

**Ain Shams University**  
**Faculty of Computer & Information Sciences**

**Computer Science Department**

Project Title

*This is the cover page, so put your own design, colors*

[ITIDA Logo if exists]

**July 2016**

[Sponsor Logo if exists]



**By:**

Team Member 1 [Department]

Team Member 2 [Department]

Team Member 3 [Department]

Team Member 4 [Department]

Project Title

*You can change to alignment of things the way you want.*

**Under Supervision of :**

[Supervisor 1]

[Supervisor Title],

………………. Department,

Faculty of Computer and Information Sciences,

Ain Shams University.

[Supervisor 2]

[Supervisor Title],

………………. Department,

Faculty of Computer and Information Sciences,

Ain Shams University.

**Ain Shams University**  
**Faculty of Computer & Information Sciences**

**Computer Science Department**

# Acknowledgement

If you want to include thank you notes to any one you should put it here. (The acknowledgement is optional)

# Abstract

The abstract is a one page summary of the whole project including: why the project is needed, what are the main features of the project and what are the final results obtained by the developed system.

It’s the most important page in the whole documentation, it should be the last thing you write.

**Table of Contents**

[Acknowledgement i](#_Toc516166322)

[Abstract ii](#_Toc516166323)

[List of Figures iv](#_Toc516166324)

[List of Abbreviations v](#_Toc516166325)

[1- Introduction 1](#_Toc516166326)

[1.1 Motivation 1](#_Toc516166327)

[1.2 Problem Definition 1](#_Toc516166328)

[1.3 Objective 1](#_Toc516166329)

[1.4 Time Plan 1](#_Toc516166330)

[1.5 Document Organization 1](#_Toc516166331)

[2- Background 2](#_Toc516166332)

[3- Analysis and Design 3](#_Toc516166333)

[3.1 System Overview 3](#_Toc516166334)

[3.1.1 System Architecture 3](#_Toc516166335)

[3.1.2 System Users 3](#_Toc516166336)

[3.2 System Analysis & Design 3](#_Toc516166337)

[3.2.1 Use Case Diagram 3](#_Toc516166338)

[3.2.2 Class Diagram 3](#_Toc516166339)

[3.2.3 Sequence Diagram 3](#_Toc516166340)

[3.2.4 Database Diagram 4](#_Toc516166341)

[4- Implementation and Testing 5](#_Toc516166342)

[**4.1 Implementation** 5](#_Toc516166343)

[5- User Manual 11](#_Toc516166344)

[6- Conclusion and Future Work 12](#_Toc516166345)

[6.1 Conclusion 12](#_Toc516166346)

[6.2 Future Work 12](#_Toc516166347)

[References 13](#_Toc516166348)

# List of Figures

[**Figure 1- Neural Network general architecture** 4](#_Toc417423219)

Note: Always number your figures and include a caption under each one like this. Then if you update the list above it’ll be updated automatically.

When adding a figure, right click on the image -> insert caption.

After you finish the document, write click on the table and choose update field, then update entire table.



**Figure 1- Neural Network general architecture**

Add list of Tables if you have tables in your text in the same manner

# List of Abbreviations

|  |  |  |
| --- | --- | --- |
| Abbreviation | What the abbreviation stands for |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Note: Any abbreviations used throughout the document should be included here. The list should be sorted **alphabetically**.

# Introduction

## 1.1 Motivation

This section should include why the project is needed. This includes what is happening in the field of the project that motivated the idea.

## 1.2 Problem Definition

What is the specific problem the project is solving?

## 1.3 Objective

What are the objectives of the project and what was done to achieve these objectives?

## 1.4 Time Plan

## 1.5 Document Organization

Include a paragraph for each chapter describing what was discussed in this chapter starting with chapter 2.

# Background

This chapter should introduce the following:

* A detailed description of the field of the project.
* All the scientific background related to the project.
* A survey of the work done in the field.
* Description of existing similar systems.
* Description of any technology used: Bluetooth, GPS… etc.

# Analysis and Design

## 3.1 System Overview

### 3.1.1 System Architecture

Include a figure of the system architecture and a description of all modules.

You may add Functional and non-functional requirements section –If needed–

### 3.1.2 System Users

1. *Intended Users:*

To whom the system is built, and how each group of users will use the system.

1. *User Characteristics*

What kind of experience or skills are required from the users to be able to operate the project effectively.

