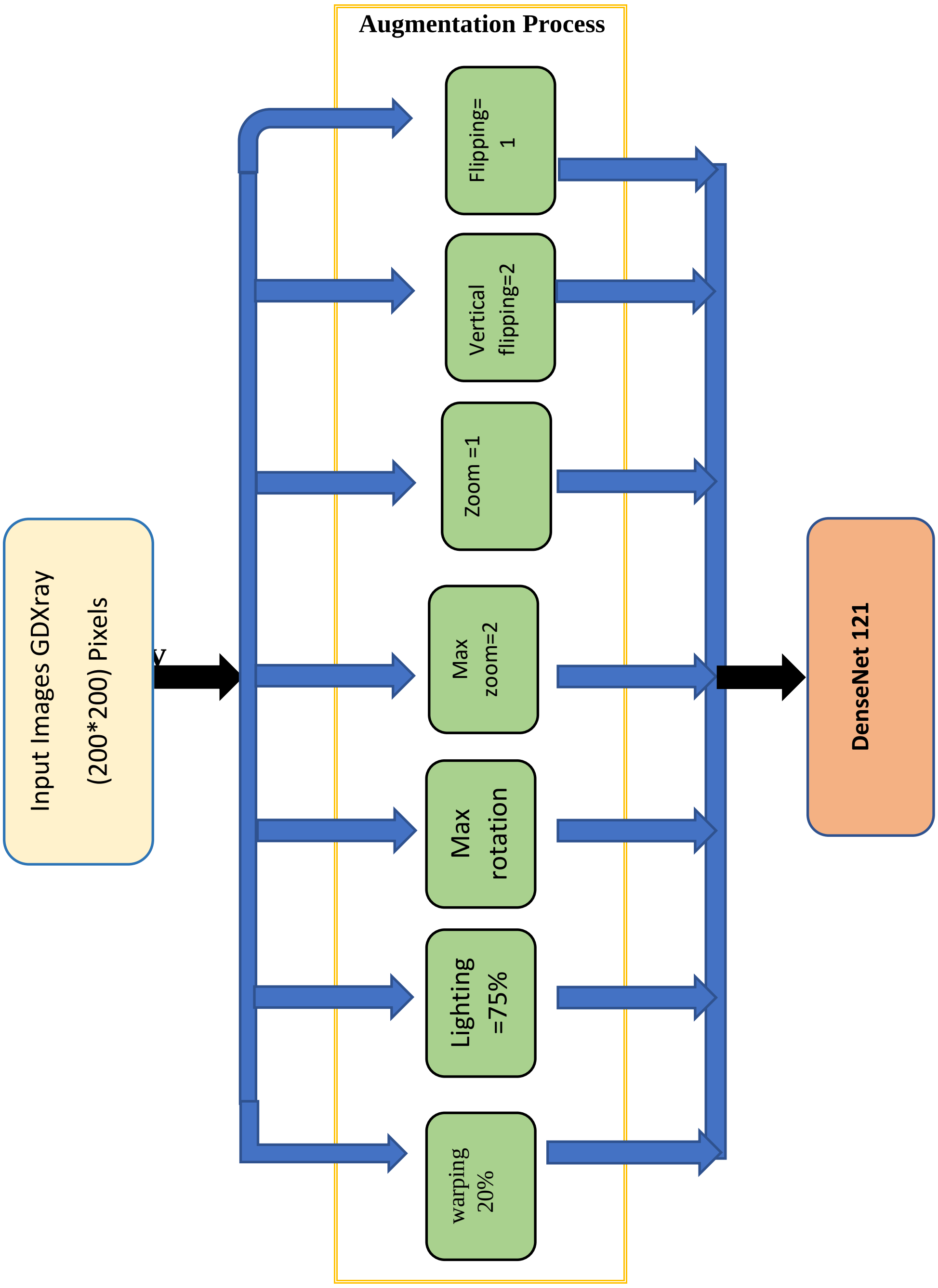


Figure (3.2): Blockdiagram of the Augmentation Process

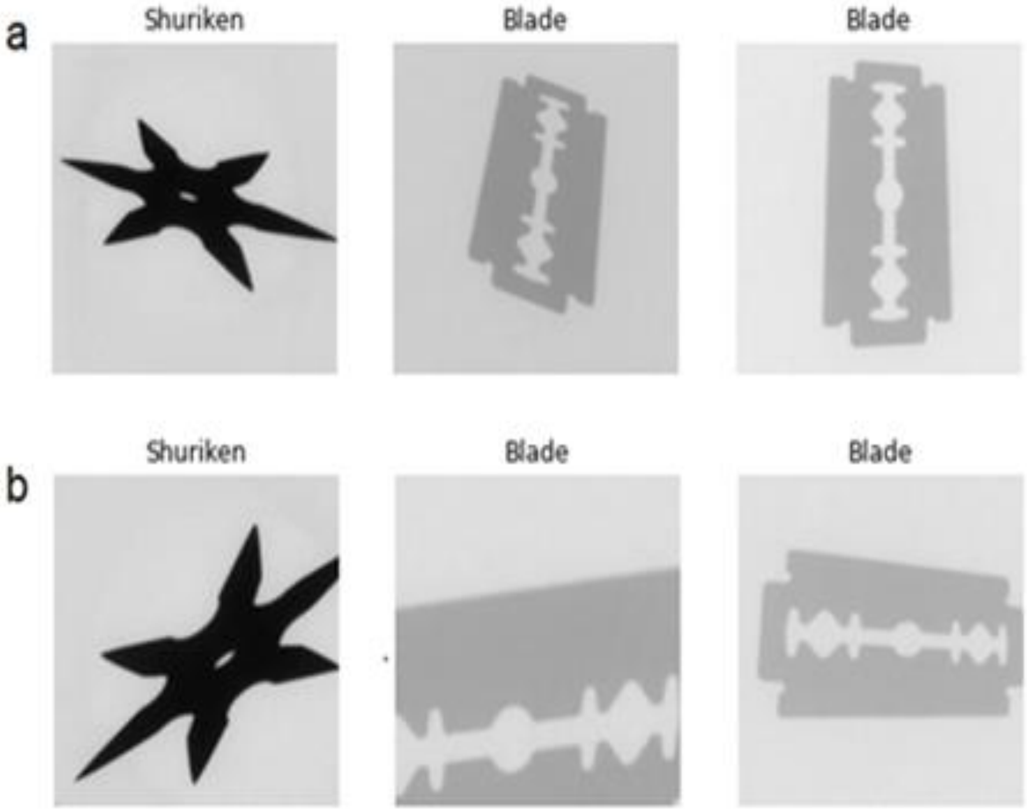


Figure (3.3): a) samples of dataset. b) Effect augmentation on samples of dataset

3.3.2 Test Time Augmentation (TTA)

The technique of increasing the test time achieved a good performance to improve the models of convolutional neural networks during the testing phase. This technique crosses the second part of the proposed system, as shown in figure (3.4). The working principle of test time augmentation is to select random parts from the original images and pass them over the network during the testing phase to obtain a prediction. One of the benefits of using this technique is to augment the data set during testing and many image variables are extracted. Two methods of test time augmentation were applied in the proposed system, namely predicted of TTA mean (number of epochs n=10) and TTA predicted maximum (number of epochs n=10). The prediction is obtained using the softmax function in the last layer.

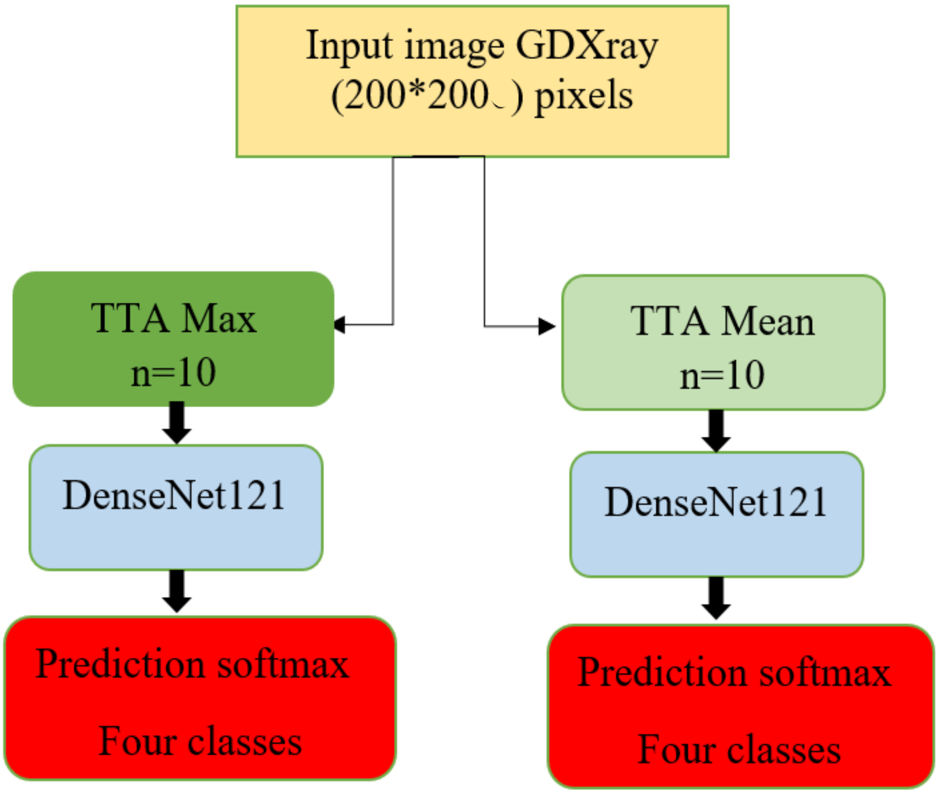


Figure (3.4): Blockdiagram of the TTA mean and TTA max

3.3.3 Logistic Regression (LR)

The logistic regression algorithm is simple and widely used in the classification task and is based on probability. The logistic regression algorithm is the last part of the proposed TOD system. Figure (3.5) shows all the stages of the proposed TOD system. The TOD proposed system is to make the logistic regression have three inputs as follows:

1. An additional feature that uses the mean for predictions of data augmentation and TTA predictions.
2. Data augmentation prediction.
3. Prediction of TTA max.

Then the logistic regression output is the final prediction of the data set and the softmax function is the final output of the proposed system for classifying the four classes. The proposed system achieved a high performance for the task of classifying the X-ray image data set in table (4.1).

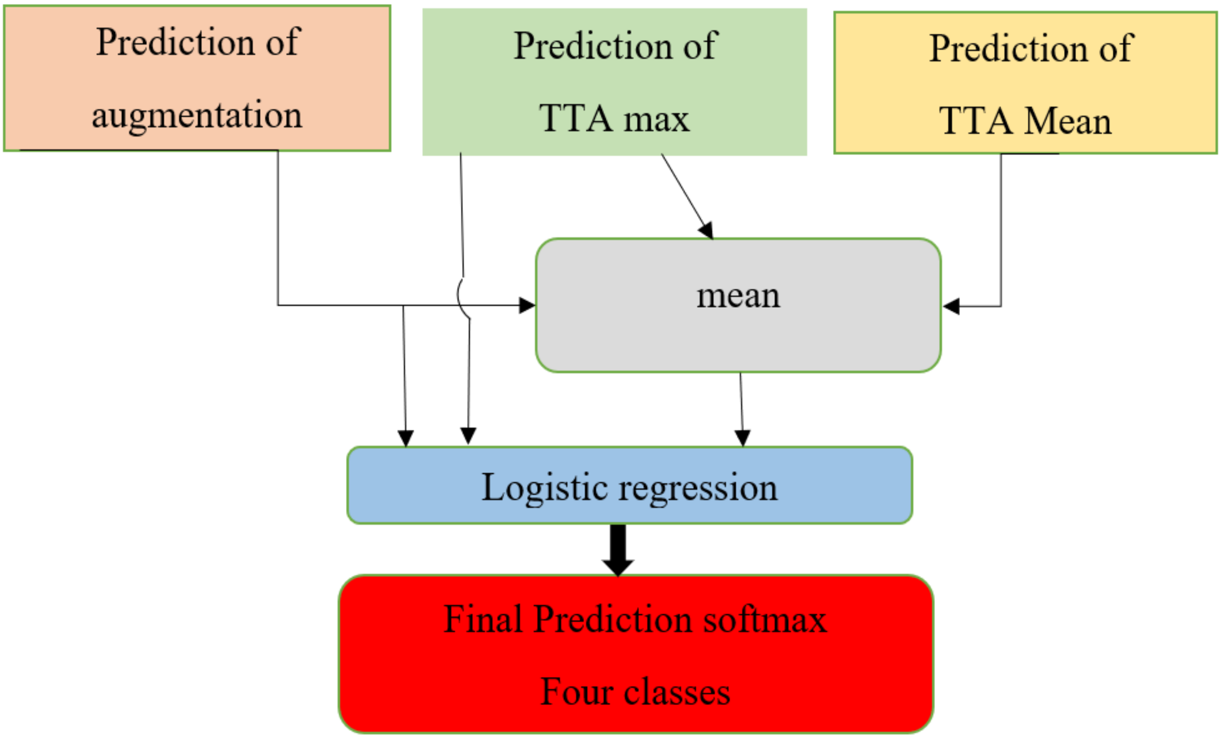


Figure (3.5): Blockdiagram of proposed TOD system

3.4 Embedded Implementation Using Raspberry Pi

Proposed TOD system will be implemented Embedded system, as shown in figure (3.6). To implement TOD system, a Raspberry Pi 4 model B board was used. This board supports the camera, Internet connection, USB ports to use the mouse and keyboard, secure digital card (SD-card) and HDMI ports to use the display screen. we take the following steps:

1. Download Raspberry Pi 4 Model B Operating System.
2. Download the Python language with the required libraries to run the code, download the proposed pre-trained system, and write a code to run the test.

