Democratic and Popular Republic of Algeria
Ministry of Higher Education and Scientific Research
University of Kasdi Merbah Ouargla
Faculty of new information and communication technologies
Department of computer science and Information technology

Thesis Submitted to the Department of computer science and Information in Candidacy
for the Degree of Master in computer science, option **“industrial computer science”**.

Topic

A healthcare system using deep learning

Presented by:

BENNOUH Radja

Committee members

Dr.AZZAOUI Nadjet	President
Dr. MERABTI Houssin	Examiner
Dr. AIADI Oussama	Supervisor

Academic Year 2021/2022

ACKNOWLEDGMENTS

First and foremost, praise and thanks to ALLAH the sacred and the mighty for giving me the ability and helping me in accomplishing this thesis.
Then, I would like to express my sincerest gratitude to my supervisor Dr. AIADI Oussama for his continuous support, help, guidance, motivation and patience throughout this thesis.
A very special thanks go to the honorable members of my dissertation committee jury who accepted to examine and evaluate this thesis
I gratefully acknowledge all the professors within the Department of computer science and information technology

ABSTRACT

Classification of images is useful for extracting different useful information, for example, classification of animals (cat, dog...etc), classification of plants, classification of X-ray images to tell the injured from the uninfected, all these things are useful, in this thesis we are interested in developing A system for identifying the type of insect from the stinger using deep learning.

The proposed approach is to try several models of the convolution neural network (CNN) and see the classification ability of each model and then apply the learning set method to improve the system's classification accuracy.

We run experiments on our own dataset created with careful research and effort and the help of an expert in the medical field and guidance to know the features of each bite

Keyword: deep learning, X-ray, dataset, CNN, ensemble learning

الملخص

يعد تصنيف الصور مفيدا لاستخراج معلومات مفيدة مختلفة ،على سبيل المثال ، تصنيف الحيوانات (قط، كلب ....الخ)، تصنيف النباتات، تصنيف صور اشعة صينية لمعرفة المصاب من غير المصاب ، كل هذه الامور مفيدة ، في هذه . الاطروحة نحن مهتمون بتطوير نظام للتعرف على نوع الحشرة من اثر اللذغة باستخدام التعلم العميق

النهج المقترح هو تجربة عدة نماذج من الشبكة العصبية الالتفافية ومعرفة قدرة تصنيف كل نموذج ومن ثم تطبيق طريقة مجموعة التعلم لتحسين دقة النظام في التصنيف

نحن نجري التجارب على مجموعة البيانات الخاصة بنا تم انشاؤها بمجهود وبحث دقيق والاستعانة بخبير في المجال الطبي وتوجيهما لمعرفة ميزات كل لذغة.

كلمات مفتاحيه : تصنيف ، اشعة صينية، الشبكة العصبية الالتفافية ، مجموعة التعلم

TABLE OF CONTENTS

CONTENTS

LISTE OF FIGURES

LISTE OF TABELS

CHAPTER 1: GENERAL INTRODUCTION

1. Introduction

The human brain is a strange and wonderful thing, a complex biological network we are born with that gives us the cognitive fuel to distinguish between horses and cows, cars and trains, happy and sad faces. Unfortunately, the electronic communication network known as a computer is nowhere good at this sort of thing, and so experts around the world have spent decades trying to figure out if it's actually possible to simulate human perception inside an electronic box. Although we have seen amazing advances in the ability to store and process information, artificial intelligence has always been left behind, but in recent years there has been a buzz around a concept that appears to be a game-changer in the end, this concept called deep learning.

The interpretation of what is meant by deep learning depends critically on understanding the term machine learning. This expression is often misinterpreted, and in fact does not refer to the use of machines "for" learning, the self, and the learner. In machine learning, computers are fed massive amounts of data that are analyzed and "learn" through the clever use of algorithms. In other words, a computer is effectively 'trained' on data, which enables it to learn how to perform a particular task without being specifically programmed to do so. Deep learning goes a step further by combining machine learning, neural networks, and computational models of the brain that enable data classification and interpretation more effectively through mathematical techniques such as weighting and probability, and relying on so-called computer vision that focuses on designing computer systems. These systems capture and interpret image and video data, then translate it into insights. The ultimate goal of computer vision is to use image data and to develop ways to reproduce the capabilities of human vision.

Computer vision has traditionally relied on image processing algorithms, where the main process has been to extract image features, by detecting colors, edges, angles and objects as the first step to take when performing a computer vision task. Deep learning techniques appeared in the field of computer vision a few years ago, and have shown a significant increase in performance and accuracy in computer vision applications. With deep learning, feature definition is automatically implemented by deep neural networks and this is what we will rely on in the thesis.

2. Problematic

In this thesis, we look at dealing with the research question related to identifying a special image of a sting effect on human skin to determine the type of insect that bit it.

We live in an environment full of insects, and some of them are poisonous. These insects leave traces on our skin through our sting. We cannot distinguish them except by going to a specialist or a doctor to determine whether the matter is cause for concern or not. So, through this thesis, we can suggest an approach based on the latest findings. Science in artificial intelligence to distinguish between these effects and determine the type of insect from a digital image.

3. Motivation:

The biggest motivation for working on this thesis is the importance of the topic in the environment in which we live because many people are exposed to unknown stings that cause several diseases and sometimes death.

Especially those who live in secluded areas or far from medical centers and desert residents because it is home to many types of poisonous and deadly insects such as scorpions, as well as adventurous people in nature, mountain climbers, or diggers, and farmers because they are more susceptible to unknown bites

Insect bites are one of the main causes of the spread of dangerous diseases and epidemics. Working on this message is a simple addition and help in the medical field, by exploiting the technological development in the field of artificial intelligence to reduce the risks to many exposed to the bites of unknown insects.

4. Contributions:

The contributions of this thesis can be summarized as follows:

- Work on creating a new data set of images, containing 7 types of stings, each category is a set of images of stings for one insect.
- The use of an expert and careful research on the characteristics of each sting in each insect
- To classify pictures adopted method, ensemble learning to get a good result in context

5. Thesis structure:

This thesis is organized as follows:

The first chapter explains the general background to the work, which includes general concepts such as image and digital image, methods of classification, CNN...etc.

A second chapter is devoted to presenting the proposed tools and method of work.

The third chapter presents the experimental results after implementing the proposed model on digital images

Finally, a general conclusion is given with some points of view that may be usefulto work in the future in the same vein.

CHAPTER 2: WORK BACKGROUND

1. Introduction

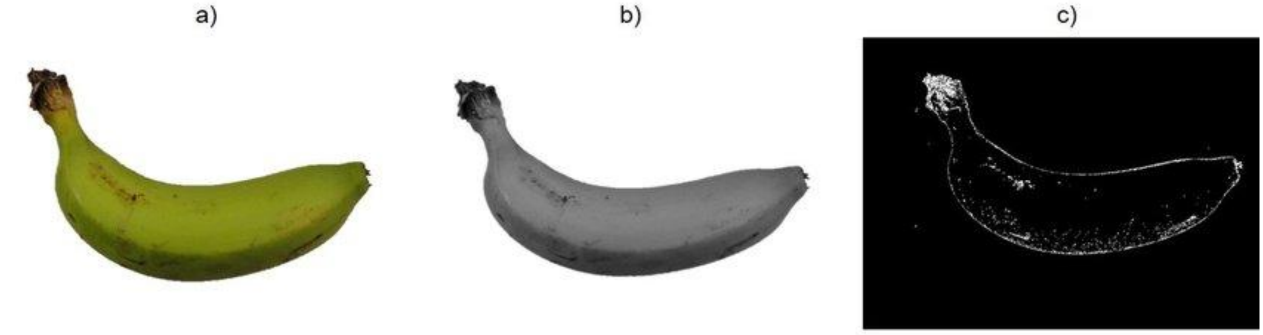
At present, with the remarkable progress in image technologyacquisition devices, a large amount of different images are available applications including personal, medical
and industrial applications. And so it becomes Dealing with such groups is complicated bytheir largeness as an example, Extracting a target image from a huge batch may require spending a large amount
amount of effort and time. Therefore, research scientists have proposed several techniques
It allows you to manage these groups intelligently, accurately and quickly.
Technologies can help us deal with such a problem related to different sub-domains
artificial intelligence (AI), including machine learning, computer vision, and pattern
to get to know. In this chapter, we are interested in presenting our basic background
Work, including details on digital image processing, computer vision applications,
Components of recognition systems, types of classification strategies, and finally
An overview of deep learning.

2. Digital image

By referring to Gonzelez and Woods book **[1]**, we can quote the following definition for the digital image:
An image may be defined as a two-dimensional function, f(x, y), where x and y are spatial (Plane) coordinates, and the amplitude off at any pair of coordinates (x, y) is called the intensity or gray level of the image at that point. When x, y and the
amplitude values of f are all finite, discrete quantities, we call the image a “digital image” from this definition, authors have pointed out that image is analogous to a 2D discrete function, where the pixels values represent the function image and pixel coordinates represent the 2D entries of this function.

3. Digital image processing

Digital image processing aims at manipulating and analyzing digital images automatically by using programmable algorithms on computer. Basic level algorithms include the ones interested with enhancing image quality (e.g., increase image contrast), detecting edges see (**Figure. 1**), removing noise…etc. Another level in image processing is to convert image to a set of numerical values that can further be used for other tasks e.g., image classification. In summary, digital image processing takes an image as input and produce either another image or numerical feature vector.[2]



4. Machine Learning

Machine learning is an area of artificial intelligence (AI) with a concept that a computer program can learn and adapt to new data without human intervention.

Figure 1:Edge detection for the image at left

A complex algorithm or source code is built into a computer that allows for the machine to identify data and build predictions around the data that it identifies, machine learning is useful in parsing the immense amount of information that is consistently and readily available in the world to assist in decision making, Machine learning can be applied in a variety of areas, such as in investing, advertising, lending, organizing news, fraud detection, and more....

Model uses parameters built in the algorithm to form patterns for its decision-making process. When new or additional data becomes available, the algorithm automatically adjusts the parameters to check for a pattern change, if any.[3]

5. Computer vision

Computer vision is the science concerned with obtaining, processing, analysing digital images. Computer vision aims to imitate the human visual system automatically and desirably in an accurate and efficient manners. Computer vision is composed of three levels as follows:

Low-level: this level is the intersection between image processing and vision. Low-level include basic operations to process the input image e.g., detecting Object edges in image**.**

Intermediate-level**:** a representative example for this level is visual object recognition e.g., recognizing different types of visual including cars, animals, persons, digits…etc.**(Figure. 2).**

High-level: this level is more advanced than the former one, as it can provide us With high-level descriptions for images. For instance, associating an image with a complete annotations (statements) that describe activities, events and intention.