## 3.2 System Analysis & Design

### 3.2.1 Use Case Diagram

The use case diagram + fully dressed use cases describing each function of the project if applicable.

### 3.2.2 Class Diagram

The diagram + description of all the main classes if applicable.

### 3.2.3 Sequence Diagram

if applicable

### 3.2.4 Database Diagram

If you are implementing a database include the database schema plus a description of the tables.

# Implementation and Testing

## **4.1 Implementation**

In this section, we talk about the implementation phases for each

module , the talk contains deep technicalities that has been used and tried on

our system.

We also mentioned a lot of references that we have used during the

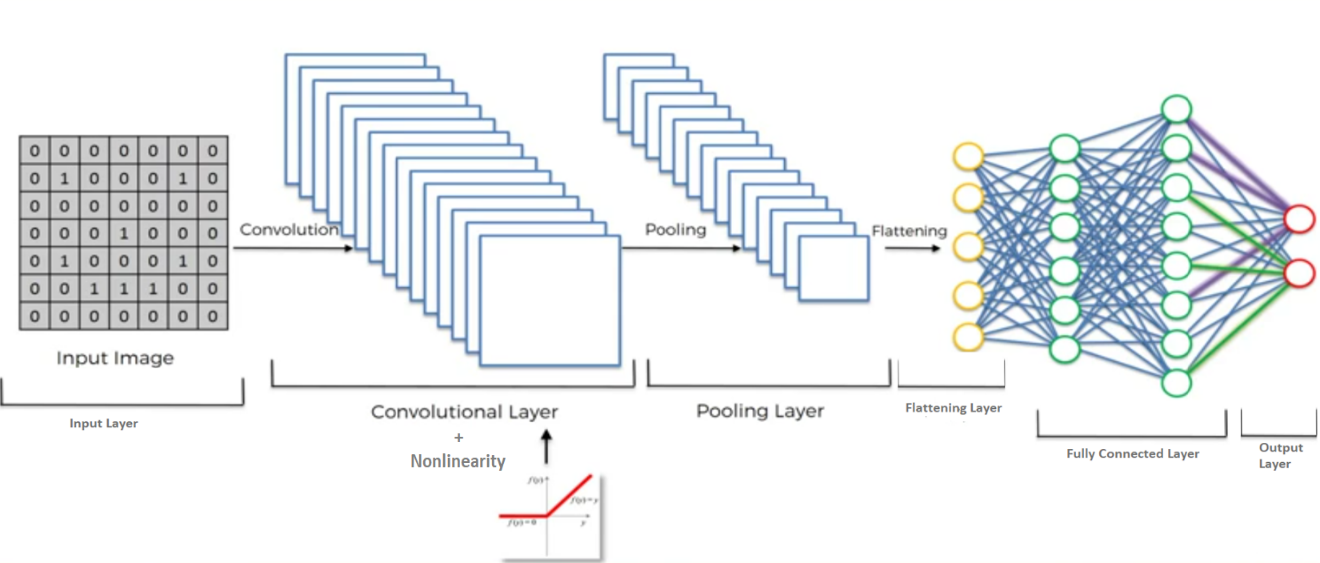
implementation phases, so, you can check them if you want to find more details about something not clear in the illustration.

**4.1.1 Convolutional neural networks**

In this subsection, we will talk about the convolutional neural networks, we will talk about its components, architectures, and features extraction.

The convolutional neural networks used in our model for extracting the features of an input image.

**1) CNN Components**

****

**Figure 4.1.1 - Convolutional Neural Networks general architecture**

**1**-**Input layer**

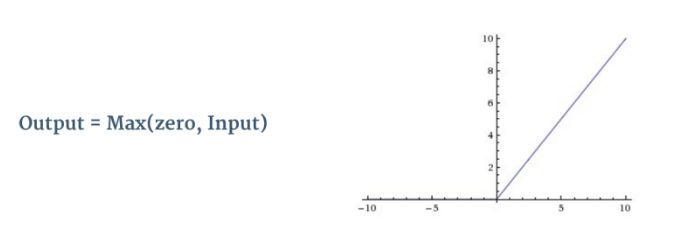
The input layer of a neural network is composed of artificial input neurons, and brings the initial data into the system for further processing by subsequent layers of artificial neurons. The input layer is the very beginning of the workflow for the artificial neural network.

**2-Convolution Layer**

The primary purpose of Convolution is to extract features from the input image. Convolution preserves the spatial relationship between pixels by learning image features using small squares of input data.