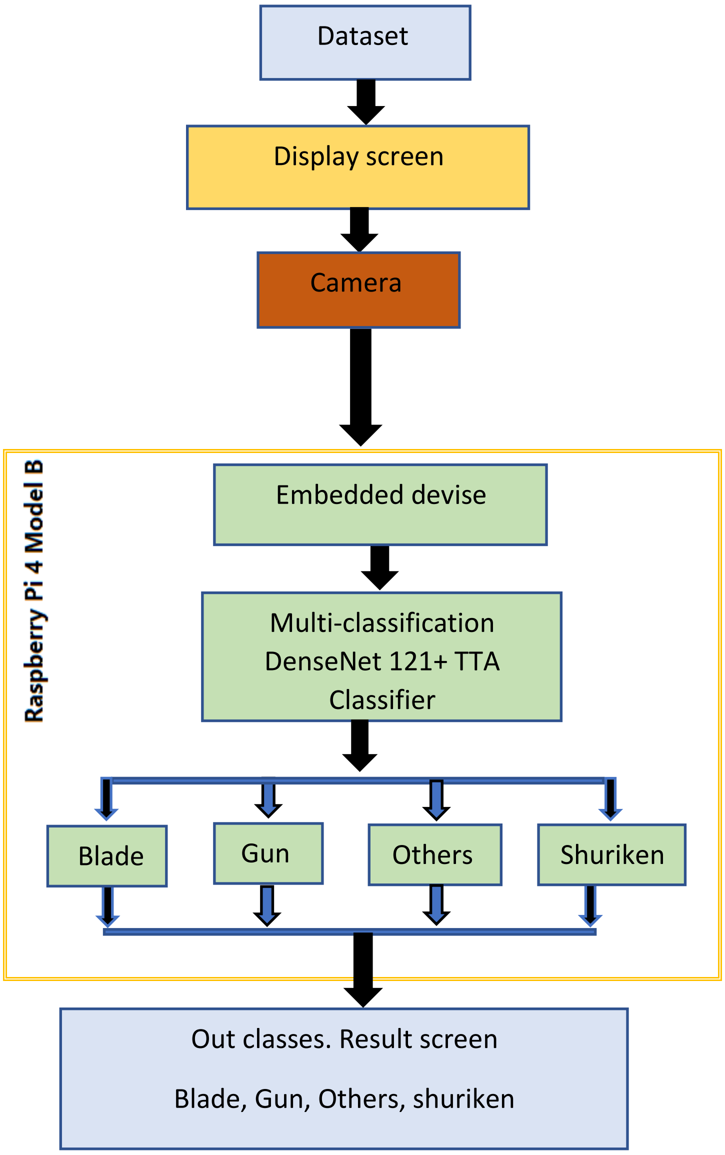

Dataset
Display screen
Camera
Raspberry Pi 4 Model B
Embedded devise
Multi-classification DenseNet 121+ TTA Classifier
Blade
Gun
Others
Shuriken
Out classes. Result screen
Blade, Gun, Others, shuriken

Figure (3.6): Blockdiagram Embedded system implementation

3.4.1 Downloading Operating System (OS)

Provide an SD card and insert it into the PC to download the operating system. After logging in to the Raspberry Imager website. Choose the 64-bit operating system of the Raspberry pi 4 and download it to an SD card. After completing the download process on the SD card, is inserted the chip into the raspberry. The display screen, keyboard, camera and mouse are connected with the Raspberry pi 4 model B and the power supply.

3.4.2 Python and the Libraries Installation and View

The operating system will work on the display screen. We activate the Internet service for the Raspberry pi 4 -B. Entering the comments icon, a black screen will open in which you can write instructions for downloading Python and the required libraries.

After that, we download the proposed trained system and write the code. Display of test images appears on the computer screen after capturing by the camera The category prediction appears on the screen connected with Raspberry pi 4-B. Figure (3.7) shows a Raspberry Pi 4 - B with all the main components of the proposed system.

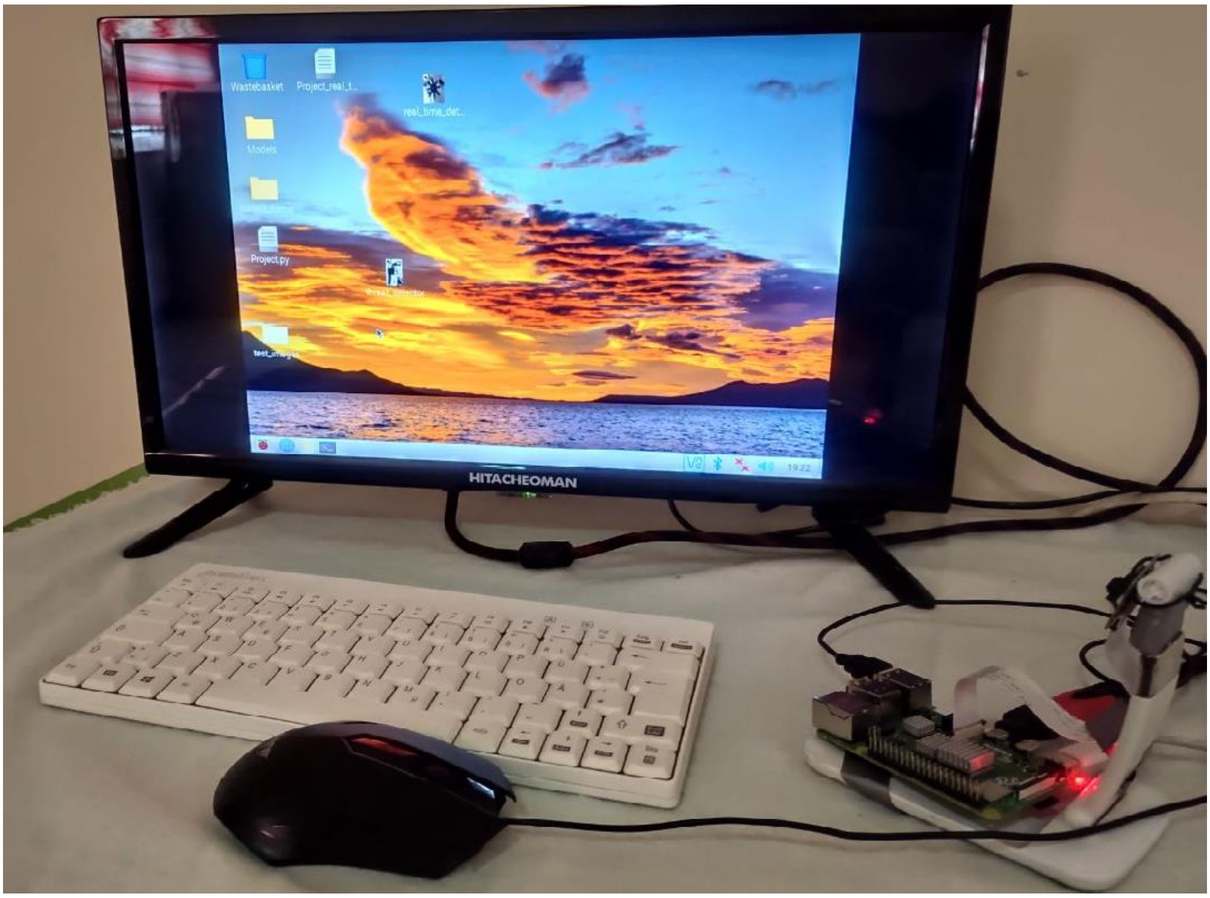


Figure (3.7): Main components of proposed TOD system

Chapter Four

Results and Discussion

CHAPTER FOUR
Results and Discussion

4.1 Introduction

This chapter presents the results and discussion of the TOD proposed system described in chapter Three. The results are presented in two phases, simulation phases and Embedded phases. The results will be displayed each phase in two parts: the first part is the DenseNet121 network and the second part is the entire TOD proposed system. The prediction results for the image classification task to detect threat elements are displayed for each phase. Based on this dataset, the work is classified into categories according to the following objects: gun, blade, shuriken, and others, as shown in table (4.1). It contains 1,950 images (divided into three sets of images: 900 images of training, 350 images of validation and 700 images of testing [2]. Before starting the training process for images and knowing the strength of the proposed system's performance, the data must be organized, re-sized to become (200 * 200) pixels because the sizes of the original images are different in size.

Table (4.1): GDXray dataset [2] used in our experiments

Set	Gun	Shuriken	Blade	Other
Training Series Images	B0049 1-200	B0050 1-100	B0051 1-100	B0078 1-500
Validation Series Images	B0079 1-50	B0080 1-50	B0081 1-50	B0082 1-200
Testing Series Images	B0079 51-150	B0080 51-150	B0081 51-150	B0082 201-600

4.2 Classification Results of (TOD) Simulation Phase

In this section the code results of the TOD system for detecting threat elements by classifying X-ray images will be presented. The first part presents the results of classifying the images in table (4. 1) using the DenseNet121 network and applying data augmentation. The second part presents the results of the entire proposed system, as detail in section (3.3). The validation data set is used to know the performance of the DenseNet121 and the proposed system. Using recall metric to evaluate the performance of the models because the data set is unbalanced, as detail in section (2.9.1).

4.2.1 Simulation Results of DL Network (DenseNet121)

After building the DenseNet121 network, it was trained on 900 images from the training data set in Table (4.1). The model did not perform well in the classification task, the recall metric was low, because the training data set is not sufficient. When evaluating the model with 350 validated images due to the occurrence of overfitting. Using data augmentation technique to reduce the overfitting, which leads to improving the performance of the model. Table (4.2) shows the calculation of the values of the loss function (cross entropy) recall metric. Use ten attempts (epochs), we get the best performance for the model at the seventh epoch, as shown in the yellow color in table (4.2). Through the validation phase, we obtained the best model of the DenseNet121 network in the seventh epoch with values; cross entropy = 0.225916, Recall = 92.75%.

Figure (4.1) shows the high loss function (cross entropy) between the training curve and validation curve. Figure (4.2) shows the improvement of the results, reducing the overfitting after applying the data augmentation to obtain the value of a lower cross entropy between the training curve and the validation curve. The best result obtained is in the seventh epoch. Therefore, it was approved and stored. The execution time for each epoch takes 23 seconds.