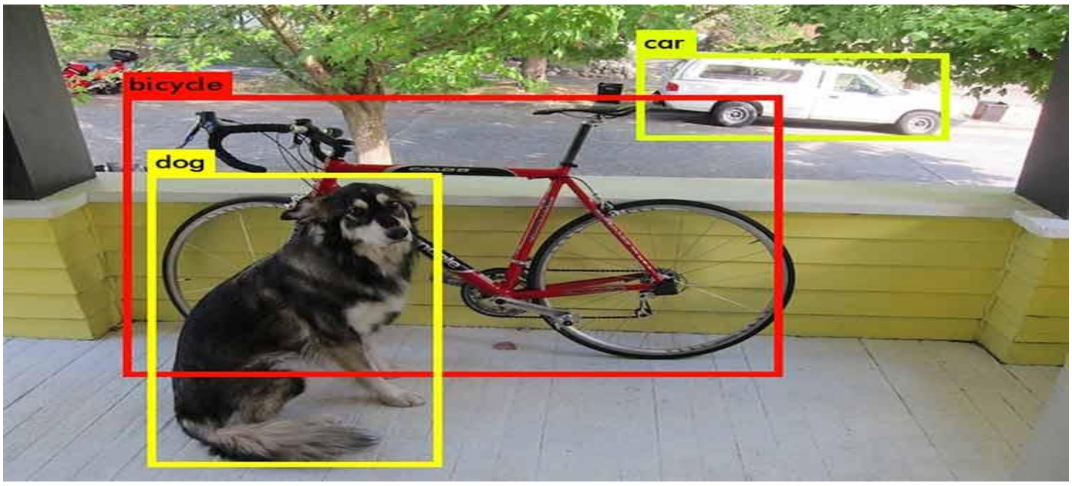


Figure 2 : example of computer vision detection

6. Applications of computer vision:

6.1. Computer Vision for Defect detection

This is, perhaps, the most common application of computer vision. Until now the detection of defects is carried out by trained people in selected batches, and total production control is usually not possible. With computer vision we can detect defects such as cracks in metals, paint defects, bad prints etc. in sizes smaller than 0.05mm. Much better than the human eye! These vision cameras need an algorithm that is the “intelligent brain” which is able to differentiate what is a defect and what is not. This algorithm is designed and trained specifically for each particular application through images with defect and without defect.

6.2. Healthcare

This is a topic with a great importance, as computer vision techniques can be used as decision-assistant when analysing medical images. In addition, multimodal medical image fusion incorporates images of the same scene taken from different views (using different sensors) e.g., X-ray, Computerized Tomography (CT), Magnetic Resonance Imaging (MRI)…etc. For instance, in clinical application, MR-T1 provide us with details about the anatomical structures, whereas MR-T2 displays the difference between normal and abnormal tissues. Fusing these two images may help taking detecting the right diagnosis.

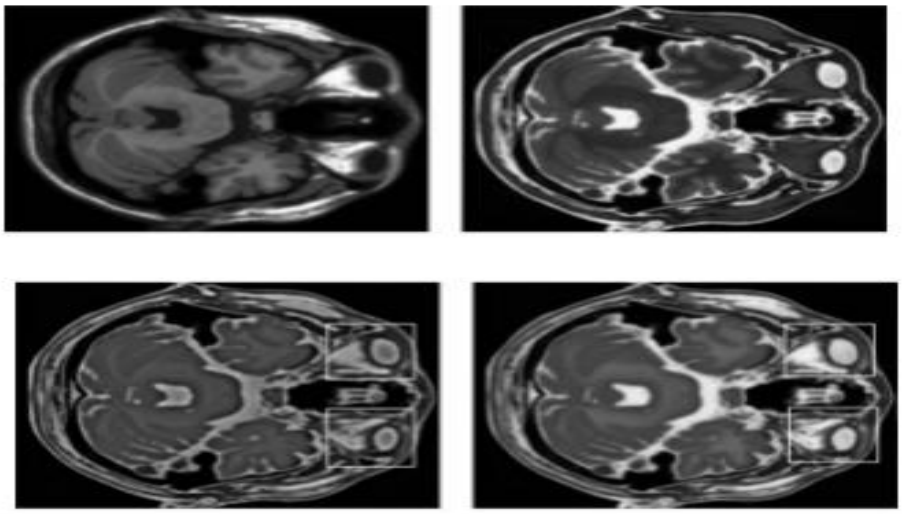


Figure 3 : illustration of multimodal medical image fusion

6.3. Computer Vision for Intruder Detection:

Through hyperspectral cameras it is possible to differentiate between a fruit and a stone, which allows, especially in food, safer products for the consumer. Hyperspectral cameras are able to differentiate the type of material through the measurement they make of the wavelength. In this way, it is possible to differentiate a stone from a fruit, a plastic from a metal or other combinations while the material is different.

6.4. Computer Vision for Assembly verification:

Every day more and more complex assemblies are made, with more parts or connections. Computer vision allows us to verify, step by step, that each piece is in its place, or at the end of the process, that the final assembly is correct. This application is very useful for the assembly of machinery, equipment, electronic boards or pre-assemblies with a lot of complexity. These systems significantly reduce cycle times of very complex operations and reoperation times

6.5. Computer Vision for Screen reader:

Sometimes it is not possible to extract data from a display screen either because it is a closed supplier system or because that system is incompatible with the one installed. A solution to this problem is to install a computer vision camera to read the screen and extract the data that appears on it (temperatures, codes, tensions… any useful information that appears on the screen and you need it). To do this, we look for the interest regions in which the information is located, we use a character recognition algorithm (OCR) to extract it, and everything perfect!

6.6. Computer Vision + robotics for bin picking:

And finally one of the applications that we are most requested, combining collaborative robotics with computer vision to be able to perform bin picking of pieces placed disorderly. The pieces are messy and therefore we need to optimize the trajectories and detect the grip coordinates. The robot needs help (the computer vision) telling him what a piece is and where it is, so it can decide what is the best way to pick up the piece**. (Figure.4)**

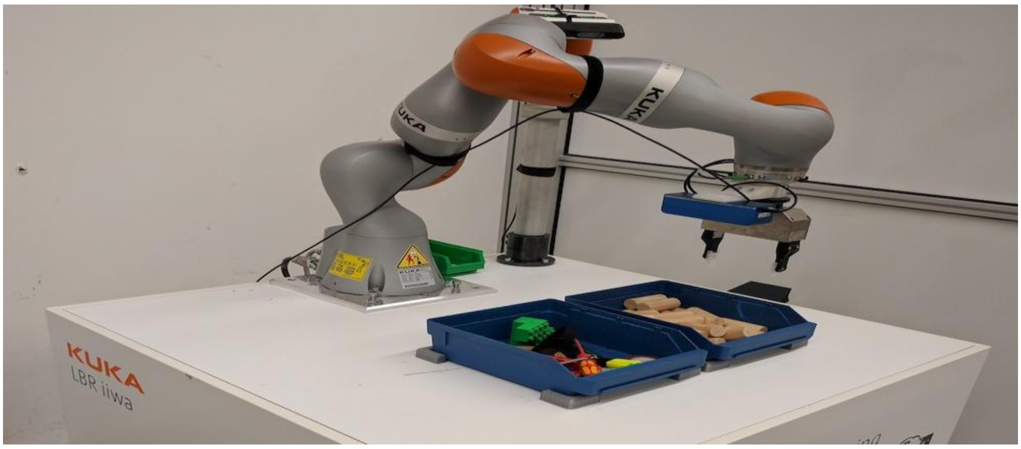


Figure 4: robotics for bin picking

A non-exhaustive list of other of computer vision applications includes also

- Image restoration
- Robots navigation
- Image retrieval and annotation
- Action recognition
- Manufacturing
- Agriculture services

..….etc.

Computer vision needs lots of data. It runs analyses of data over and over until it discerns distinctions and ultimately recognize images. For example, to train a computer to recognize automobile tires, it needs to be fed vast quantities of tire images and tire-related items to learn the differences and recognize a tire, especially one with no defect

Two essential technologies are used to accomplish this: a type of machine learning called deep learning and a convolutional neural network (CNN).

Machine learning uses algorithmic models that enable a computer to teach itself about the context of visual data. If enough data is fed through the model, the computer will “look” at the data and teach itself to tell one image from another. Algorithms enable the machine to learn by itself, rather than someone programming it to recognize an image.

7. Deep learning

Deep learning is the new state-of-the-art for artificial intelligence. Deep learning architecture is composed of an input layer, hidden layers, and an output layer. The word deep means there are more than two fully connected layers. There is a vast amount of neural network, where each architecture is designed to perform a given task. For instance, CNN works very well with pictures, RNN (Recurrent neural networks) providesimpressive results with time series and text analysis. Deep learning is now active in different fields, from finance to marketing, supply chain, and marketing. Big firms are the first one to use deep learning because they have already a large pool of data. [4]

8. Artificial Neural Network (ANN)

ANN architecture is based on the structure and function of the biological neural network. Similar to neurons in the brain, ANN also consists of neurons which are arranged in various layers. Feed forward neural network is a popular neural network which consists of an input layer to receive the external data to perform pattern recognition, an output layer which gives the problem solution, and a hidden layer is an intermediate layer which separates the other layers. The adjacent neurons from the input layer to output layer are connected through acyclic arcs. The ANN uses a training algorithm to learn the datasets which modifies the neuron weights depending on the error rate between target and actual output. In general, ANN uses the back propagation algorithm as a training algorithm to learn the datasets. The general structure of ANN is shown in (**Figure.5**).[5]

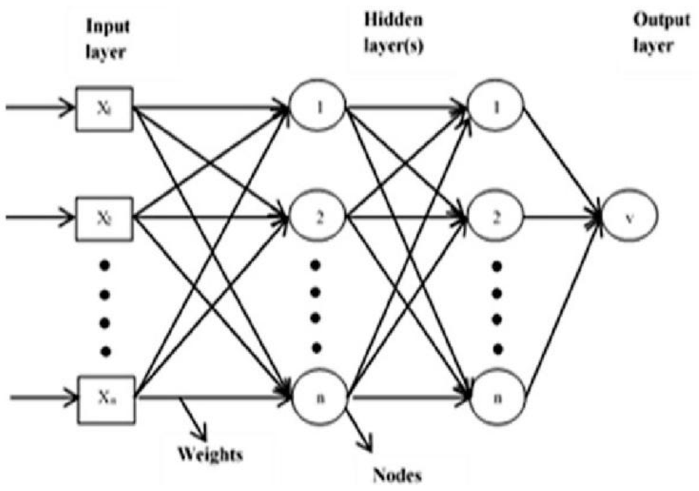


Figure 5 : The general structure of ANN

9. Convolutional neural network

Neural networks are a subset of machine learning, and they are at the heart of deep learning algorithms. They are comprised of node layers, containing an input layer, one or more hidden layers, and an output layer. Each node connects to another and has an associated weight and threshold. If the output of any individual node is above the specified threshold value, that node is activated, sending data to the next layer of the network. Otherwise, no data is passed along to the next layer of the network.