**3-Nonlinearity(RELU)**

An additional operation called ReLU has been used after every Convolution operation in **Figure 3** above. Its output is given by:

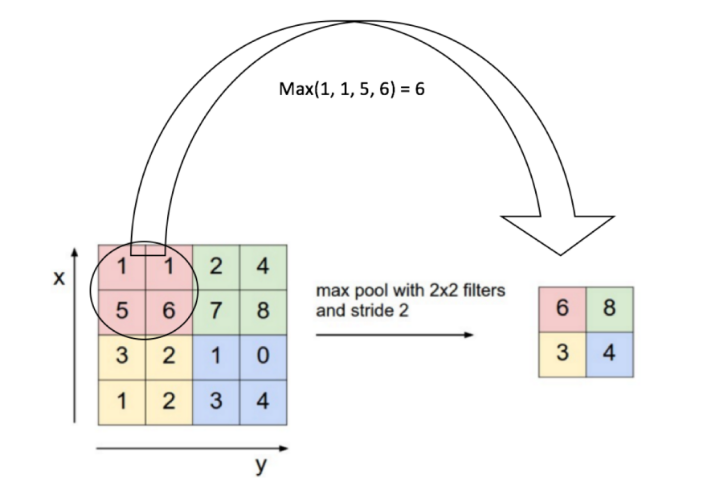


**Figure 4.1.2 - the RELU operation**

ReLU is an element wise operation (applied per pixel) and replaces all negative pixel values in the feature map by zero. The purpose of ReLU is to introduce non-linearity, since most of the real-world data would be non-linear (Convolution is a linear operation – element wise matrix multiplication and addition, so we account for non-linearity by introducing a non-linear function like ReLU).

**4- Pooling layer**

Spatial Pooling (also called subsampling or downsampling) reduces the dimensionality of each feature map but retains the most important information. Spatial Pooling can be of different types: Max, Average, Sum etc.



**Figure 4.1.3 - Max Pooling**

combines the outputs of neuron clusters at one layer into a single neuron in the next layer.

**5-Flattening Layer**

convert the output of the convolutional part of the CNN into a 1D feature vector, to be used by the ANN part of it. This operation is called flattening. It gets the output of the convolutional layers, flattens all its structure to create a single long feature vector to be used by the dense layer for the final classification.

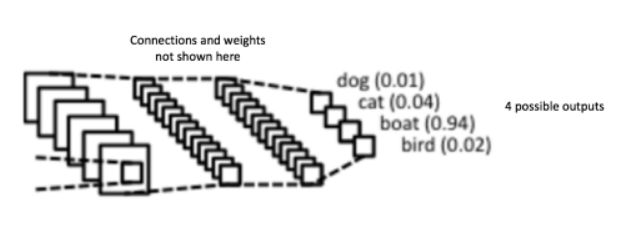
**6-Fully connected layer**

The Fully Connected layer is a traditional Multi Layer Perceptron that uses a softmax activation function in the output layer (other classifiers like SVM can also be used). The term “Fully Connected” implies that every neuron in the previous layer is connected to every neuron on the next layer

The purpose of the Fully Connected layer is to use output features of the flattening layer for classifying the input image into various classes based on the training dataset.

**7-Output layer**

The last layer of the CNN that has m artificial neurons for m classes, at the end of classification each neuron will have a probability that the input image belongs to its class.



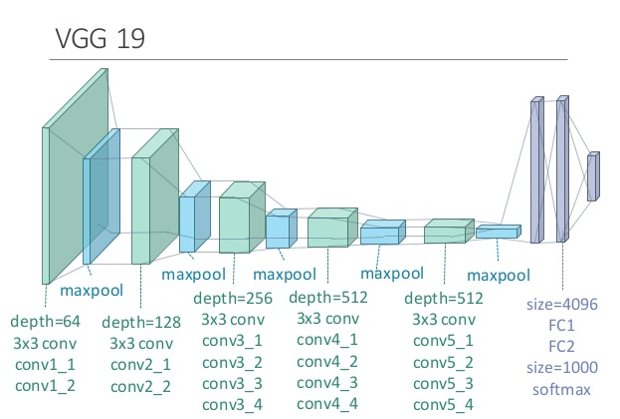
**Figure 4.1.4 - fully connected layer and output layer**

The sum of output probabilities from the output Layer is 1. This is ensured by using the [Softmax](http://cs231n.github.io/linear-classify/" \l "softmax" \t "_blank) as the activation function in the output layer