Table (4.2): Validation results

Epoch #	Train-loss	Valid-loss	Recall-score	Time in second/ epoch
1	1.260951	1.269505	0.672500	23
2	0.698972	0.683495	0.838750	22
3	0.471671	0.526461	0.818750	23
4	0.334006	0.497417	0.802500	23
5	0.286001	0.282675	0.773750	22
6	0.254676	0.562067	0.722500	23
7	**0.229521**	**0.225916**	**0.927500**	**23**
8	0.214116	0.805848	0.546250	23
9	0.205229	0.603795	0.662500	23
10	0.211667	0.652519	0.631250	23

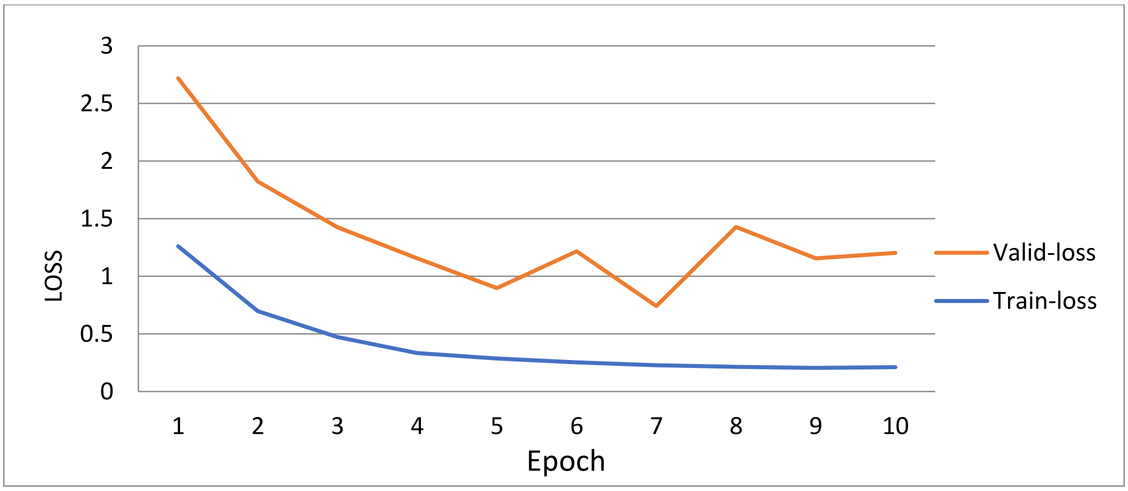


Figure (4.1): Training and validation loss curve without augmentation

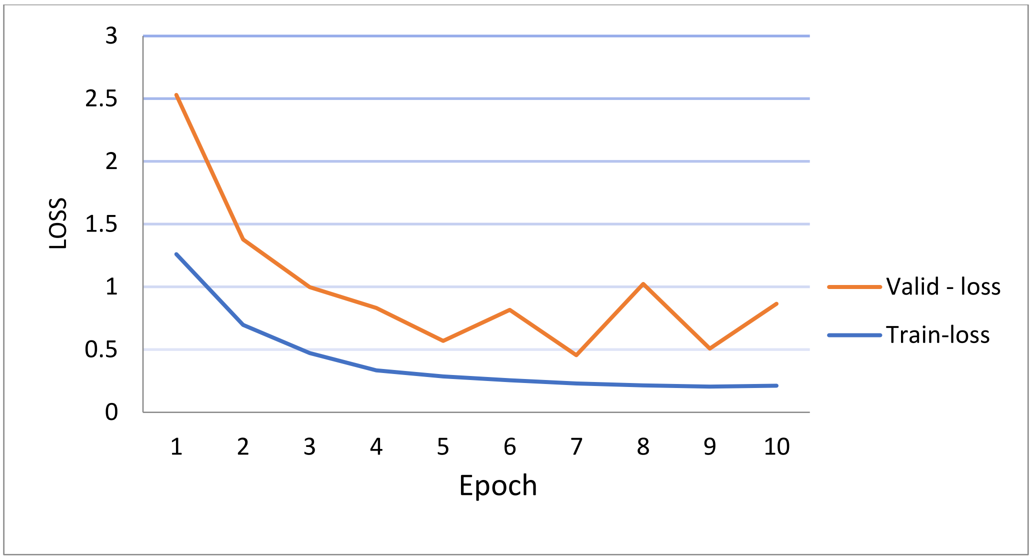


Figure (4.2): Training and validation loss curve with augmentation

The final results for the recall of classifying the test images consisting of 700 images divided into four categories (100 images of the gun, 100 images of the shuriken, 100 images of the blade and 400 images of the others) are Figure (4.3) shows the results by displaying the confusion matrix, recall is 94.1%.

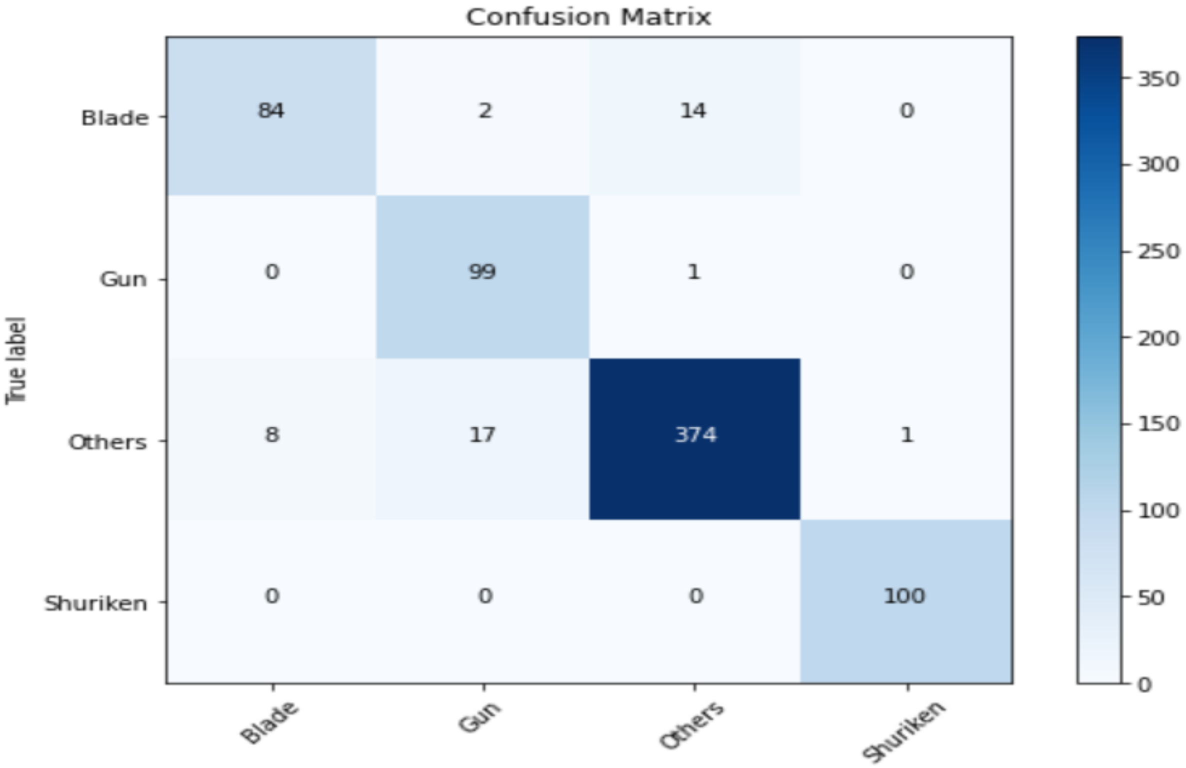


Figure (4.3): Confusion matrix of DenseNet121

4.2.2 Simulation Results of the Proposed TOD System

The results of the test images of the DenseNet121 network were not at the level of ambition, so test time augmentation and logistic regression were used to obtain the final prediction for image classification. After using the techniques as explained in section (3.3). of the TOD system. The highest possible results were obtained for the recall of classification of four categories.

4.2.2.1 Test Time Augmentation Results TTA

The TTA was applied on DenseNet121 network. The value (N) represents the number of times the data set is passed over the TTA technique. After changing the value of N more than once to get the highest recall, we chose the best value at N = 10. Table (4.3) shows several values of N and the highest Validation recall value is 95.37%.

Table (4.3): Validation recall with TTA

N	Validation recall
3	93.62%
6	94.75%
10	**95.37%**
12	95.24%

4.2.22 Logistic Regression Results

After obtaining the best possible results by applying a test time augmentation, the logistic regression algorithm was used to obtain the final prediction for a classification task, as explained in section (3.3.5). Some parameters are used to improve the performance of logistic regression, such as L1ratio and Coefficient C as shown in the section (2.4.2). The coefficients (C and L1ratio) of logistic regression were changed. The best result of L1 ratio = 0.9 according to table (4.4). The best value for C = 0.0106 according to table (4.5). After selecting the best values for the coefficient of L1 ratio and coefficient of C, the highest recall values were achieved 95.3% and 95.37% respectively.

Table (4.4): L1 ratio results with TTA

L1 ratio	**Validation recall**
0.2	94.12%
0.4	94.75%
0.6	94.6%
0.9	95.3%
1	91.75%

Table (4.5): C coefficient with TTA

C coefficient	0.001	0.0042	0.0074	0.0106	0.0138	0.0171
Recall validation %	25	25	95.2	95.37	94.87	94.87

After showing the results of verifying the application of the test time augmentation and using the logistic regression algorithm, the highest possible recall of the TOD proposed system. Figure (4.4) shows the confusion matrix, which shows the final results for recall. The recall of each category was calculated using equation (2.17) and the final average for the four categories was calculated using equation (2.18). A final recall of the TOD proposed system has been obtained (recall =97.68%).

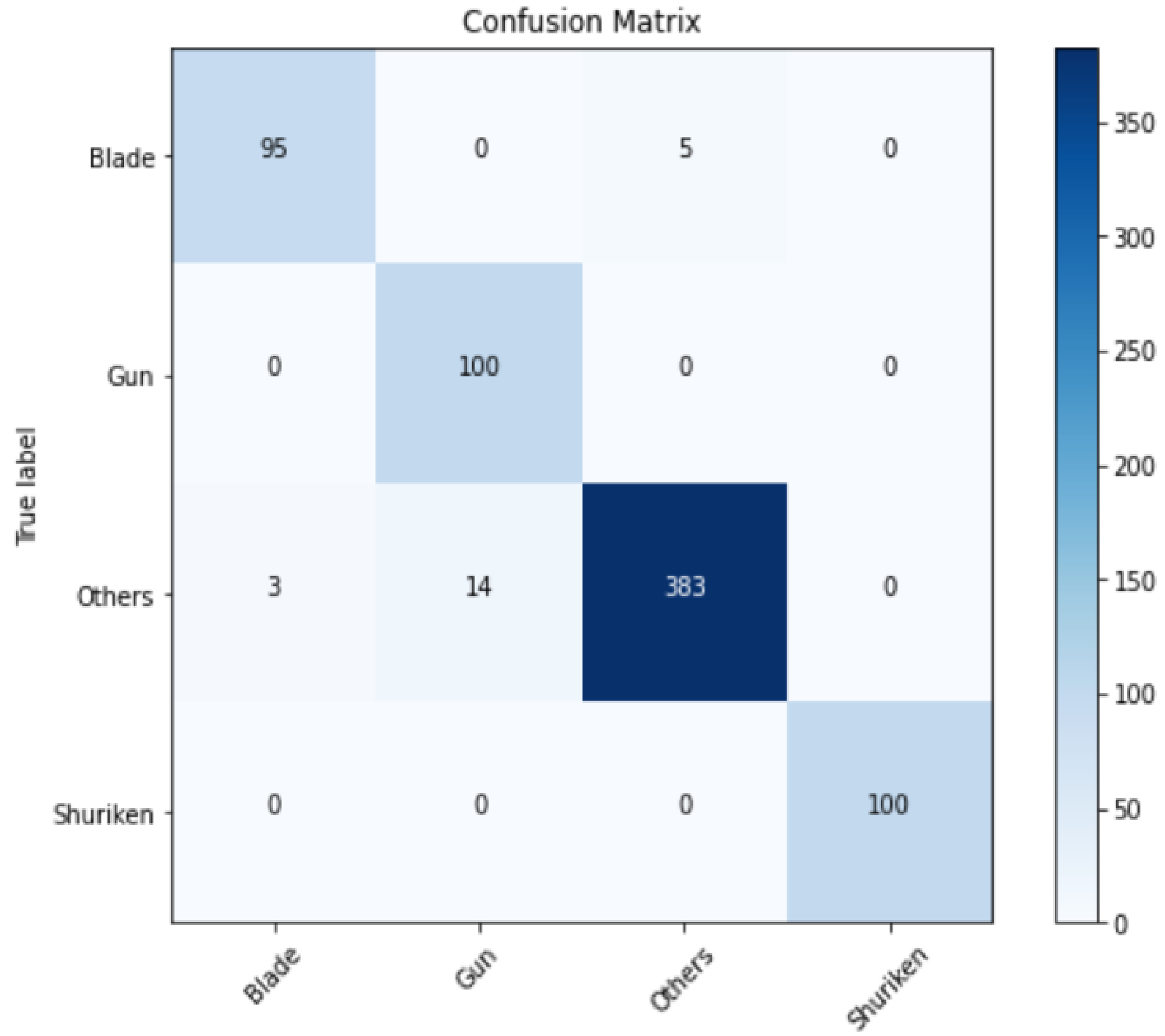


Figure (4.4): Confusion matrix of TOD proposed system

Table (4.6) shows some of the metrics that were used to determine the strength of the TOD proposed system's performance in classifying images to detect threat elements as detail in section (2.9).

Table (4.6): Metrics for TOD proposed system.

TOD proposed system	Rcall %	Precision %	F1 score %
results	97.68	95.9	96.78

4.3 Classification Result Of (TOD) Embedded Implementation Phase

In this section, the results of classification recall of images through the implementation of the DenseNet121 network and the proposed system on the Embedded are presented.