There are various types of neural nets, which are used for different use cases and data types. For example, recurrent neural networks are commonly used for natural language processing and speech recognition whereas convolutional neural networks (ConvNets or CNNs) are more often utilized for classification and computer vision tasks. Prior to CNNs, manual, time-consuming feature extraction methods were used to identify objects in images **(Figure.6)**. However, convolutional neural networks now provide a more scalable approach to image classification and object recognition tasks, leveraging principles from linear algebra, specifically matrix multiplication, to identify patterns within an image. That said, they can be computationally demanding, requiring graphical processing units (GPUs) to train models.

Considered as the actual seed from which the CNN is developed. From 2010 to 2017, ImageNet Large Scale Visual Recognition Challenge (ILSVRC) is started, which aims to achieve to best classification performance on ImageNet which is made up of more than 14 million images from different classes. Since then, multiple CNN architectures were designed to participate in the challenge involving GoogleNet, ResNet, AlexNet and VGG16…etc.[6]

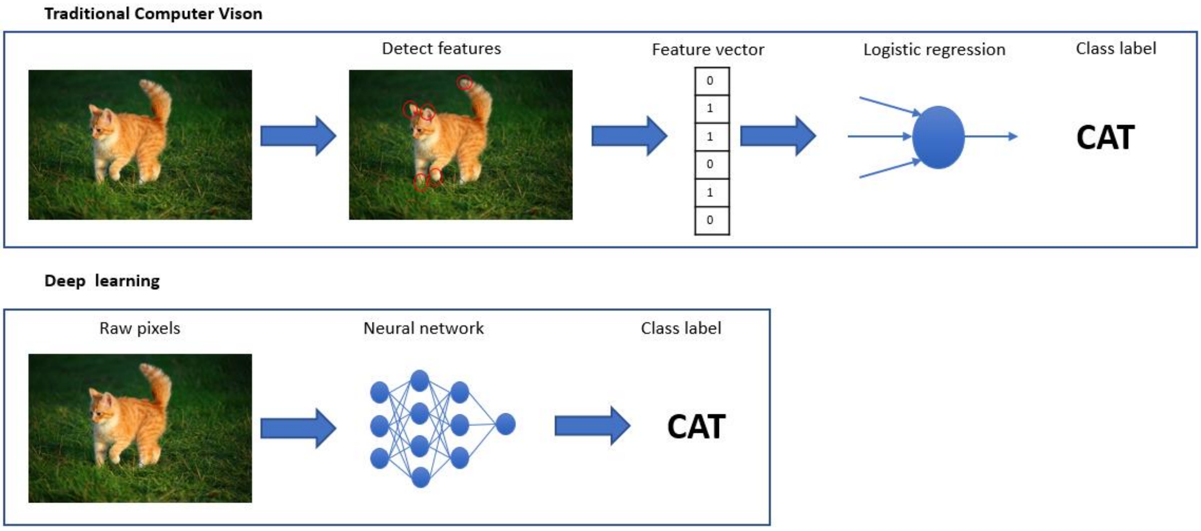


Figure 6 : computer vision before and after CNN

9.1. Applications of CNN:

CNN has been widely applied for different recognition jobs, hereafter, we outline some of them

- Biometrics: e.g., face, ear, iris recognition.
- Visual object recognition and tracking.
- Visual Object detection: e.g., YOLO framework.
- Remote sensing: e.g., for change detection.
- Self-driving cars: which can avoid collision based on deep net predications.
- Image retrieval and annotation.
- Healthcare: e.g., help experts reaching correct diagnosis.
- Speech recognition.
- …etc.

In spite of its impressive performance, CNN still have multiple cons. Mainly, CNN suffer from two major shortcomings

- It may take a great deal of time to train (i.e., slow training)
- It is data hungry because it requires a lot of labeled training data to perform training.

9.2. Convolutional neural networks in details:

Convolutional neural networks are distinguished from other neural networks by their superior performance with image, speech, or audio signal inputs. They have three main types of layers, which are:

- Convolutional layer
- Pooling layer
- Fully-connected (FC) layer

The convolutional layer is the first layer of a convolutional network. While convolutional layers can be followed by additional convolutional layers or pooling layers, the fully-connected layer is the final layer. With each layer, the CNN increases in its complexity, identifying greater portions of the image. Earlier layers focus on simple features, such as colors and edges. As the image data progresses through the layers of the CNN, it starts to recognize larger elements or shapes of the object until it finally identifies the intended object.

9.2.1. Convolutional Layer:

The convolutional layer is the core building block of a CNN, and it is where the majority of computation occurs. It requires a few components, which are input data, a filter, and a feature map. Let’s assume that the input will be a color image, which is made up of a matrix of pixels in 3D. This means that the input will have three dimensions—a height, width, and depth—which correspond to RGB in an image. We also have a feature detector, also known as a kernel or a filter, which will move across the receptive fields of the image, checking if the feature is present. This process is known as a convolution.

The feature detector is a two-dimensional (2-D) array of weights, which represents part of the image. While they can vary in size, the filter size is typically a 3x3 matrix; this also determines the size of the receptive field. The filter is then applied to an area of the image, and a dot product is calculated between the input pixels and the filter. This dot product is then fed into an output array. Afterwards, the filter shifts by a stride, repeating the process until the kernel has swept across the entire image. The final output from the series of dot products from the input and the filter is known as a feature map, activation map, or a convolved feature**.**

As you can see in the image **(Figure.7**), each output value in the feature map does not have to connect to each pixel value in the input image. It only needs to connect to the receptive field, where the filter is being applied. Since the output array does not need to map directly to each input value, convolutional (and pooling) layers are commonly referred to as “partially connected” layers. However, this characteristic can also be described as local connectivity.

Note that the weights in the feature detector remain fixed as it moves across the image, which is also known as parameter sharing. Some parameters, like the weight values, adjust during training through the process of backpropagation and gradient descent. However, there are three hyperparameters which affect the volume size of the output that need to be set before the training of the neural network begins. These include:

1. The number of filters affects the depth of the output. For example, three distinct filters would yield three different feature maps, creating a depth of three.

2. Stride is the distance, or number of pixels, that the kernel moves over the input matrix. While stride values of two or greater is rare, a larger stride yields a smaller output.

3. Zero-padding is usually used when the filters do not fit the input image. This sets all elements that fall outside of the input matrix to zero, producing a larger or equally sized output. There are three types of padding:

Valid padding: This is also known as no padding. In this case, the last convolution is dropped if dimensions do not align.

Same padding: This padding ensures that the output layer has the same size as the input layer
Full padding: This type of padding increases the size of the output by adding zeros to the border of the input.

After each convolution operation, a CNN applies a Rectified Linear Unit (ReLU) transformation to the feature map, introducing nonlinearity to the model.

As we mentioned earlier, another convolution layer can follow the initial convolution layer. When this happens, the structure of the CNN can become hierarchical as the later layers can see the pixels within the receptive fields of prior layers. As an example, let’s assume that we’re trying to determine if an image contains a bicycle. You can think of the bicycle as a sum of parts. It is comprised of a frame, handlebars, wheels, pedals, et cetera. Each individual part of the bicycle makes up a lower-level pattern in the neural net, and the combination of its parts represents a higher-level pattern, creating a feature hierarchy within the CNN.

Finally, the convolutional layer converts the image into numerical values, allowing the neural network to interpret and extract related patterns.

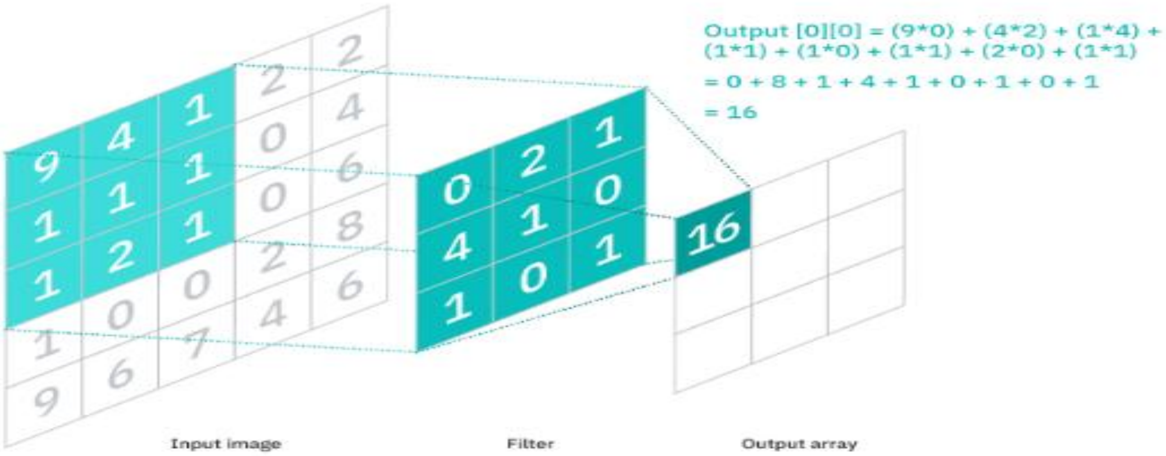


Figure 7: The feature detector

9.2.2. Pooling Layer:

Pooling layers, also known as down sampling, conducts dimensionality reduction, reducing the number of parameters in the input. Similar to the convolutional layer, the pooling operation sweeps a filter across the entire input, but the difference is that this filter does not have any weights. Instead, the kernel applies an aggregation function to the values within the receptive field, populating the output array. There are two main types of pooling:

Max pooling: As the filter moves across the input, it selects the pixel with the maximum value to send to the output array. As an aside, this approach tends to be used more often compared to average pooling.

Average pooling: As the filter moves across the input, it calculates the average value within the receptive field to send to the output array.

While a lot of information is lost in the pooling layer, it also has a number of benefits to the CNN. They help to reduce complexity, improve efficiency, and limit risk of overfitting.

9.2.3. Fully-Connected Layer:

The name of the full-connected layer aptly describes itself. As mentioned earlier, the pixel values of the input image are not directly connected to the output layer in partially connected layers. However, in the fully-connected layer, each node in the output layer connects directly to a node in the previous layer.(fig.8).

This layer performs the task of classification based on the features extracted through the previous layers and their different filters. While convolutional and pooling layers tend to use ReLu functions, FC layers usually leverage a softmax activation function to classify inputs appropriately, producing a probability from 0 to 1.