**2) CNN Architectures**

**-** There is several CNN architectures:AlexNet**,** VGG**,** GoogLeNet**,** ResNet etc.

- In our model we will use **VGG19**

****

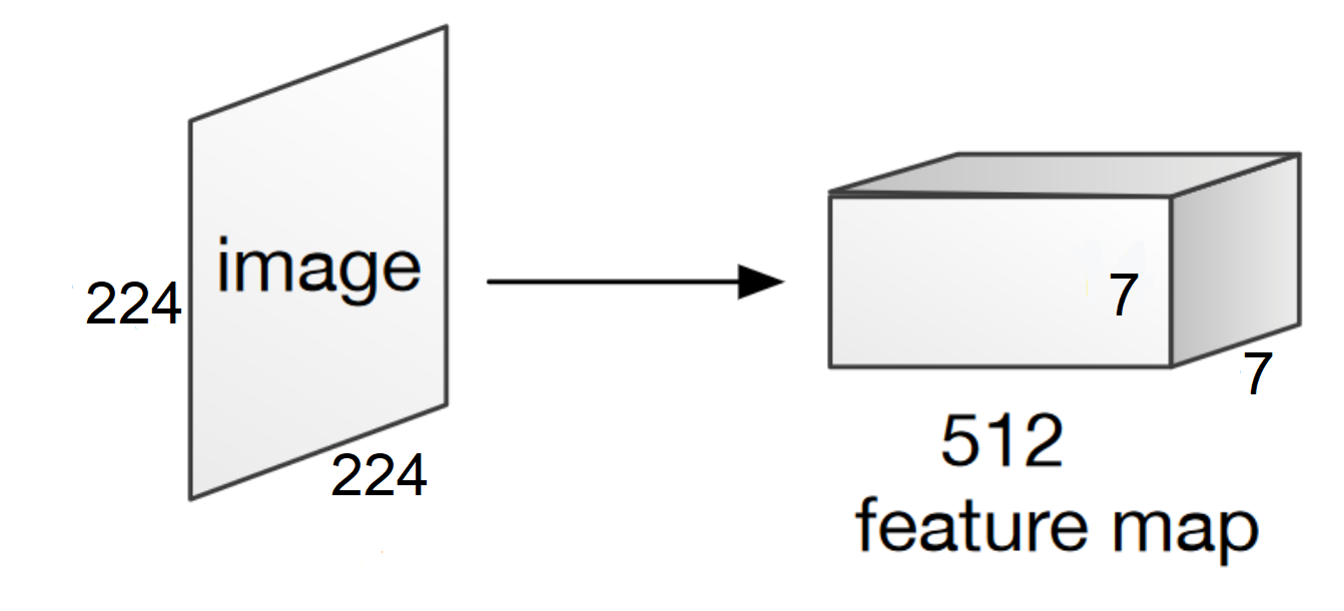
**Figure 4.1.5 - VGG16**

**2) Features Extraction**

- we rescale the input image into **224x224.**

- we extracts the features of the input image from the **last pooling layer.**

- the last pooling layer has dimensions of **7x7x512.**



**Figure 4.1.6 – output of the last pooling layer**

-we consider that the image has **49 regions** as 7x7 dimension of the output 7x7x512.

-each region has feature vector of length 512 as the third dimension of the output 7x7x512.

-so, we have 49 regions each one has feature vector of length 512.

# User Manual

This chapter should describe in details how to operate the project along with screen shots of the project representing all steps.

This chapter should also include an "Installation Guide" that would describe how to install the program, and all required third party tools that needs to be available for the project to run. The installation guide will also be included as a readme file in the CDs delivered at the end of the year.

# Conclusion and Future Work

## 6.1 Conclusion

A complete summary of the whole project along with the results obtained.

## 6.2 Future Work

What can be done in the future to improve the performance of the project and what additional functions could be added?

Add Appendices if you think it’s needed like:

1. Main code segments
2. Any surveys made

# References

[1] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. CoRR, abs/1409.1556, 2014.

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition.In CVPR, 2016.

[3] Mengye Ren, Ryan Kiros, and Richard Zemel. Exploring models and data for image question answering.In NIPS, 2015

[4] Jiasen Lu, Jianwei Yang, Dhruv Batra , and Devi Parikh. Hierarchical Question-Image Co-Attention for Visual Question Answering, 2017