4.3.1 Results of Embedded Implementation of DenseNet121

After loading the trained DenseNet121 network on the training data set in table (4.1) on Raspberry pi 4 model B. Capture test image using a raspberry camera. Reducing noise such as reflections and stabilized the camera in front of the test images in the best possible way (control the position of the camera and the lighting of the surrounding environment. Paying attention to the way test images are displayed on the computer. The best possible recall was obtained, which reached a value of 91.812%. Figure (4.5) shows the classification results for the four image categories through the confusion matrix.

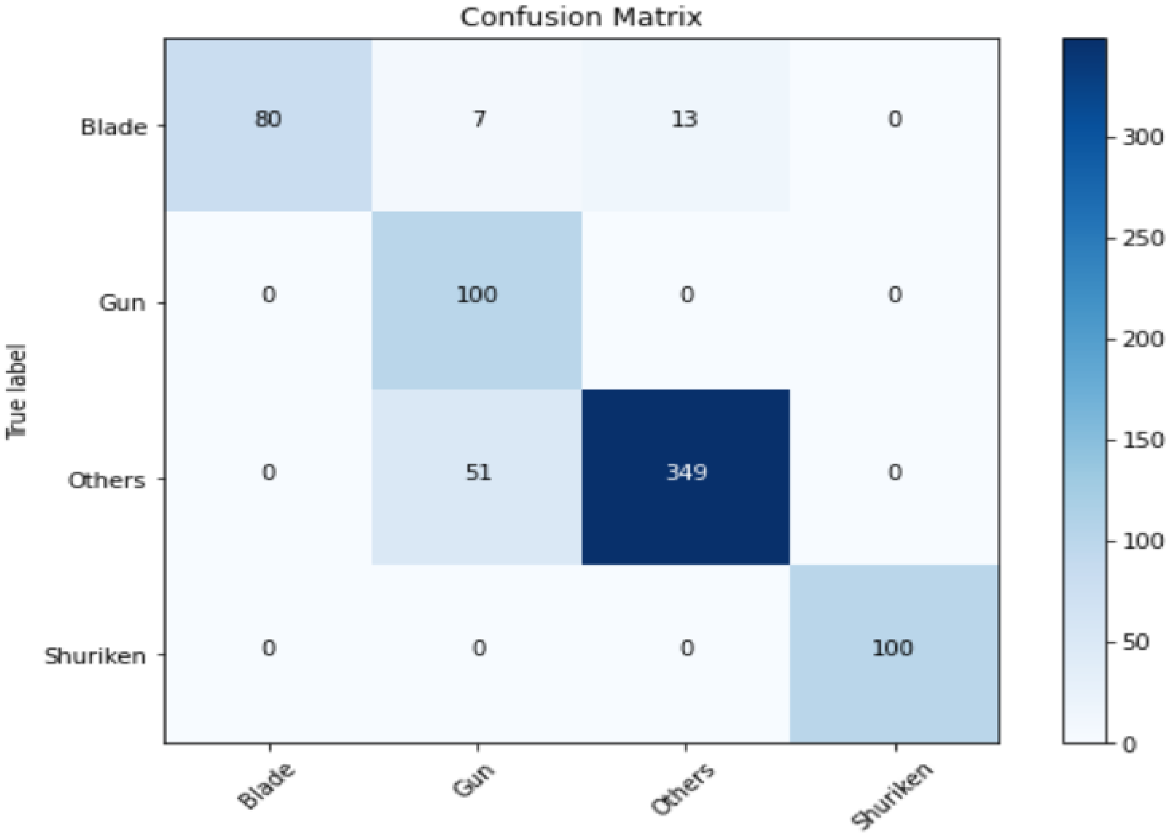


Figure (4.5): Confusion matrix of Embedded DenseNet121

4.3.2 Results of Embedded implantation Of Proposed TOD System

The proposed system has been implemented on a Embedded platform (Raspberry Pi 4). The Raspberry Pi 4- B display shows the results of the classification task for the TOD proposed system. A recall of 94.25% was obtained. Figure (4.6) shows the final prediction of image classification on Raspberry using confusion matrix.

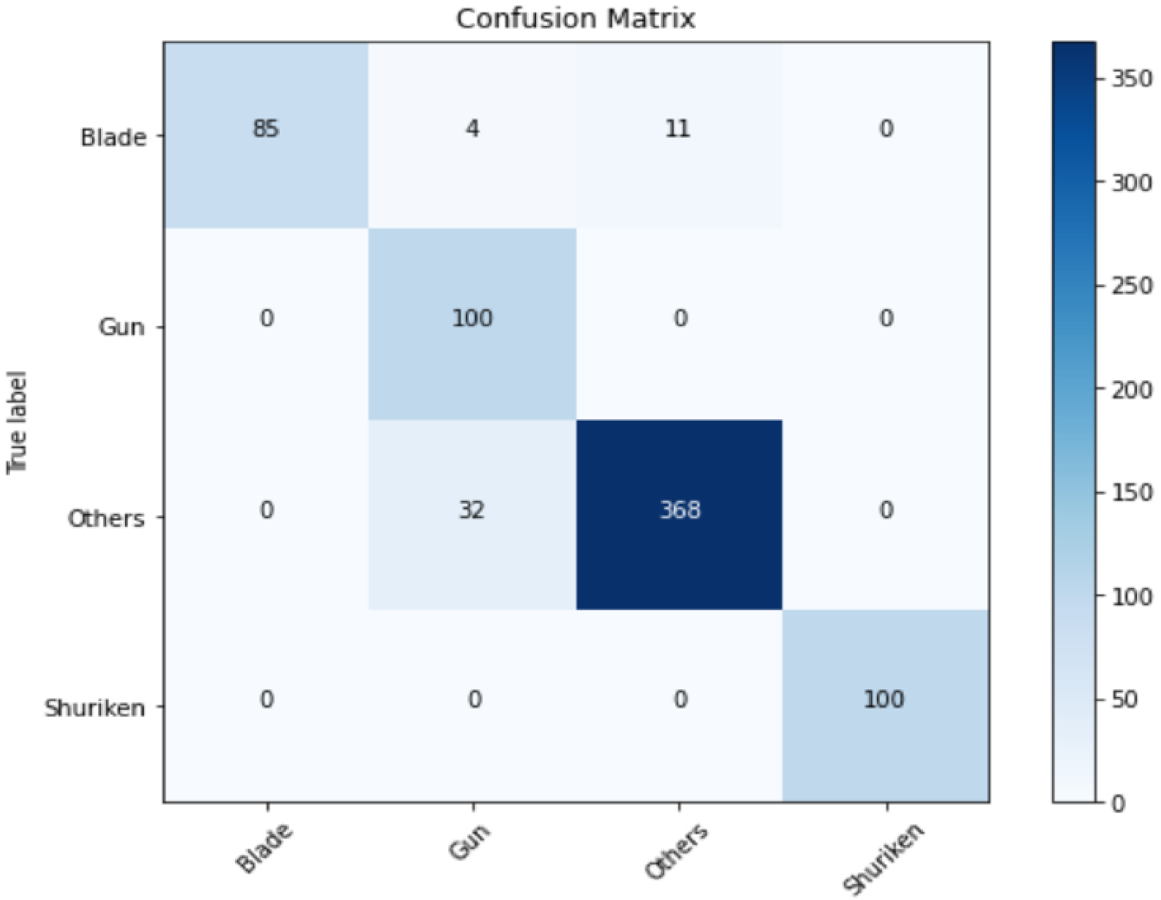
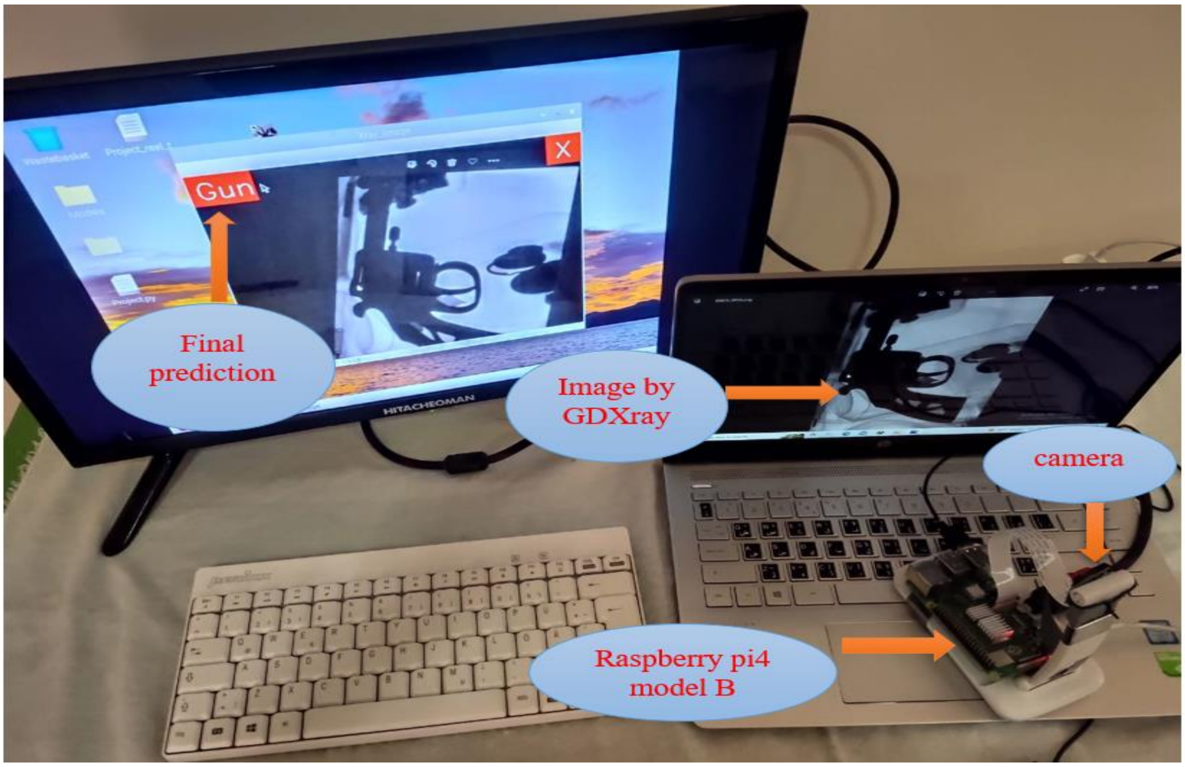


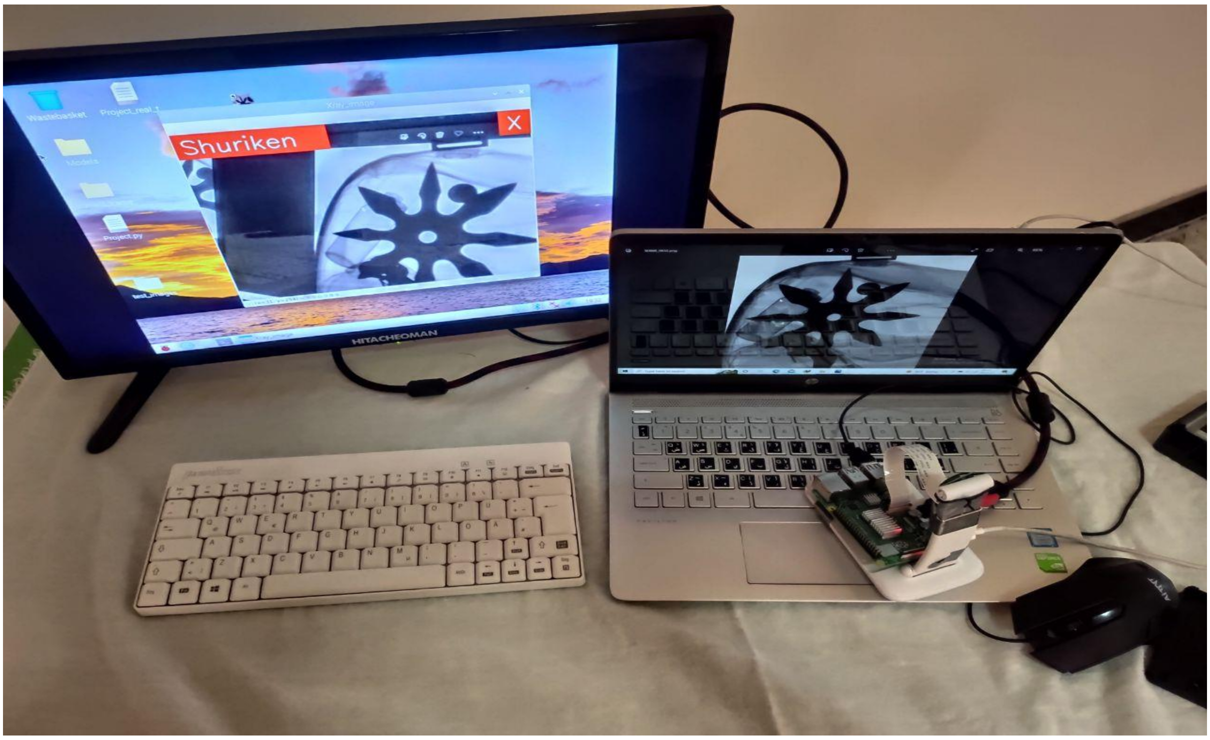
Figure (4.6): Results of TOD proposed system Embedded

Figures (4.7) show the practical results of the classification task. Display dataset (test image) by the computer screen, the camera inserts the test images displayed on the computer screen into the Raspberry pi 4 - B.