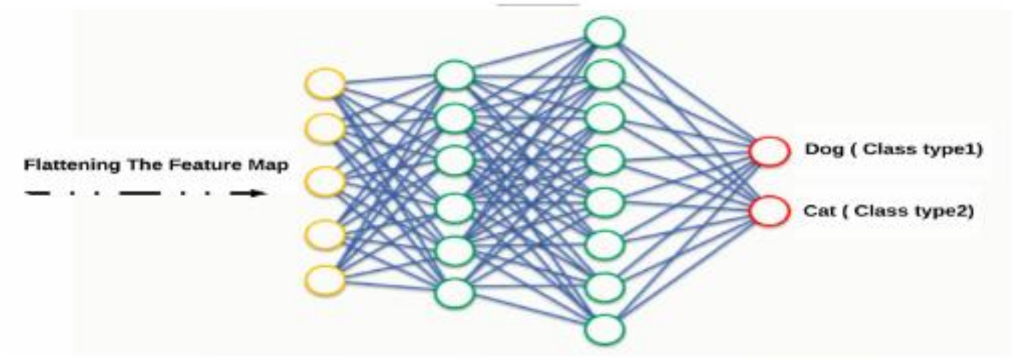


Figure 8: The architecture of Fully Connected Layers

ReLU

[7]The Rectifier Linear Unit (ReLU)is the most commonly used activation function inConvolutional Neural Networks. It is used to convert all the input values to positive numbers.

The advantage of ReLU is that it requires very minimal computation load compared to others. The mathematical representation of ReLU is:

$$f(x)\text{ReLU} = \max(0, x)$$

9.3. The major CNN models for image classification

- LeNet-5:

The LeNet-5[21] is one of the earliest CNN architecture, which was designed for classifying thehandwritten digits. It was introduced by LeCun et al. in 1998. The LeNet-5 has 5 weighted (trainable) layers, that is, three convolutional layer and two FC layers. Among them, each offirst two convolution layer is followed by a max-pooling layer (to sub-sample the feature maps) and afterward, the last convolution layer is followed by two fully connected layers. The lastlayer of those fully connected layers is used as the classifier, which can classify 10 digits.

- AlexNet:

Inspired from **LeNet**, **Krizhevky** et al. designed first large-scale CNN model, called AlexNet in 2012, which is designed to classify ImageNet data. It consists of eight weighted (learnable) layers among which the first five layers are convolutional layers, and afterward, the last threelayers are fully connected layers. Since it was designed for ImageNet data, so the last outputlayer classify the input images into one of the thousand classes of the ImageNet dataset withthe help of 1,000 units.

- ZFNet:

ZFNetwas presented by Zeiler and Fergus in ECCV-2014, it has almost similar architectureas AlexNet except that here they used 7×7 filter with stride 2 in 1’st convolutional layer. In case of AlexNet, Krizhevky et al. use 11×11 filter with stride 4 in 1’st convolutional layer. As a result ZFNet becomes more efficient than AlexNet and become the winner of ILSVRC-2013

- VGGNet:

VGGNetis one of the most popular CNN architecture, which is introduced by Simonyan andZisserman in 2014. The authors introduced a total 6 diff erent CNN configurations, among them, the**VGGNet-16** (configuration D) and **VGGNet-19** (configuration E) are the most successfulones.

- GoogLeNet:

The GoogleNet architecture is different from all the previously discussed conventional CNN models, it uses network branches instead of using single line sequential architecture. The GoogleNet was proposed by Szegedy et al. in 2014. The GoogleNet has 22 weighted (learn-able) layers, it used “Inception Module” as the basic building block of the network. The processing of this module happens in parallel in the network, and each (a simple basic) module consist of 1 ×1, 3 ×3 and 5 ×5 filtered convolution layers in parallel and then it combines their output feature maps, that can resulted in very high-dimensional feature output. To solve this issue they used inception module with dimensionality reduction in their network architecture instead of the naive (basic) version of inception module.

- ResNet:

Since a deep CNN model suffers from vanishing gradient problems as we discussed earlier, He et al. from Microsoft, introduced the idea of “identity skip connection” to solve vanishing gradient problem by proposing the ResNet model. The ResNet’s architecture use residual mapping (H(x) = F(x) + x) instead of learning a direct mapping (H(x) = F(x)) and these blocks are called residual bocks. The complete ResNet architecture is consist of many residual bocks with3×3 convolution layers.

- DenseNet:

DenseNet extends the idea of residual mapping by propagating the output of each blockto all subsequent blocks inside each dense block in the network. By propagating the information in both forward and backward directions during the training ofthe model it strengthens feature propagation ability and solve the vanishing gradient problem. DenseNet was introduced by Huang et al. in 2016 and it becomes the winner of ILSVRC-2016.[8]

9.4. Convolutional neural networks and computer vision:

Convolutional neural networks power image recognition and computer vision tasks. Computer vision is a field of artificial intelligence (AI) that enables computers and systems to derive meaningful information from digital images, videos and other visual inputs, and based on those inputs, it can take action. This ability to provide recommendations distinguishes it from image recognition tasks. Some common applications of this computer vision today can be seen in:

- Marketing:Social media platforms provide suggestions on who might be in photograph that has been posted on a profile, making it easier to tag friends in photo albums.
- Healthcare:Computer vision has been incorporated into radiology technology, enabling doctors to better identify cancerous tumors in healthy anatomy.
- Retail:Visual search has been incorporated into some e-commerce platforms, allowing brands to recommend items that would complement an existing wardrobe.
- Automotive: While the age of driverless cars hasn't quite emerged, the underlying technology has started to make its way into automobiles, improving driver and passenger safety through features like lane line detection

10. Conclusion:

In this chapter, we have an overview of some important concepts, starting with the digital image, Processing and computer vision, passing through recognition and classification systems

Strategies, finalizing a convolutional neural network. Our goal was to presentsome details about the background of the work, making it easier to understand the work and its context with clarity

CHAPTER 3: MATERIAL AND METHOD

1. Introduction

CNN assists machine learning or a deep learning model by dividing images into pixels that are given tags or labels. It uses labels to perform convolutions (a calculation on two functions to produce a third function) and makes predictions about what it sees. The neural network runs convolutions and checks the accuracy of its predictions in a series of iterations until the predictions begin to check. The images are then recognized or seen in a similar way to humans.

Much like a human making a remote image, CNN first distinguishes hard edges and simple shapes, and then fills in the information as it runs its prediction iterations. CNN is used to understand individual images. In this chapter we will show the working tools and CNN model we used to train and test our dataset.

2. Dataset:

In this work using my dataset, which is a collection of **191** images in 7 different categories. Each image is stored in a folder with a name corresponding to its category after validation of each image according to expert advice and research in the features of the shape of each bite, **126**images are used for training, and **65** images are used for testing / validation. Below I'll explain how to collect and validate all of these images in each category:

- **First** we talk about 7 categories of bites and features in each category

1. Ant bite

Its shape is characterized by swelling, redness and swelling, and the white color appears above the swelling of the skin

As in the pictures below.

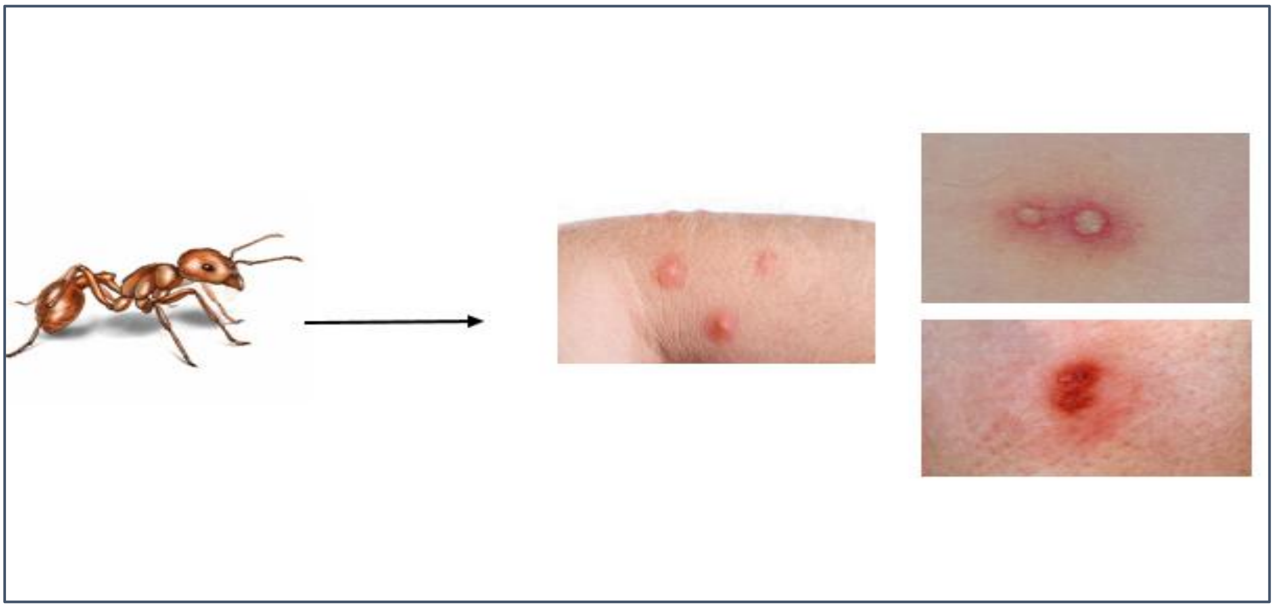
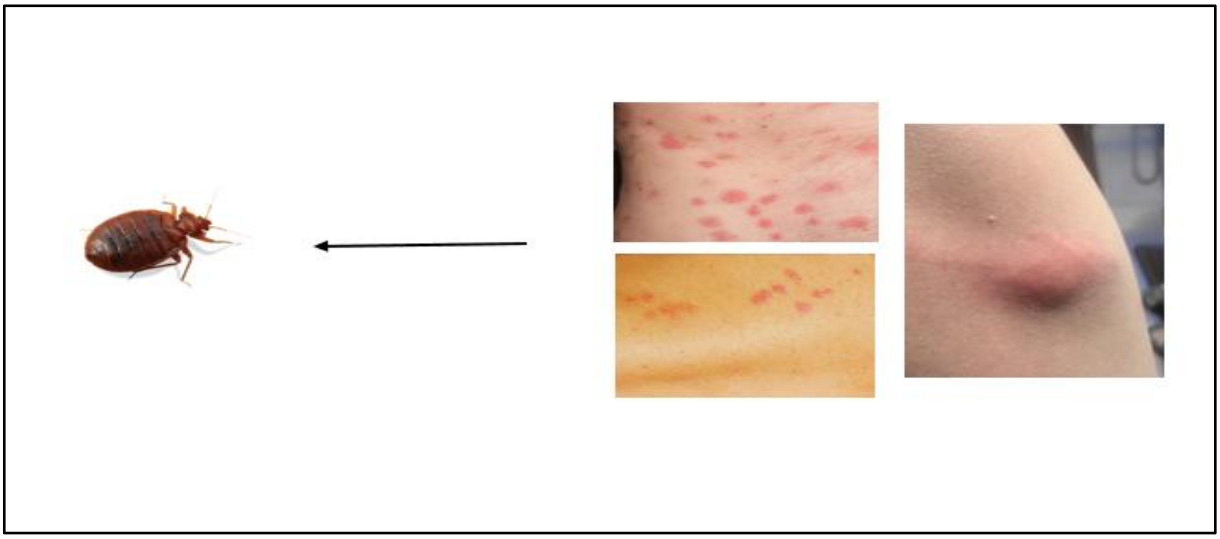


Figure 9 : Samples of ant bites

2. Bed bug bite

Almost all bed bug bites will produce some degree of discomfort, typically itchiness and inflammation. Other signs of bed bug bites include:

- A red itchy bump with a dark center and lighter swollen surrounding area.
- Small red bumps or welts in a zigzag pattern or a line.
- Small red bumps surrounded by blisters or hives.
- Popular eruptions or areas of skin with raised or flat patches that may be inflamed.