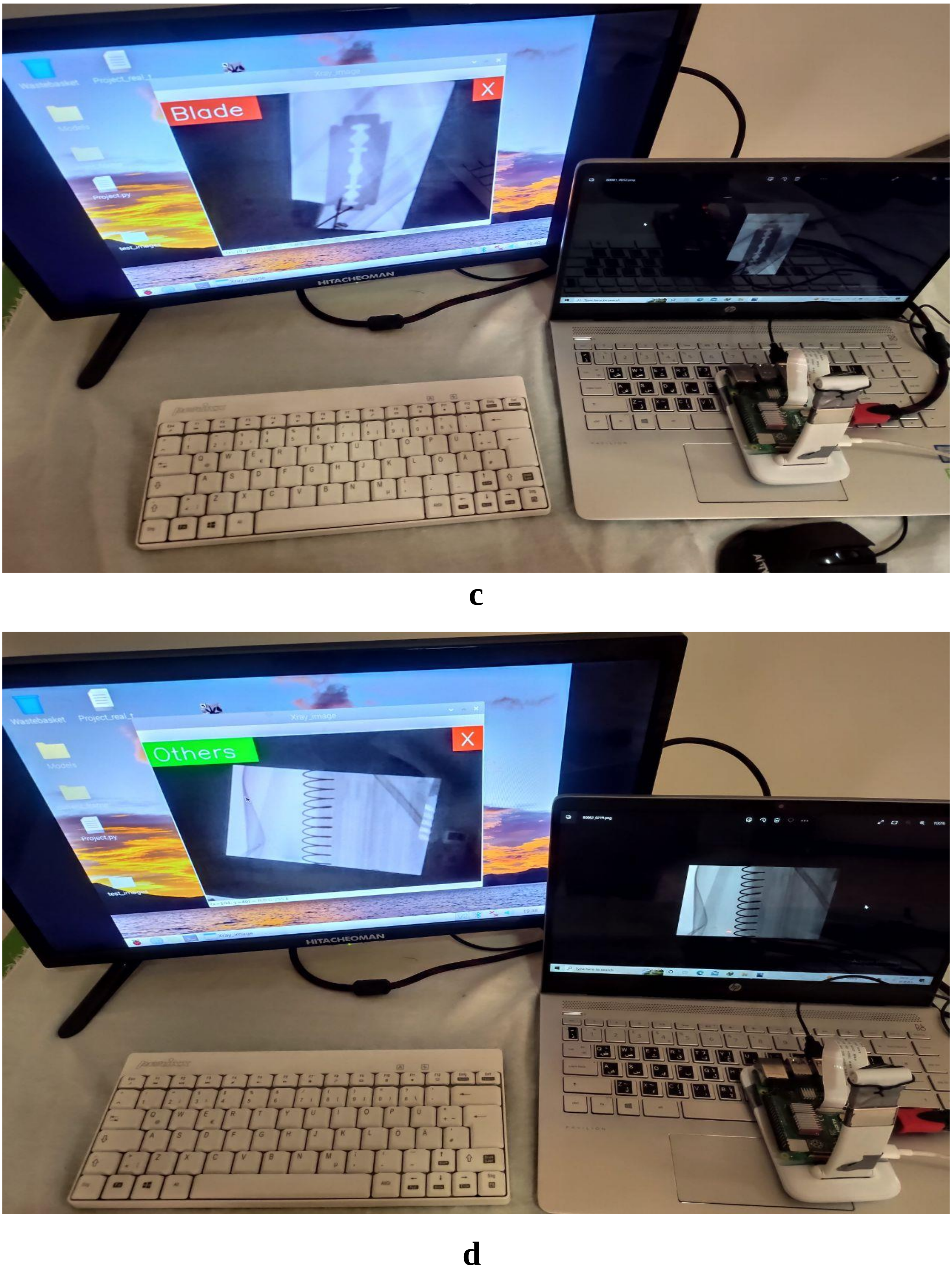
Prediction results appear on the Raspberry pi 4 - B display, four categories; gun, shuriken, blade, and others (a, b, c, d) respectively.



a



b



c

d

Figure(4.7): Demo of the Embedded TOD System. (a) Predict of Gun. (b) Predict of Shuriken. (c) Predict of Blade. (d) Predict of Others

4.4 Results Discussion and Comparison

In this section, the results of the DenseNet121 network using data augmentation and the results of the proposed system will be discussed. Also, simulation and Embedded results will be discussed. Finally, the simulation and Embedded results are compared with the previous literature.

4.4.1 Performance Improvement Using TTA and LR

The DenseNet121 network results after applying the data augmentation technique greatly improved during training due to the reduction of the overfitting. The improvement of the results was not at the required level, so we used the test time augmentation technique. The logistic regression algorithm, and a new feature was added to the prediction Mean, which produced the proposed system, as shown in Figure (3.1). The proposed system achieved high recall in the classification task. Table (4.7) shows a comparison of simulation results for metrics between the DenseNet121 network and the proposed system. The results of the Embedded for the proposed system were better in terms of performance and the effect of noise was less on the results, but the image processing time took 29 seconds compared to DenseNet121 network, which took 4 seconds. Table (4.8) shows a comparison of Embedded results for metrics between the DenseNet121 network and the TOD proposed system.

Table (4.7): Simulation results the proposed DenseNet121and TOD system

model	Recall %	Precision %	F1 score %
DenseNet121 Without TTA	94.1	92.92	93.49
proposed system With TTA	97.68	95.9	96.78

Table (4.8): Results of embedded proposed system

model	Recall %	Precision %	F1 score %
DenseNet121	91.81	89.75	90.76
proposed system	94.25	92.62	93.43

4.4.2 Comparison Between the Results of Simulation and Embedded System Implementation

Simulation results achieved higher classification recall than results achieved on Embedded system. Because of GPU online servers, it has high speed and recall in image processing, while Raspberry Pi's image processing capabilities are limited. Entering the test data set into the raspberry pi using a camera, which reduced the recall due to the presence of noises such as lights and camera reflections. However, good TOD system rating recall results achieved using a low-cost embedded device.

4.4.3 Comparison With State-of-the-Art Results

Table (4.9) shows the comparison between three convolution algorithms for the same image dataset in table (2.1). AlexNet and GoogleNet were selected as the ones with the highest results for classifying images in the field of deep learning for the used publically published dataset. General features were extracted from the AlexNet and GoogleNet. The Nearest neighbour classifier KNN was trained on the features extracted from the above convolution algorithms to obtain the final prediction for four categories. The above networks had poor predictive return so the KNN classification algorithm was used with these networks. While the DenseNet121 network achieved good recall in the final prediction of the task of classifying the data set. The logistic regression algorithm was used in the proposed system to obtain the best possible recall.

Table (4.9): Recall of a four-categories problem

Models	Recall of gun η_1 %	Recall of shuriken η_2 %	Recall of blade η_3 %	Recall of others η_4%	Total Recall η%
AlexNet+KNN [2]	99	100	72.0	93.5	91.2
GoogleNet+KNN [2]	100	100	95	90..0	96.3
Our Proposed system	**100**	**100**	**95**	**95.75**	**97.68**

4.4.4 Comparing the Embedded Results With Related Work

The implementation of TOD system on embedded devices are very rare to our best knowledge. One research paper implementing a TOD system on an embedded device was created using a different dataset [25]. The following are the results of the comparison:

1. The researchers presented a CNN YOL v3 model, to detect x-ray images and then it is classified as hazardous or non-hazardous. Data is transferred from the computer to the FPGA-based PYNQ directly.

2. Mean

   Average

   Precision (mAP) is 98.9%. These results were higher than those achieved by the proposed system implemented on Raspberry Pi4 model B, because the researchers made a binary classification while our proposed system is a multi classification consisting of four classes.
3. The researchers implemented more expensive and complex Embedded, while our proposed system is less expensive and complex.
4. Transferring data from computer to device directly, while our proposed system transmits data using a camera, our method negatively affects recall of results due to noise.

Chapter Five

Conclusion and Future Works

CHAPTER FIVE

CONCLUSION AND FUTURE WORKS

5.1 Conclusions

The interest in using CNN network to detect threat objects. The use of the Threat Objects Detection TOD system in the task of classifying X-ray images has increased due to CNNs achieving high accuracy and speed in baggage screening systems. In this work, A new threat objects detection and classification system has been proposed, tested and implemented successfully. The results of simulation and Embedded implementation, illustrated the ability of the system is to detect the threat objects accurately. The work was implementation using open servers (Kaggle). The results of the proposed system in the classification of X-ray images are: Recall is 97.68%, Precision is 95.9%, F1 Score is 96.78%.

These results have been gained due to.

1. The TOD system was successfully implemented and achieved high results, as the DenseNet121 network achieved high results after applying the data augmentation and Test Time Augmentation technology, with the use of the logistic regression algorithm for the final prediction.
2. A predictive mean feature consisting of DenseNet121 network, TTA Mean and TTA max. Logistic regression inputs consist of DenseNet121 network, TTA max and mean feature, which lead to the improvement of the results.
3. The threat object was detected and classified through X-ray images of the baggage in the GDXray dataset. The dataset contains four classes of baggage (Blade, Gun, Shuriken and Others). The first three categories were detected and classified as containing threatening objects, the last class is classified as not containing a threat object.

4. Implementing the proposed system on Raspberry Pi 4model B and achieve high results in real time.

5.2 Future works

The goal in the future works is to improve results and increase speed for future systems. Suggestions that may lead to a better performing system are as follows:

1. Experiment with other CNN networks to get higher accuracy results.
2. Using other embedded devices efficiently and quickly, such as FPGA.

References

References

[1] T. W. Rogers, N. Jaccard, E. D. Protonotarios, J. Ollier, E. J. Morton, and L. D. Griffin, "Threat Image Projection (TIP) into X-ray images of cargo containers for training humans and machines," in *2016 IEEE International Carnahan Conference on Security Technology (ICCST)*, 2016: IEEE, pp. 1-7.

[2] D. Mery, E. Svec, M. Arias, V. Riffo, J. M. Saavedra, and S. Banerjee, "Modern computer vision techniques for x-ray testing in baggage inspection," *IEEE Transactions on Systems, Man, and Cybernetics: Systems,* vol. 47, no. 4, pp. 682-692, 2016.

[3] L. Zou, T. Yusuke, and I. Hitoshi, "Dangerous objects detection of X-ray images using convolution neural network," in *International Conference on Security with Intelligent Computing and Big-data Services*, 2018: Springer, pp. 714-728.

[4] D. K. Jain, "An evaluation of deep learning based object detection strategies for threat object detection in baggage security imagery," *pattern recognition letters,* vol. 120, pp. 112-119, 2019.

[5] D. Mery, D. Saavedra, and M. Prasad, "X-ray baggage inspection with computer vision: A survey," *IEEE Access,* vol. 8, pp. 145620-145633, 2020.

[6] S. Akcay, M. E. Kundegorski, C. G. Willcocks, and T. P. Breckon, "Using deep convolutional neural network architectures for object classification and detection within x-ray baggage security imagery," *IEEE transactions on information forensics and security,* vol. 13, no. 9, pp. 2203-2215, 2018.

[7] G. Zentai, "X-ray imaging for homeland security," *International Journal of Signal and Imaging Systems Engineering,* vol. 3, no. 1, pp. 13-20, 2010.

[8] A. Bolfing, T. Halbherr, and A. Schwaninger, "How image based factors and human factors contribute to threat detection performance in X-ray aviation security screening," in *symposium of the Austrian HCI and Usability Engineering Group*, 2008: Springer, pp. 419-438.

[9] N. Jaccard, T. W. Rogers, E. J. Morton, and L. D. Griffin, "Automated detection of smuggled high-risk security threats using Deep Learning," in *7th International Conference on Imaging for Crime Detection and Prevention (ICDP 2016)*, 2016: IET, pp. 1-6.

[10] A. Schwaninger, A. Bolfing, T. Halbherr, S. Helman, A. Belyavin, and L. Hay, "The impact of image based factors and training on threat detection performance in X-ray screening," 2008.

[11] S. Michel, S. M. Koller, J. C. de Ruiter, R. Moerland, M. Hogervorst, and A. Schwaninger, "Computer-based training increases efficiency in X-ray image interpretation by aviation security screeners," in *2007 41st Annual IEEE international Carnahan conference on security technology*, 2007: IEEE, pp. 201-206.

[12] S. Akçay, M. E. Kundegorski, M. Devereux, and T. P. Breckon, "Transfer learning using convolutional neural networks for object classification within X-ray baggage security imagery," in *2016 IEEE International Conference on Image Processing (ICIP)*, 2016: IEEE, pp. 1057-1061.

References

[13] N. Jaccard, T. W. Rogers, E. J. Morton, and L. D. Griffin, "Tackling the X-ray cargo inspection challenge using machine learning," in *Anomaly Detection and Imaging with X-Rays (ADIX)*, 2016, vol. 9847: SPIE, pp. 131-143.

[14] S. Akcay and T. P. Breckon, "An evaluation of region based object detection strategies within x-ray baggage security imagery," in *2017 IEEE International Conference on Image Processing (ICIP)*, 2017: IEEE, pp. 1337-1341.

[15] V. Riffo, I. Godoy, and D. Mery, "Handgun detection in single-spectrum multiple X-ray views based on 3d object recognition," *Journal of Nondestructive Evaluation*, vol. 38, no. 3, pp. 1-11, 2019.

[16] I. AYDIN, M. KARAKOSE, and A. Erhan, "A new approach for baggage inspection by using deep convolutional neural networks," in *2018 International Conference on Artificial Intelligence and Data Processing (IDAP)*, 2018: IEEE, pp. 1-6.

[17] J. Yang, Z. Zhao, H. Zhang, and Y. Shi, "Data augmentation for X-ray prohibited item images using generative adversarial networks," *IEEE Access,* vol. 7, pp. 28894-28902, 2019.

[18] Y. Wei and X. Liu, "Dangerous goods detection based on transfer learning in X-ray images," *Neural Computing and Applications,* vol. 32, no. 12, pp. 8711-8724, 2020.

[19] J. K. Dumagpi and Y.-J. Jeong, "Evaluating gan-based image augmentation for threat detection in large-scale xray security images," *Applied Sciences,* vol. 11, no. 1, p. 36, 2020.

[20] P. Steno, A. Alsadoon, P. Prasad, T. Al-Dala'in, and O. H. Alsadoon, "A novel enhanced region proposal network and modified loss function: threat object detection in secure screening using deep learning," *The Journal of Supercomputing,* vol. 77, no. 4, pp. 3840-3869, 2021.

[21] Y. Wei, Z. Zhu, H. Yu, and W. Zhang, "An automated detection model of threat objects for X-ray baggage inspection based on depthwise separable convolution," *Journal of Real-Time Image Processing,* vol. 18, no. 3, pp. 923-935, 2021.

[22] Z. Zhu and F. Xu, "Demonstration of 3-D security imaging at 24 GHz with a 1-D sparse MIMO array," *IEEE Geoscience and Remote Sensing Letters,* vol. 17, no. 12, pp. 2090-2094, 2020.

[23] Y. Wei, Z. Zhu, H. Yu, and W. Zhang, "A real-time Threat Image Projection (TIP) model base on deep learning for X-ray baggage inspection," *Physics Letters A,* vol. 400, p. 127306, 2021.

[24] V. Ponnusamy, D. R. Marur, D. Dhanaskodi, and T. Palaniappan, "Deep Learning-Based X-Ray Baggage Hazardous Object Detection–An FPGA Implementation," *Revue d'Intelligence Artificielle,* vol. 35, no. 5, pp. 431-435, 2021.

[25] S. Yaman and B. Karakaya, "A Novel Normalization Algorithm to Facilitate Pre-Assessment of Covid-19 Disease from Chest X-ray Image and its FPGA implementation," 2022.

References

[26] V. Cutler and S. Paddock, "Use of threat image projection (TIP) to enhance security performance," in *43rd Annual 2009 International Carnahan Conference on Security Technology*, 2009: IEEE, pp. 46-51.

[27] D. Mery, "Computer vision for X-Ray testing," *Switzerland: Springer International Publishing,* vol. 10, pp. 978-3, 2015.

[28] L. W. Schmerr, *Fundamentals of ultrasonic nondestructive evaluation*. Springer, 2016.

[29] D. G. Manolakis, R. B. Lockwood, and T. W. Cooley, *Hyperspectral imaging remote sensing: physics, sensors, and algorithms*. Cambridge University Press, 2016.

[30] D. Mery, V. Riffo, I. Zuccar, and C. Pieringer, "Object recognition in X-ray testing using an efficient search algorithm in multiple views," *Insight-Non-Destructive Testing and Condition Monitoring,* vol. 59, no. 2, pp. 85-92, 2017.

[31] R. Lanier, "Recent developments in X-ray imaging technology," 2012.

[32] I. El Naqa and M. J. Murphy, "What is machine learning?," in *machine learning in radiation oncology*: Springer, 2015, pp. 3-11.

[33] D. P. Penumuru, S. Muthuswamy, and P. Karumbu, "Identification and classification of materials using machine vision and machine learning in the context of industry 4.0," *Journal of Intelligent Manufacturing,* vol. 31, no. 5, pp. 1229-1241, 2020.

[34] H. Kukreja, N. Bharath, C. Siddesh, and S. Kuldeep, "An introduction to artificial neural network," *Int J Adv Res Innov Ideas Educ,* vol. 1, pp. 27-30, 2016.

[35] I. El Naqa and M. J. Murphy, "What Are Machine and Deep Learning?," in *Machine and Deep Learning in Oncology, Medical Physics and Radiology*: Springer, 2022, pp. 3-15.

[36] D. Saavedra, S. Banerjee, and D. Mery, "Detection of threat objects in baggage inspection with X-ray images using deep learning," *Neural Computing and Applications,* vol. 33, no. 13, pp. 7803-7819, 2021.

[37] R. Riz à Porta, Y. Sterchi, and A. Schwaninger, "How Realistic Is Threat Image Projection for X-ray Baggage Screening?," *Sensors,* vol. 22, no. 6, p. 2220, 2022.

[38] J. A. Seibert, "X-ray imaging physics for nuclear medicine technologists. Part 1: Basic principles of x-ray production," *Journal of nuclear medicine technology,* vol. 32, no. 3, pp. 139-147, 2004.

[39] A. Maier, S. Steidl, V. Christlein, and J. Hornegger, "Medical imaging systems: An introductory guide," 2018.

[40] V. Nasteski, "An overview of the supervised machine learning methods," *Horizons. b,* vol. 4, pp. 51-62, 2017.

[41] G. Kostopoulos, S. Karlos, S. Kotsiantis, and O. Ragos, "Semi-supervised regression: A recent review," *Journal of Intelligent & Fuzzy Systems,* vol. 35, no. 2, pp. 1483-1500, 2018.

References

[42] S. Youn and D. McLeod, "A comparative study for email classification," in *Advances and innovations in systems, computing sciences and software engineering*: Springer, 2007, pp. 387-391.

[43] Q. Bi, K. E. Goodman, J. Kaminsky, and J. Lessler, "What is machine learning? A primer for the epidemiologist," *American journal of epidemiology,* vol. 188, no. 12, pp. 2222-2239, 2019.

[44] Q. Dong, X. Zhu, and S. Gong, "Single-label multi-class image classification by deep logistic regression," in *Proceedings of the AAAI conference on artificial intelligence*, 2019, vol. 33, no. 01, pp. 3486-3493.

[45] S. Raschka and V. Mirjalili, *Python machine learning: Machine learning and deep learning with Python, scikit-learn, and TensorFlow 2*. Packt Publishing Ltd, 2019.

[46] A. Robles-Guerrero, T. Saucedo-Anaya, E. González-Ramírez, and J. I. De La Rosa-Vargas, "Analysis of a multiclass classification problem by lasso logistic regression and singular value decomposition to identify sound patterns in queenless bee colonies," *Computers and Electronics in Agriculture,* vol. 159, pp. 69-74, 2019.

[47] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700-4708.

[48]

[49] G. Guo, H. Wang, D. Bell, Y. Bi, and K. Greer, "KNN model-based approach in classification," in *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*, 2003: Springer, pp. 986-996.

[50] M. Amrane, S. Oukid, I. Gagaoua, and T. Ensari, "Breast cancer classification using machine learning," in *2018 Electric Electronics, Computer Science, Biomedical Engineerings' Meeting (EBBT)*, 2018: IEEE, pp. 1-4.

[51] R. A. Nugrahaeni and K. Mutijarsa, "Comparative analysis of machine learning KNN, SVM, and random forests algorithm for facial expression classification," in *2016 International Seminar on Application for Technology of Information and Communication (ISemantic)*, 2016: IEEE, pp. 163-168.

[52] K. Shah, H. Patel, D. Sanghvi, and M. Shah, "A comparative analysis of logistic regression, random forest and KNN models for the text classification," *Augmented Human Research,* vol. 5, no. 1, pp. 1-16, 2020.

[53] A. Dongare, R. Kharde, and A. D. Kachare, "Introduction to artificial neural network," *International Journal of Engineering and Innovative Technology (IJEIT),* vol. 2, no. 1, pp. 189-194, 2012.

[54] D. J. Livingstone, *Artificial neural networks: methods and applications*. Springer, 2008.

[55] J. Zou, Y. Han, and S.-S. So, "Overview of artificial neural networks," *Artificial Neural Networks,* pp. 14-22, 2008.

[56] N. Shone, T. N. Ngoc, V. D. Phai, and Q. Shi, "A deep learning approach to network intrusion detection," *IEEE transactions on emerging topics in computational intelligence,* vol. 2, no. 1, pp. 41-50, 2018.

References

[57] A. Kamilaris and F. X. Prenafeta-Boldú, "Deep learning in agriculture: A survey," *Computers and electronics in agriculture,* vol. 147, pp. 70-90, 2018.

[58] N. Khare *et al.*, "Smo-dnn: Spider monkey optimization and deep neural network hybrid classifier model for intrusion detection," *Electronics,* vol. 9, no. 4, p. 692, 2020.

[59] T. Beysolow II, *Introduction to deep learning using R: A step-by-step guide to learning and implementing deep learning models using R*. Apress, 2017.

[60] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems,* vol. 25, 2012.

[61] J. Subramanian and R. Simon, "Overfitting in prediction models–is it a problem only in high dimensions?," *Contemporary clinical trials,* vol. 36, no. 2, pp. 636-641, 2013.

[62] S. C. Wong, A. Gatt, V. Stamatescu, and M. D. McDonnell, "Understanding data augmentation for classification: when to warp?," in *2016 international conference on digital image computing: techniques and applications (DICTA),* 2016: IEEE, pp. 1-6.

[63] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *Journal of big data,* vol. 6, no. 1, pp. 1-48, 2019.

[64] M. Sardogan, A. Tuncer, and Y. Ozen, "Plant leaf disease detection and classification based on CNN with LVQ algorithm," in *2018 3rd International Conference on Computer Science and Engineering (UBMK),* 2018: IEEE, pp. 382-385.

[65] J. Wang and L. Perez, "The effectiveness of data augmentation in image classification using deep learning," *Convolutional Neural Networks Vis. Recognit,* vol. 11, pp. 1-8, 2017.

[66] I. Kim, Y. Kim, and S. Kim, "Learning loss for test-time augmentation," *Advances in Neural Information Processing Systems,* vol. 33, pp. 4163-4174, 2020.

[67] F. Perez, C. Vasconcelos, S. Avila, and E. Valle, "Data augmentation for skin lesion analysis," in *OR 2.0 Context-Aware Operating Theaters, Computer Assisted Robotic Endoscopy, Clinical Image-Based Procedures, and Skin Image Analysis*: Springer, 2018, pp. 303-311.

[68] I. Kandel and M. Castelli, "Improving convolutional neural networks performance for image classification using test time augmentation: a case study using MURA dataset," *Health information science and systems,* vol. 9, no. 1, pp. 1-22, 2021.

[69] H. He, J. Teng, and Y. Yuan, "Anomaly Detection with Test Time Augmentation and Consistency Evaluation," *arXiv preprint arXiv:2206.02345,* 2022.

[70] O. Lucena, A. Junior, V. Moia, R. Souza, E. Valle, and R. Lotufo, "Transfer learning using convolutional neural networks for face anti-spoofing," in *International conference image analysis and recognition,* 2017: Springer, pp. 27-34.

References

[71] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," in *2017 international conference on engineering and technology (ICET)*, 2017: Ieee, pp. 1-6.