3. Bee bite

Often the sign of a bee sting is as follows:

- A red spot in the sting area
- Swelling around the sting area

As you can see in the figure below

Figure 10 : Samples of bed bug bites

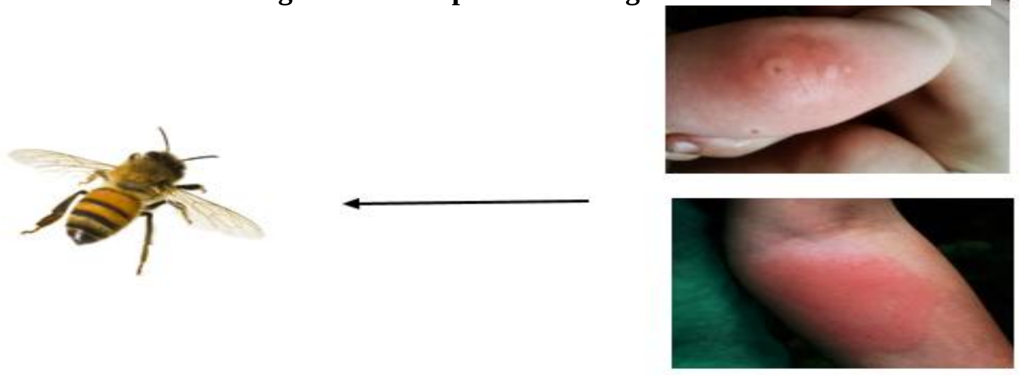


Figure 11 : Samples of bee bites

4. Mosquito bite

Mosquito bite signs include:

- A puffy, white and reddish bump that appears a few minutes after the bite
- A hard, itchy, reddish brown bump, or multiple bumps, appearing a day or so after the bite or bites.
- Small blisters instead of hard bumps
- Dark spots that look like bruises

- A large area of swelling and redness

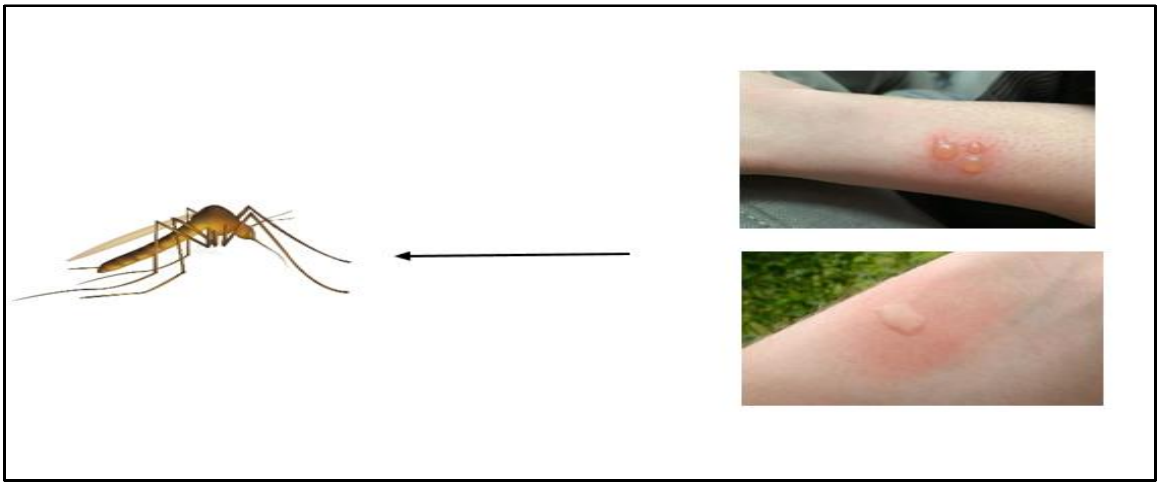


Figure 12 : Samples of mosquito bites

5. Scorpion sting

The majority of scorpion stings cause local symptoms only, such as warmth and pain at the site of the sting. Symptoms can be very severe, signs at the site of the sting can include:

- Swelling around the bite
- Redness and swelling around the sting, similar to the effect of a simple burn on the skin

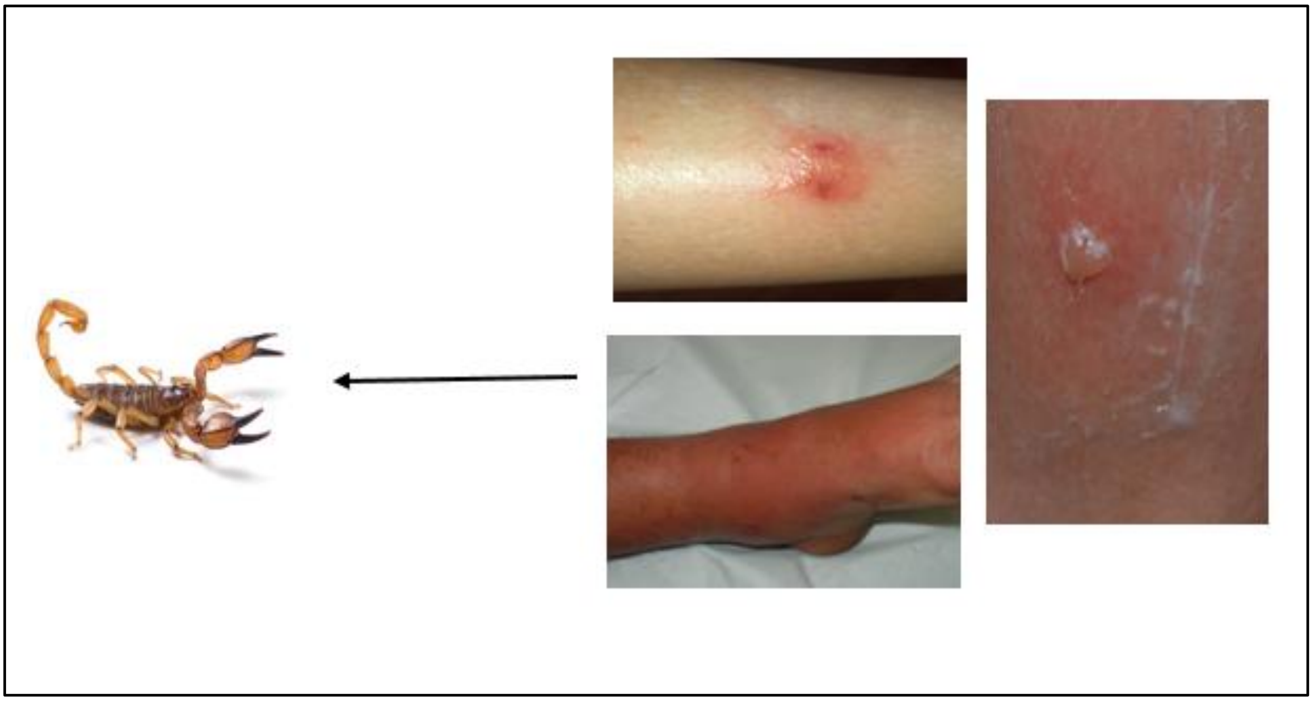


Figure 13 : Samples of scorpion bites

-

6. Spider bite

Symptoms of a spider bite vary depending on the type of spider, and they cause an immediate, sharp pain that resembles a pin prick. Then the sting area becomes numb. General spider bite signs include:

- Dangling or puffy eyes.
- Rash and itching.
- Blister surrounded by a bruise or reddish skin color (similar to a bull's eye). The bulla may rupture and form a skin ulcer that later scars.
- Skin redness and swelling.
- Blisters that produce pus.