[72] M. A. Mohammed, B. Al-Khateeb, A. N. Rashid, D. A. Ibrahim, M. K. Abd Ghani, and S. A. Mostafa, "Neural network and multi-fractal dimension features for breast cancer classification from ultrasound images," *Computers & Electrical Engineering,* vol. 70, pp. 871-882, 2018.

[73] A. M. Taqi, F. Al-Azzo, M. Mariofanna, and J. M. Al-Saadi, "Classification and discrimination of focal and non-focal EEG signals based on deep neural network," in *2017 international conference on current research in computer science and information technology (ICCIT)*, 2017: IEEE, pp. 86-92.

[74] P. Kim, "Convolutional neural network," in *MATLAB deep learning*: Springer, 2017, pp. 121-147.

[75] L. Lu, Y. Zheng, G. Carneiro, and L. Yang, "Deep learning and convolutional neural networks for medical image computing," *Advances in computer vision and pattern recognition,* vol. 10, pp. 978-3, 2017.

[76] U. Michelucci, "Applied Deep Learning," *A Case-Based Approach to Understanding Deep Neural Networks*, 2018.

[77] J. Wu, "Introduction to convolutional neural networks," *National Key Lab for Novel Software Technology. Nanjing University. China,* vol. 5, no. 23, p. 495, 2017.

[78] J. D. Prusa and T. M. Khoshgoftaar, "Improving deep neural network design with new text data representations," *Journal of Big Data,* vol. 4, no. 1, pp. 1-16, 2017.

[79] M. Francis and C. Deisy, "Disease detection and classification in agricultural plants using convolutional neural networks—a visual understanding," in *2019 6th International Conference on Signal Processing and Integrated Networks (SPIN)*, 2019: IEEE, pp. 1063-1068.

[80] Z. Li, F. Liu, W. Yang, S. Peng, and J. Zhou, "A survey of convolutional neural networks: analysis, applications, and prospects," *IEEE transactions on neural networks and learning systems,* 2021.

[81] B. Gao and L. Pavel, "On the properties of the softmax function with application in game theory and reinforcement learning," *arXiv preprint arXiv:1704.00805,* 2017.

[82] K. Eckle and J. Schmidt-Hieber, "A comparison of deep networks with ReLU activation function and linear spline-type methods," *Neural Networks,* vol. 110, pp. 232-242, 2019.

[83] J. Xu, Z. Li, B. Du, M. Zhang, and J. Liu, "Reluplex made more practical: Leaky ReLU," in *2020 IEEE Symposium on Computers and communications (ISCC)*, 2020: IEEE, pp. 1-7.

[84] J. Jin, C. Zhang, F. Feng, W. Na, J. Ma, and Q.-J. Zhang, "Deep neural network technique for high-dimensional microwave modeling and applications to parameter extraction of microwave filters," *IEEE Transactions on Microwave Theory and Techniques,* vol. 67, no. 10, pp. 4140-4155, 2019.

References

[85] A. Solanki and S. Pandey, "Music instrument recognition using deep convolutional neural networks," *International Journal of Information Technology*, pp. 1-10, 2019.

[86] X. Fan, W. Jiang, H. Luo, and M. Fei, "Spherereid: Deep hypersphere manifold embedding for person re-identification," *Journal of Visual Communication and Image Representation*, vol. 60, pp. 51-58, 2019.

[87] X. Geng, J. Lin, B. Zhao, A. Kong, M. M. S. Aly, and V. Chandrasekhar, "Hardware-aware softmax approximation for deep neural networks," in *Asian Conference on Computer Vision*, 2018: Springer, pp. 107-122.

[88] S. S. Basha, S. R. Dubey, V. Pulabaigari, and S. Mukherjee, "Impact of fully connected layers on performance of convolutional neural networks for image classification," *Neurocomputing*, vol. 378, pp. 112-119, 2020.

[89] L. Alzubaidi *et al.*, "Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions," *Journal of big Data*, vol. 8, no. 1, pp. 1-74, 2021.

[90] H. Qassim, A. Verma, and D. Feinzimer, "Compressed residual-VGG16 CNN model for big data places image recognition," in *2018 IEEE 8th annual computing and communication workshop and conference (CCWC)*, 2018: IEEE, pp. 169-175.

[91] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1-9.

[92] X. Han, Y. Zhong, L. Cao, and L. Zhang, "Pre-trained alexnet architecture with pyramid pooling and supervision for high spatial resolution remote sensing image scene classification," *Remote Sensing*, vol. 9, no. 8, p. 848, 2017.

[93] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[94] Q. Ji, J. Huang, W. He, and Y. Sun, "Optimized deep convolutional neural networks for identification of macular diseases from optical coherence tomography images," *Algorithms*, vol. 12, no. 3, p. 51, 2019.

[95] R. Lanier.

[96] S. Akcay and T. Breckon, "Towards automatic threat detection: A survey of advances of deep learning within X-ray security imaging," *Pattern Recognition*, vol. 122, p. 108245, 2022.

[97] A. Thakur, *Approaching (almost) any machine learning problem*. Abhishek Thakur, 2020.

[98] A. Santra and C. J. Christy, "Genetic algorithm and confusion matrix for document clustering," *International Journal of Computer Science Issues (IJCSI)*, vol. 9, no. 1, p. 322, 2012.

[99] Y. Rana, "Python: Simple though an Important Programming language," *International Research Journal of Engineering and Technology (IRJET)*, vol. 6, no. 2, pp. 1856-1858, 2019.

[100] K. Srinath, "Python–the fastest growing programming language," *International Research Journal of Engineering and Technology*, vol. 4, no. 12, pp. 354-357, 2017.

References

[101] kaggle.com/learn.

[102] G. Wang, "THE APPLICATION STUDIES OF EMBEDDED SYSTEM DEVELOPMENT PLATFORMS."

[103] G. Sun, P. Simmons, and B. Decker, "Designing a Low-Cost Embedded System Development Board with Convertible Intelligent Layers," in *2020 CIEC*, 2021.

[104] M. Barr and A. Massa, *Programming embedded systems: with C and GNU development tools*. " O'Reilly Media, Inc.", 2006.

[105] T. Noergaard, *Demystifying embedded systems middleware*. Elsevier, 2010.

[106] J. Ganssle *et al.*, *Embedded Hardware: Know It All*. Newnes, 2007.

[107] K. Khapper, "Embedded Systems–A look into Optimized Energy Metering Devices," *International Journal of Research in Engineering, Science and Management*, vol. 4, no. 1, pp. 19-25, 2021.

[108] K. Pothuganti, A. Haile, and S. Pothuganti, "A comparative study of real time operating systems for embedded systems," *International Journal of Innovative Research in Computer and Communication Engineering*, vol. 4, no. 6, p. 12008, 2016.

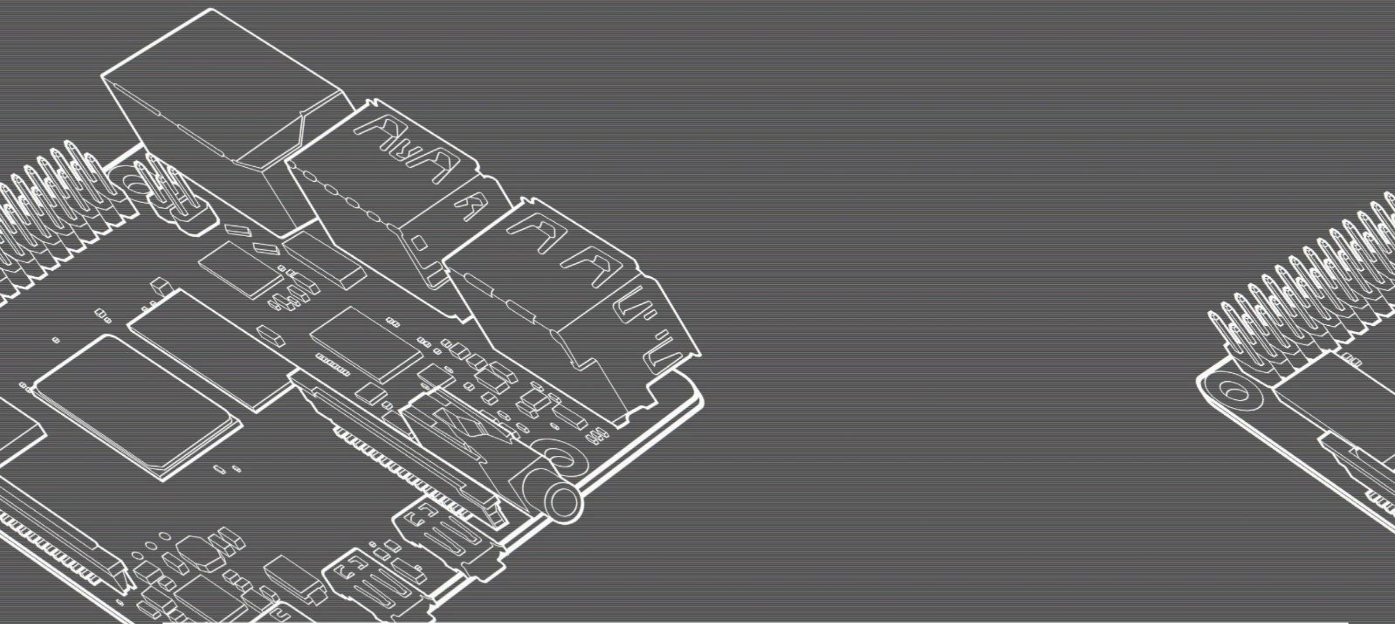
[109] https://www.raspberrypi.com/products/.

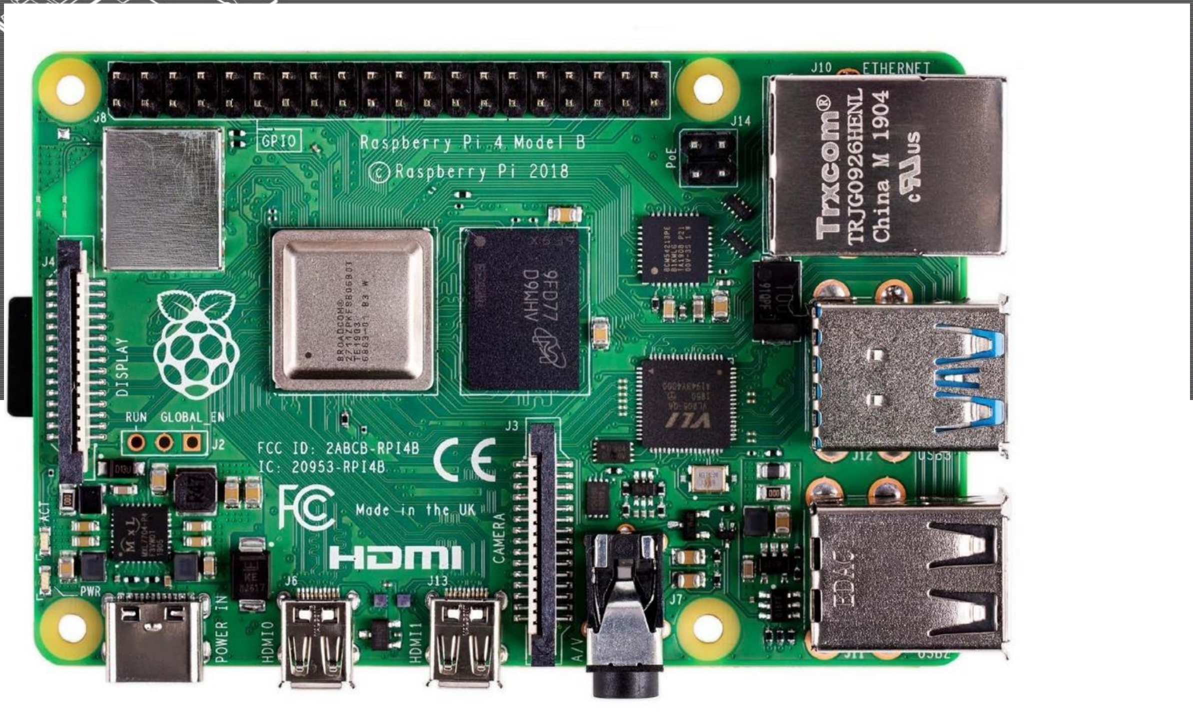
Appendix

Raspberry pi4 model B

Raspberry Pi 4 Computer

Model B





Raspberry Pi 4 Model B is the latest product in the popular Raspberry Pi range of computers. It offers ground-breaking increases in processor speed, multimedia performance, memory, and connectivity compared to the prior-generation Raspberry Pi 3 Model B+, while retaining backwards compatibility and similar power consumption. For the end user, Raspberry Pi 4 Model B provides desktop performance comparable to entry-level x86 PC systems.

This product's key features include a high-performance 64-bit quad-core processor, dual-display support at resolutions up to 4K via a pair of micro-HDMI ports, Embedded video decode at up to 4Kp60, up to 4GB of RAM, dual-band 2.4/5.0 GHz wireless LAN, Bluetooth 5.0, Gigabit Ethernet, USB 3.0, and PoE capability (via a separate PoE HAT add-on).

Thedual-bandwirelessLANandBluetoothhavemodularcompliancecertification, allowingtheboardtobedesignedintoendproductswithsignificantlyreduced compliance testing, improving both cost and time to market.

Specification

Processor:	Broadcom BCM2711, quad-core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz
Memory:	1GB, 2GB or 4GB LPDDR4 (depending on model)
Connectivity:	2.4 GHz and 5.0 GHz IEEE 802.11b/g/n/ac wireless LAN, Bluetooth 5.0, BLE Gigabit Ethernet 2 × USB 3.0 ports 2 × USB 2.0 ports.
GPIO:	Standard 40-pin GPIO header (fully backwards-compatible with previous boards)
Video & sound:	2 × micro HDMI ports (up to 4Kp60 supported) 2-lane MIPI DSI display port 2-lane MIPI CSI camera port 4-pole stereo audio and composite video port

Multimedia:	H.265 (4Kp60 decode);
	H.264 (1080p60 decode, 1080p30 encode);
	OpenGL ES, 3.0 graphics

SD card support:	Micro SD card slot for loading operating system and data storage
Input power:	5V DC via USB-C connector (minimum 3A[1])

Specification

5V DC via GPIO header (minimum 3A[1])
Power over Ethernet (PoE)–enabled
(requires separate PoE HAT)

Environment: Operating temperature 0–50ºC

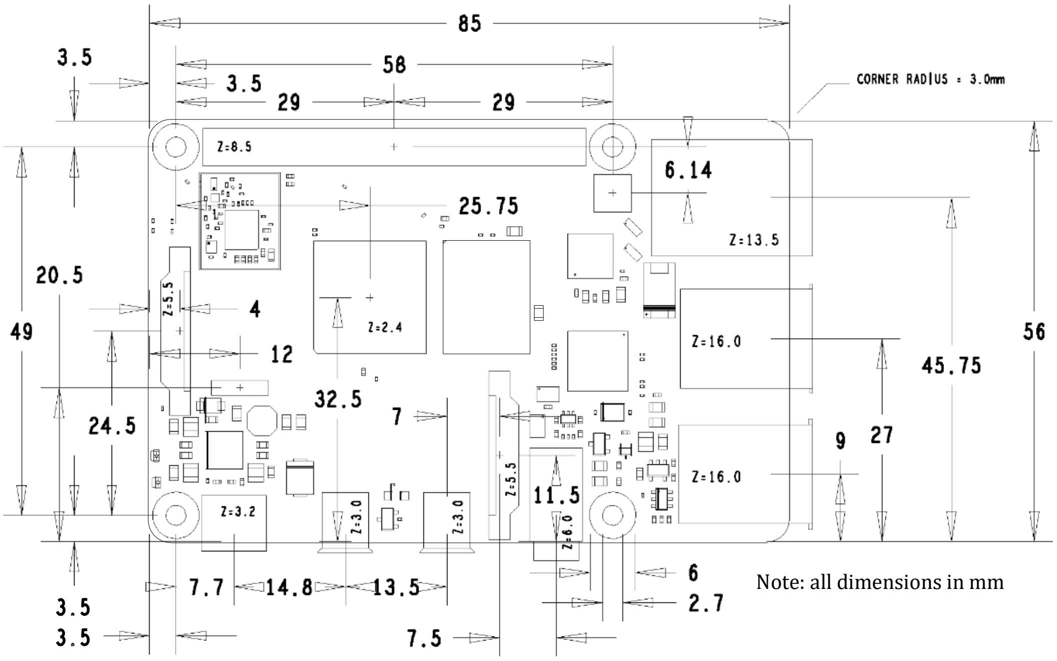
Compliance: For a full list of local and regional product approvals, please visit

https://ww w.raspberrypi.org/documentation/

Embedded/raspberrypi/conformity.md

Production lifetime: The Raspberry Pi 4 Model B will remain in production until at least January 2026.

PhysicalSpecifications



Note: all dimensions in mm

WARNINGS

- This product should only be connected to an external power supply rated at 5V/3A DC or 5.1V/ 3A DC minimum[1]. Any external power supply used with the Raspberry Pi 4 Model B shall comply with relevant regulations and standards applicable in the country of intended use.

- This product should be operated in a well-ventilated environment and, if used inside a case, the case should not be covered.

- This product should be placed on a stable, flat, non-conductive surface in use and should not be contacted by conductive items.

- The connection of incompatible devices to the GPIO connection may affect compliance and result in damage to the unit and invalidate the warranty.

- All peripherals used with this product should comply with relevant standards for the country of use and be marked accordingly to ensure that safety and performance requirements are met. These articles

PhysicalSpecifications

include but are not limited to keyboards, monitors and mice when used in conjunction with the Raspberry Pi.

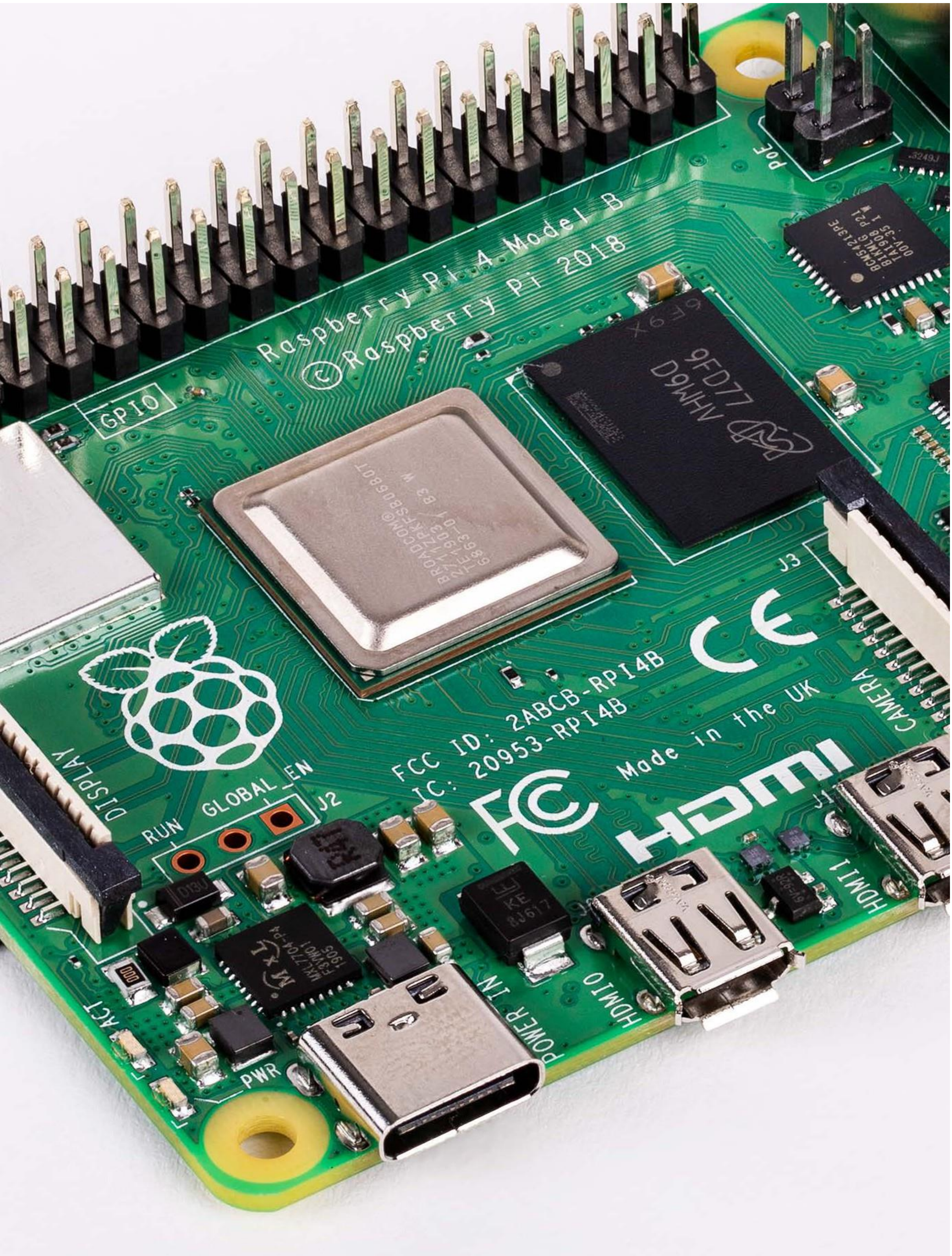
- Where peripherals are connected that do not include the cable or connector, the cable or connector must offer adequate insulation and operation in order that the relevant performance and safety requirements are met.

SAFETY INSTRUCTIONS

To avoid malfunction or damage to this product please observe the following:

- Do not expose to water, moisture or place on a conductive surface whilst in operation.
- Do not expose it to heat from any source; Raspberry Pi 4 Model B is designed for reliable operation at normal ambient room temperatures.
- Take care whilst handling to avoid mechanical or electrical damage to the printed circuit board and connectors.
- Avoid handling the printed circuit board whilst it is powered and only handle by the edges to minimise the risk of electrostatic discharge damage.

[1] A good quality 2.5A power supply can be used if downstream USB peripherals consume less than 500mA in total.


Raspberry Pi 4 Model B
© Raspberry Pi 2018
GPIO
PoE
J3
CAMERA
DISPLAY
GLOBAL_EN
RUN
J2
FCC ID: 2ABCB-RPI4B
IC: 20953-RPI4B
Made in the UK
HDMI
HDMI0
HDMI1
POWER IN
PWR
ACT
BROADCOM
9FD77
D9WHV

www.raspberrypi.org

Raspberry Pi

الخلاصة

كشف كائنات التهديد (TOD) هو نظام يستخدم لمسح صور الأشعة السينية لأمتعة الركاب بغرض الكشف عن كائنات التهديد وتصنيفها. التفتيش البشري مهمة تستغرق وقتًا طويلاً وعملية بطيئة ، حيث حدثت العديد من الأخطاء البشرية الإضافية ، خاصة في ساعات الذروة. في الآونة الأخيرة ، تم استخدام تقنيات جديدة في أنظمة TOD لزيادة الدقة ووتيرة عملية التفتيش مثل التعلم العميق المستخدم. حققت مناهج التعلم العميق أداءً عاليًا وسرعة في معالجة الصور بالأشعة السينية لاكتشاف عناصر التهديد أثناء فحص الأمتعة. تم استخدام شبكات الالتفاف العصبية (CNNs) في مهمة تصنيف صور الأشعة السينية للحصول على أداء قوي أثناء عملية فحص الأمتعة.

لتحقيق اكتشاف عالي الدقة لأجسام التهديد ، تم استخدام خوارزمية DenseNet 121 مع زيادة وقت الاختبار (TTA). بيئة تنفيذ العمل هي خوادم مفتوحة (Kaggle) ، يتم من خلالها تنزيل شبكة DenseNet 121 وخوارزمية الانحدار اللوجستي. تم تدريب شبكة DenseNet 121 واختبارها على مجموعة بيانات GDX-ray التي تحتوي على صور للأمتعة لأربعة أنواع من (البنادق واداة النينجا وشفرات الحلاقة وغيرها). يستخدم النظام المقترح خوارزمية الانحدار اللوجستي للحصول على التنبؤ النهائي لصور الأشعة السينية. كان النظام المقترح قادرًا على التفوق في الأداء على خوارزمية AlexNet وخوارزمية GoogleNet ، التي استخدمت نفس مجموعة البيانات ، نتائج الأسترجاع لتصنيف الصور 97.68٪. بقدر ما نعلم ، لم يتم تنفيذ مجموعة بيانات GDX-ray على جهاز مضمن باستخدام طراز Raspberry Pi4 B. وحققت مرحلة التنفيذ نتائج عالية في الاسترجاع لتصنيف الصور في الوقت الفعلي بنسبة 94.25٪.

جمهورية العراق

وزارة التعليم العالي والبحث العلمي

جامعة بابــل / كلية الهندسة

قسم الهندسة الكهربائية

اكتشاف وتصنيف العناصرالخطرة في صور الاشعة السينية باستخدام التعلم العميق والنظام المدمج

رســالـــة
مقدمة الى كلية الهندسة / جامعة بابل
كجزء من متطلبات نيل درجة الماجستير في الهندسة/ الهندسة الكهربائية/
الكترونيك صناعي

من قبل

سيف سرمد حسين عمران

اشراف

الاستاذ المساعد الدكتور هلال عبد الحسين عبود

2022 A.D. **1444 A.H.**