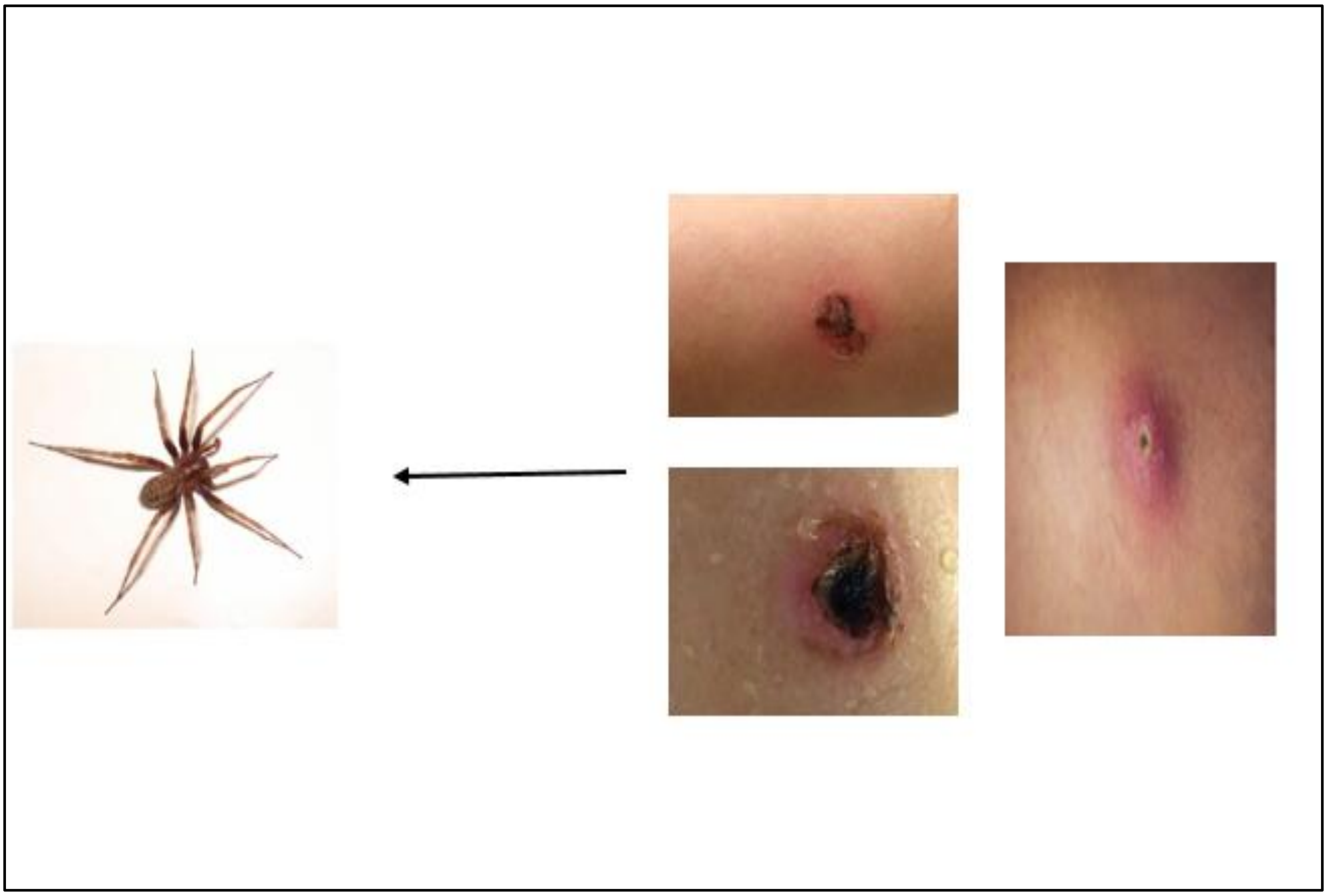


Figure 14 : Samples of spider bites

7. Tick bite

The marks that ticks leave on the skin include:

- A red or discolored spot or rash near the bite site
- All over body rash
- Swelling
- Pain or swelling at the bite site
- A rash
- A burning sensation at the bite site
- Blisters

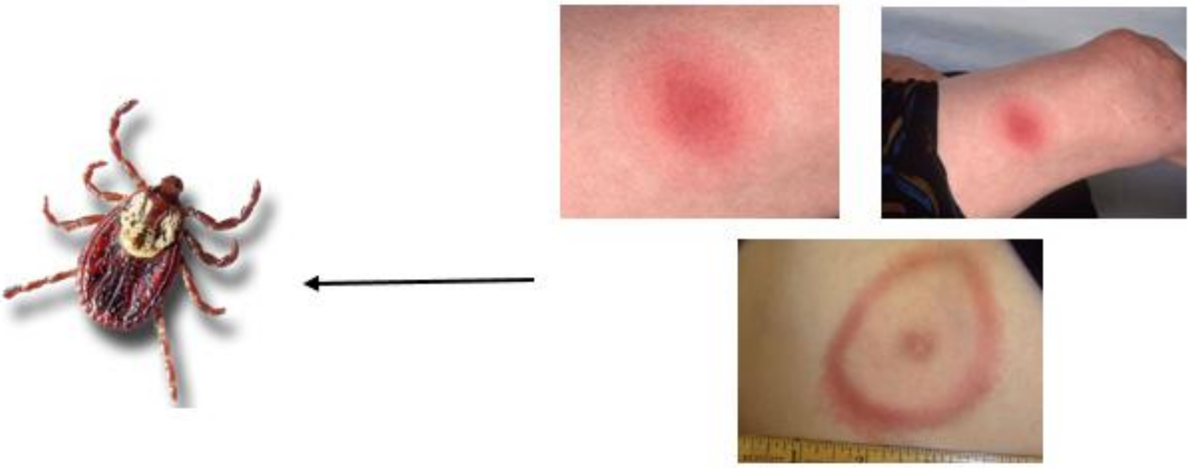


Figure 15 : Samples of tick bites

- ➢ **Secondly**, when you collect the pictures, number them (1, 2, 3 ...... etc.) for each picture in its own category folder (ant bites, bed bug bites …..etc.) and so on with all the pictures in all the categories.

- After collecting the images, we divide them into two folders (train, test) in each folder, the same folders with the name of the categories as in the following table:

Tableau 1 : Count the images in the dataset

Category / folder	train	test
Ant bites	7	5
Bed bug bites	33	15
Bee bites	15	9
Mosquito bites	21	10
Scorpion sting	6	7
Spider bites	23	9
Tick bites	21	10

3. Proposed method:

Initially, I divided my dataset, RGB images from 7 classes of insect bites into two folders, a training folder and a test folder. I can add one for validation but the dataset is small.

I gave it up for good results and to take advantage of the validation data I mentioned earlier

And you split the data in an automatic way using split-folder

After segmenting the data set and processing

- **First**, I used the data augmentation technique
- **Second,** ensemblelearning technique and then started building models

I used pre-trained models and only changed the last layer to re-train my new dataset

The models used are:

- ResNet50
- VGG16
- InceptionV3

Implementing my code and using CNN models these models are responsible for classifying images after that extract the results of each model.,

❖ ResNet50:

I will be classifying images using the ResNet50 deep learning model in this tutorial. I'm using my own dataset to train and test the model, the code is written in Python. ResNet50 is a residual deep learning neural network model with 50 layers. ResNet was the 2015 ImageNet Competition (ILSVRC) winning model and is a popular image classification model, often also used as a backbone model for object detection in an image.

The neural network includes weights, a logging function, and a loss function. The neural network learns in the feedback loop, and adjusts its weights based on the results from the outcome function and the loss function. A simple neural network includes three layers, an input layer, a hidden layer, and an output layer. Deep learning neural network models have more than 3 layers and often use convolutional layers to classify images.

Deep neural network models are difficult to train because of the vanishing gradient problem, repeated multiplication makes the gradient too small and the model stops learning. ResNet solves the gradient fading problem with a skip connection that allows input to flow in a shortcut to the activation layer.

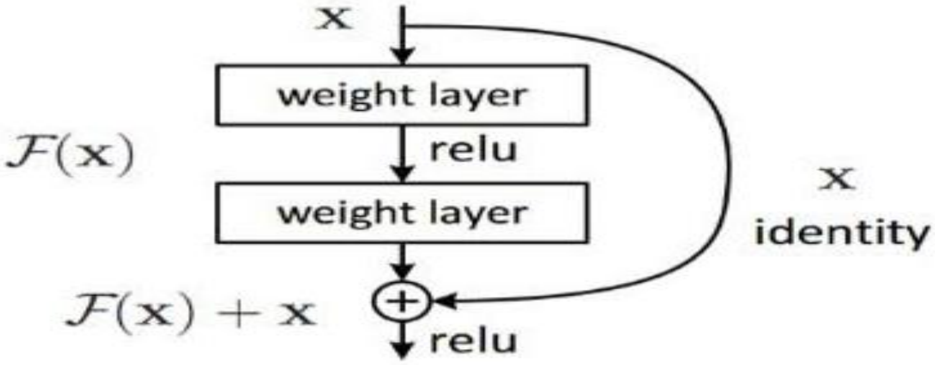


Figure 16 : ResNet50 architecture

ResNet-50 Architecture:

- A convolution with a kernel size of 7 * 7 and 64 different kernels all with a stride of size 2 giving us **1 layer**.
- Next we see max pooling with also a stride size of 2.
- In the next convolution there is a 1 * 1,64 kernel following this a 3 * 3,64 kernel and at last a 1 * 1,256 kernel, These three layers are repeated in total 3 time so giving us **9 layers** in this step.
- Next we see kernel of 1 * 1,128 after that a kernel of 3 * 3,128 and at last a kernel of 1 * 1,512 this step was repeated 4 time so giving us **12 layers** in this step.
- After that there is a kernal of 1 * 1,256 and two more kernels with 3 * 3,256 and 1 * 1, 1024 and this is repeated 6 time giving us a total of **18 layers**.
- And then again a 1 * 1,512 kernel with two more of 3 * 3,512 and 1 * 1, 2048 and this was repeated 3 times giving us a total of **9 layers**.
- After that we do a average pool and end it with a fully connected layer containing 1000 nodes and at the end a softmax function so this gives us **1 layer**.

We don't actually count the activation functions and the max/ average pooling layers.
so totaling this it gives us a 1 + 9 + 12 + 18 + 9 + 1 = 50 layers Deep Convolutional network

❖ VGG16

The VGG-16 model is a convolutional neural network model proposed by **K. Simonyan** and A. Zisserman of the University of Oxford in an article entitled "Very Deep Convolutional Networks for Large-Scale Image Recognition". It was one of the popular models submitted to ILSVRC-2014. It improves on AlexNet by replacing the large kernel-size filtering (11 and 5 in the 1st and 2nd convolutional layer, respectively) with several 3×3 kernel-sized liters one by one. The VGG16 had been training for weeks and was using a black NVIDIA Titan GPU. Shown below is VGG16.[16]

VGG16 Architecture:

The input to the network is image of dimensions (224, 224, 3). The first two layers have 64 channels of 3*3 filter size and same padding. Then after a max pool layer of stride (2, 2), two layers which have convolution layers of 256 filter size and filter size (3, 3). This followed by a max pooling layer of stride (2, 2) which is same as previous layer. Then there are 2 convolution layers of filter size (3, 3) and 256 filter. After that there are 2 sets of 3 convolution layer and a max pool layer. Each have 512 filters of (3, 3) size with same padding. This image is then passed to the stack of two convolution layers. In these convolution and max pooling layers, the filters we use is of the size 3*3 instead of 11*11 in AlexNet and 7*7 in ZF-Net. In some of the layers, it also uses 1*1 pixel which is used to manipulate the number of input channels. There is a padding of 1-pixel (same padding) done after each convolution layer to prevent the spatial feature of the image.

After the stack of convolution and max-pooling layer, we got a (7, 7, 512) feature map. We flatten this output to make it a (1, 25088) feature vector.After this there are 3 fully connected layer, the first layer takes input from the last feature vector and outputs a (1, 4096) vector, second layer also outputs a vector of size (1, 4096) but the third layer output a 1000 channels for 1000 classes of ILSVRC challenge, then after the output of 3rd fully connected layer is passed to softmax layer in order to normalize the classification vector. After the output of classification vector top-5 categories for evaluation. All the hidden layers use ReLU as its activation function. ReLU is more computationally efficient because it results in faster learning and it also decreases the likelihood of vanishing gradient problem.

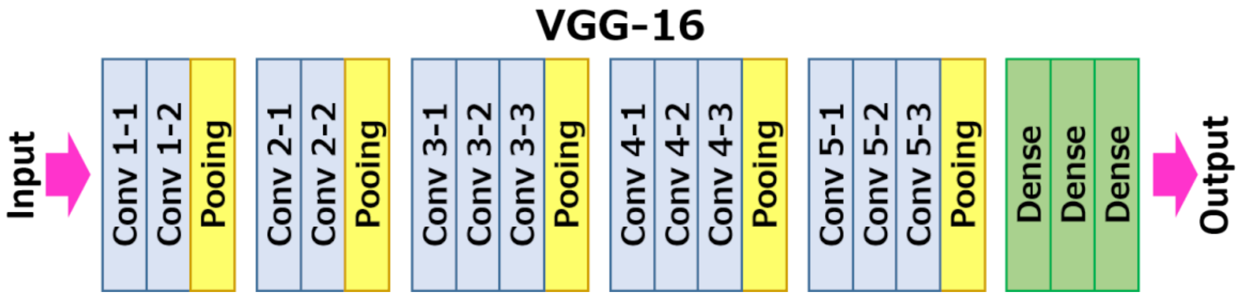


Figure 17:VGG16 Architecture

❖ InceptionV3:

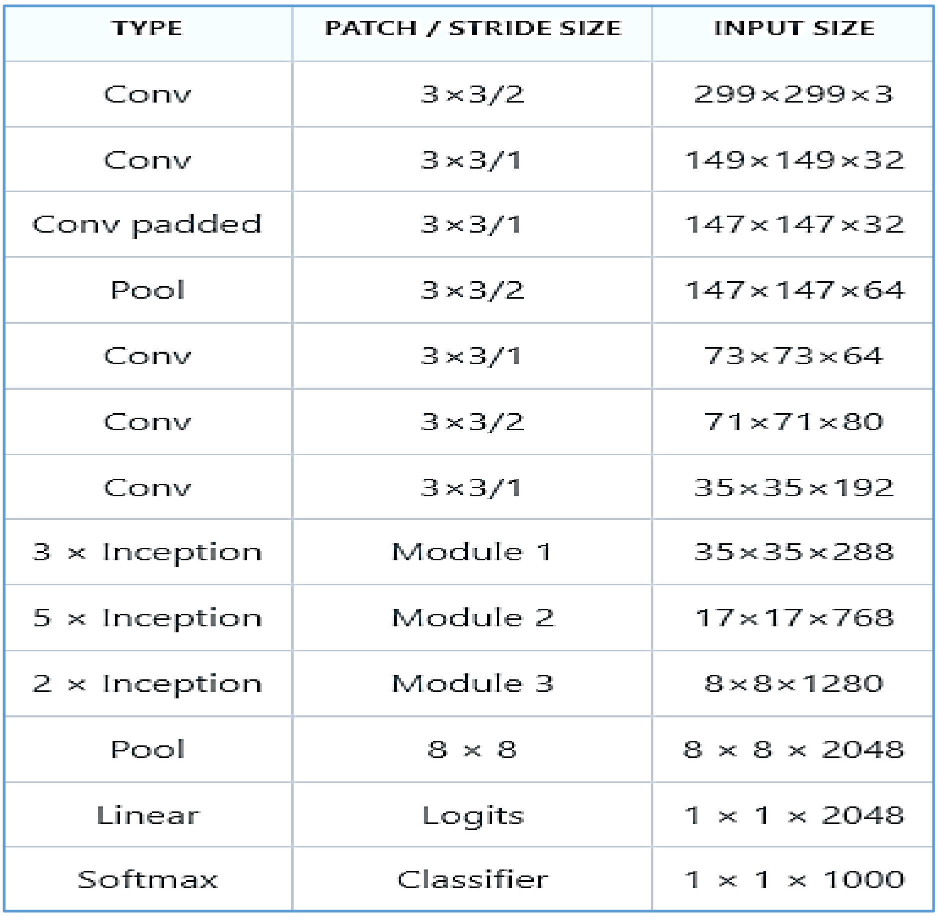
The inception V3 is just the advanced and optimized version of the inception V1 model. The Inception V3 model used several techniques for optimizing the network for better model adaptation.

- It has higher efficiency
- It has a deeper network compared to the Inception V1 and V2 models, but its speed isn't compromised.
- It is computationally less expensive.
- It uses auxiliary Classifiers as regularizes

InceptionV3 Architecture:

The inception V3 model is made up of 42 layers which is a bit higher than the previous inception V1 and V2 models. But the efficiency of this model is really impressive. We will get to it in a bit, but before it let's just see in detail what are the components the Inception V3 model is made of .

Tableau 2 : InceptionV3 layers



TYPE	PATCH / STRIDE SIZE	INPUT SIZE
Conv	3×3/2	299×299×3
Conv	3×3/1	149×149×32
Conv padded	3×3/1	147×147×32
Pool	3×3/2	147×147×64
Conv	3×3/1	73×73×64
Conv	3×3/2	71×71×80
Conv	3×3/1	35×35×192
3 × Inception	Module 1	35×35×288
5 × Inception	Module 2	17×17×768
2 × Inception	Module 3	8×8×1280
Pool	8 × 8	8 × 8 × 2048
Linear	Logits	1 × 1 × 2048
Softmax	Classifier	1 × 1 × 1000

4. Ensemble learning

Ensemble learning is an approach in which two or more models are fitted to the same data, and the predictions of each model are combined. Ensemble learning aims to achieve better performance with the ensemble of models than with any individual model. This requires deciding how to create the models used in the ensemble and how best to combine the predictions of the ensemble members.

Ensemble learning is useful for improving predictability and reducing variance in stochastic learning algorithms, such as artificial neural networks. Some examples of popular group-learning algorithms are weighted average, stacked generalization (stacking), and bootstrap aggregation (packing), In this work, we used the popular group learning technique in deep learning for better classification accuracy, which is :

- Probability Averaging:

In this ensemble technique, the probability scores for multiple models are first computed. Then, the scores are averaged over all the models for all the classes in the dataset.

Probability scores are the confidence in predictions by a particular model. So, here we are pooling the confidences of several models to generate a final probability score for the ensemble. The class that has the highest probability after the averaging operation is assigned as the predicted class.

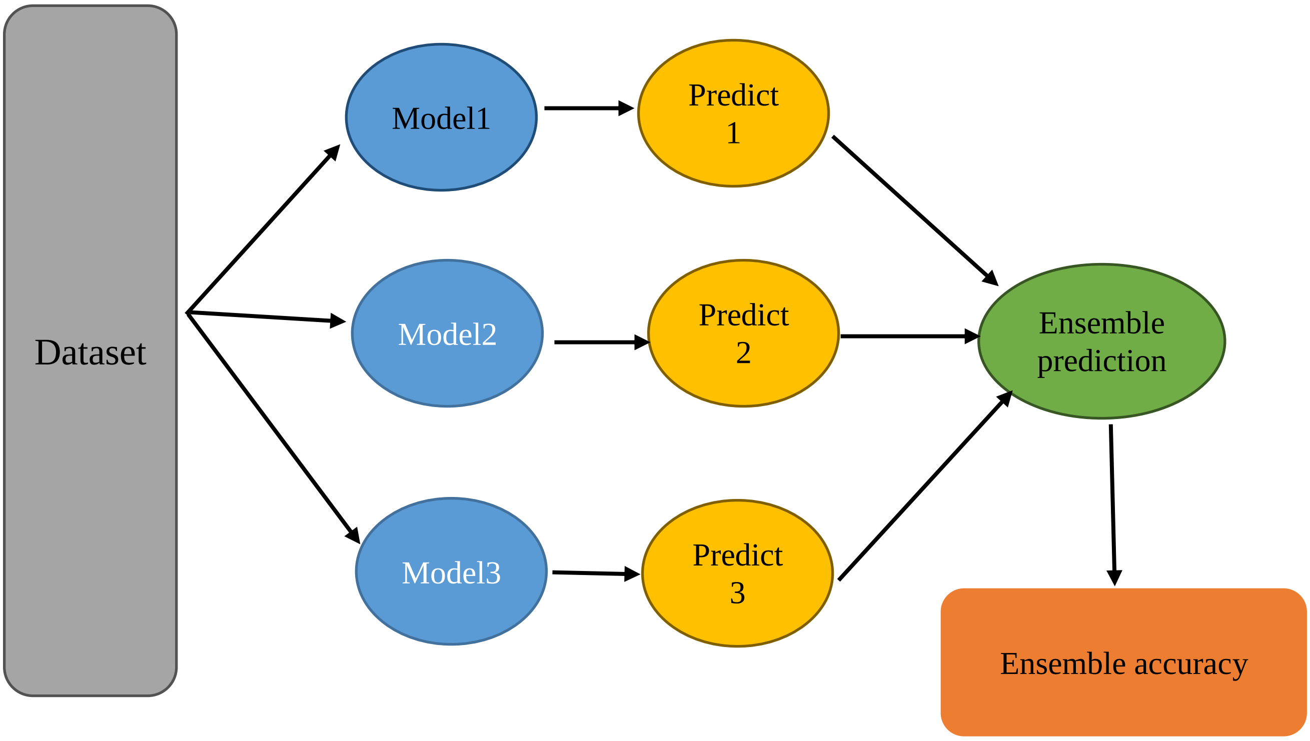


Figure 18: Shema of ensemble learning

5. Conclusion:

In this chapter, we presented a working method that was used in the thesis of creating the data set and the categories used in it, which took time and careful research, followed by the CNN models used and their explanation

At the end of the chapter the proposed technique is presented. In group learning with a diagram illustrating the work

CHAPTER 4: EXPERIMENTAL RESULTS

1. INTRODECTION:

In this chapter, the results obtained after implementing the proposed models are presented. Then evaluate the results.

2. Python:

In my thesis I use Python, a high-level general-purpose programming language widely used in data science and for the production of deep learning algorithms. This brief tutorial introduces Python and its libraries.

3. Libraries:

Our model need to perform better than this. I am using the following libraries: os, pandas, numpy, cv2, keras, Matplotlib and TensorFlow

- Tensorflow

TensorFlow est une plate-forme Open Source de bout en bout dédiée au machine learning. Elle propose un écosystème complet et flexible d'outils, de bibliothèques et de ressources communautaires permettant aux chercheurs d'avancer dans le domaine du machine learning, et aux développeurs de créer et de déployer facilement des applications qui exploitent cette technologie.[9]

- Keras

Keras is a deep learning API written in Python, running on top of the machine learning platform TensorFlow. It was developed with a focus on enabling fast experimentation. Being able to go from idea to result as fast as possible is key to doing good research.[10]

- Numpy

This library, whose name means numerical Python, constitutes the core of many other Python libraries that have originated from it. Indeed, NumPy is the foundation library for scientific computing in Python since it provides data structures and high-performing functions that the basic package of the Python cannot provide. In fact, NumPy defines a specific data structure that is an N-dimensional array defined as ndarray.[11]

- Matplotlib

This package is the Python library that is currently most popular for producing plots and other data visualizations in 2D. Since data analysis requires visualization tools, this is the library that best suits this purpose.[12]

- OpenCV

OpenCV-Python is a library of Python bindings designed to solve computer vision problems.cv2.line () method is used to draw a line on any image.[13]

- Os

The OS module in Python provides functions for interacting with the operating system. OS comes under Python’s standard utility modules. This module provides a portable way of using operating system dependent functionality. The *os* and *os.path* modules include many functions to interact with the file system.[14]

- Pandas

Pandas is an open-source Python Library providing high-performance data manipulation and analysis tool using its powerful data structures. The name Pandas is derived from the word Panel Data – an Econometrics from Multidimensional data. In 2008, developer Wes McKinney started developing pandas when in need of high performance, flexible tool for analysis of data. Prior to Pandas, Python was majorly used for data munging and preparation. It had very less contribution towards data analysis. Pandas solved this problem. Using Pandas, we can accomplish five typical steps in the processing and analysis of data, regardless of the origin of data — load, prepare, manipulate, model, and analyze. Python with Pandas is used in a wide range of fields including academic and commercial domains including finance, economics, Statistics, analytics, etc.**[15]**

4. Confusionmatrix andtest accuracy:

Confusion matrix is a very popular measure used while solving classification problems. It can be applied to binary classification as well as for multiclass classification problems. In (**figure20)**a confusion matrix for multiclass classification

From the confusion matrix, the following details will be provided:

- **True Positive (TP):** The value predicted by the algorithm corresponds to the correct bite class. We can conclude that the algorithm is correctly classified.
- **False negative (FN):**The expected output is a false negative, in which bites are incorrectly categorized
- **True Negative (TN):**The expected output is a true negative when the value expected by the algorithm corresponds to incorrect bites. We can conclude that the algorithm is appropriately classified.
- **False Positive (FP):**Inaccurately classified bite
- **Negative (N):**a value of 0 used to represent a negative state
- **Positive (P):**The value 1 is used to represent a positive state,

Confusion matrices represent counts from predicted and actual values. The output “TN” stands for True Negative which shows the number of negative examples classified accurately. Similarly, “TP” stands for True Positive which indicates the number of positive examples classified accurately. The term “FP” shows False Positive value, i.e., the number of actual negative examples classified as positive; and “FN” means a False Negative value which is the number of actual positive examples classified as negative. One of the most commonly used metrics while performing classification is accuracy. The accuracy of a model (through a confusion matrix) is calculated using the given formula below.

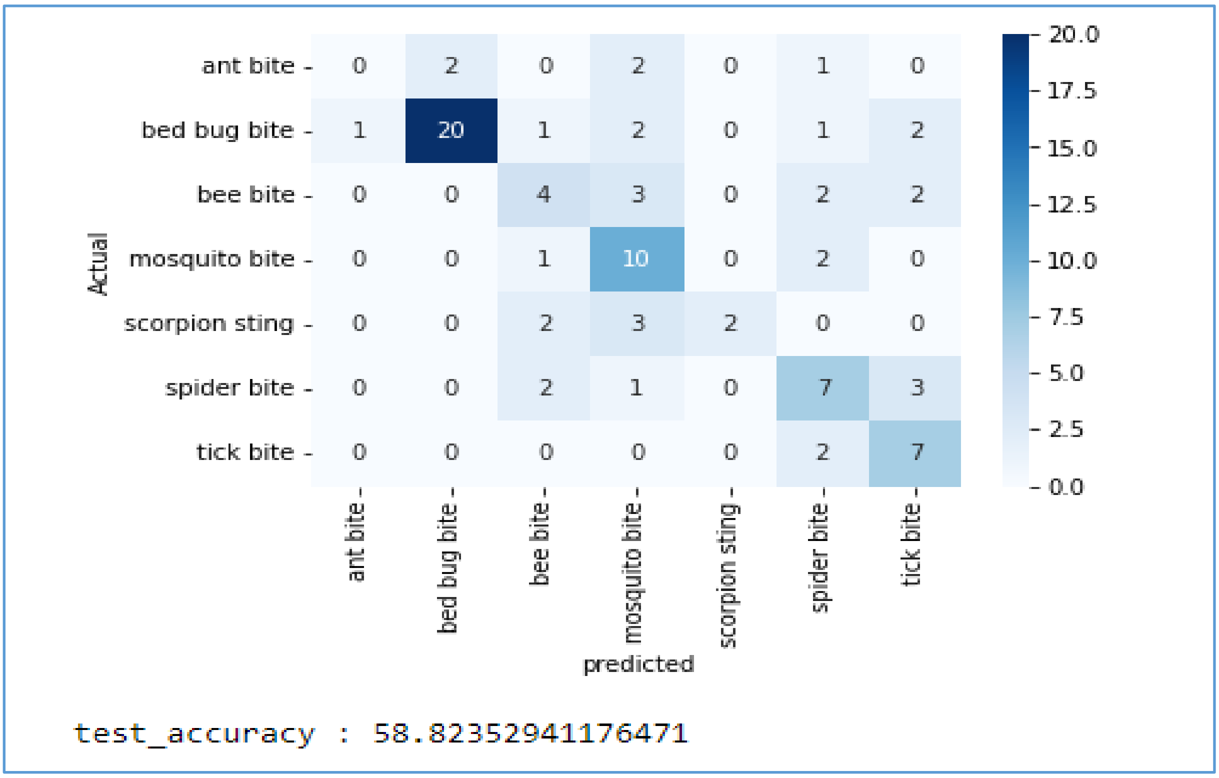
$$\text{Accuracy} = \frac{\text{TN+TP}}{\text{TN+FP+FN+TP}}$$

5. Experimental results:

After training and testing CNN models i got the following results:

- ### ResNet50 model:

The figure (**Figure 20**) below displays the confusion matrix of the (**ResNet50**) model. We note that the model was able to classify four categories relatively correctly, and the remaining three were not able to classify them clearly and accurately.



Actual \ predicted	ant bite	bed bug bite	bee bite	mosquito bite	scorpion sting	spider bite	tick bite
ant bite	0	2	0	2	0	1	0
bed bug bite	1	20	1	2	0	1	2
bee bite	0	0	4	3	0	2	2
mosquito bite	0	0	1	10	0	2	0
scorpion sting	0	0	2	3	2	0	0
spider bite	0	0	2	1	0	7	3
tick bite	0	0	0	0	0	2	7

```
test_accuracy : 58.82352941176471
```

Figure 19 : confusion matrix (ResNet50)

The curve below **(Figure 21)** shows the train value (orange) and the test loss value (blue) in relation to the number of epochs

- We note that the value of the test loss gradually decreases with the increase in the number of epochs. On the other hand, the value of the training increases until it is fixed at the optimum value.

- We conclude from this analysis that the training went perfectly in this model

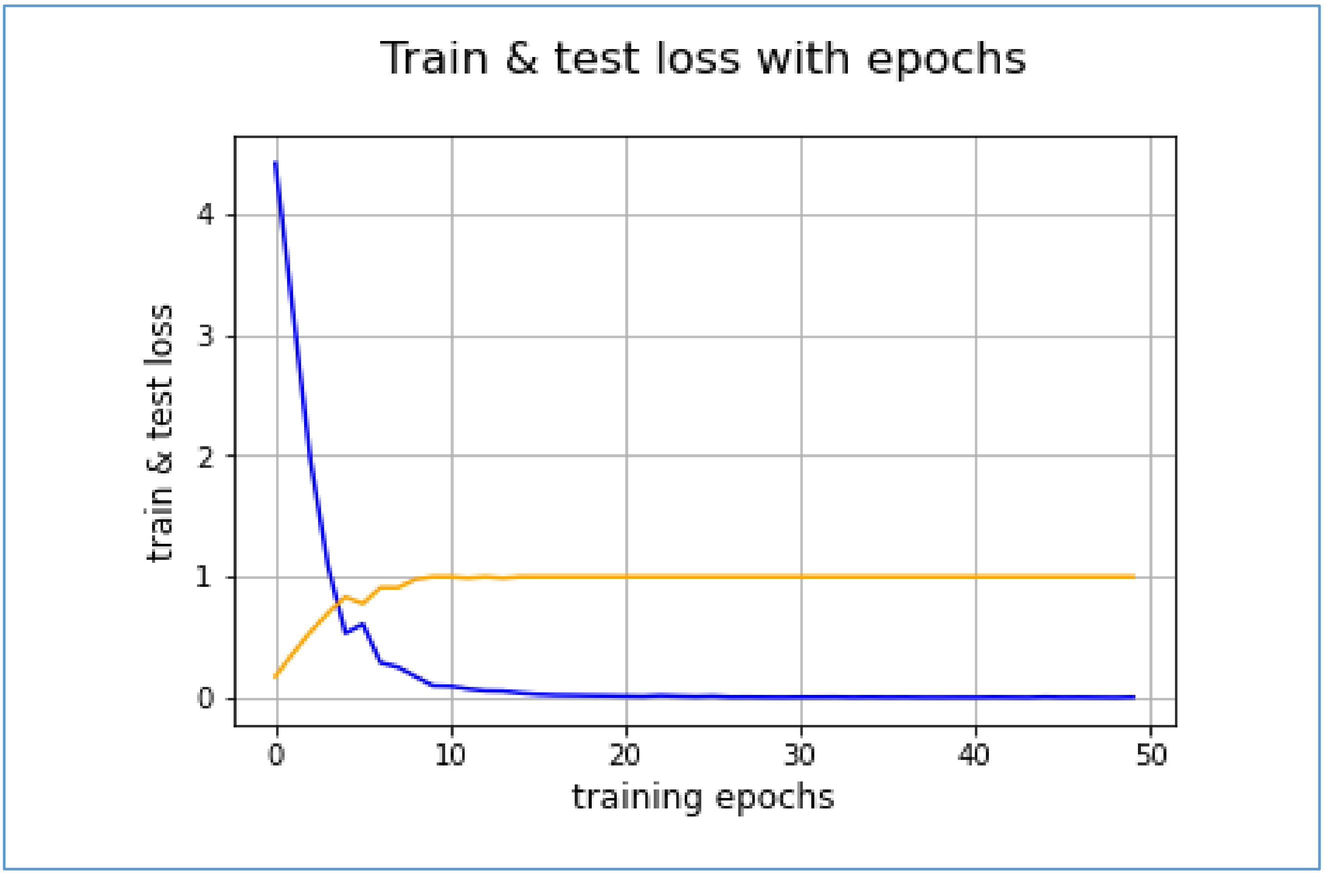
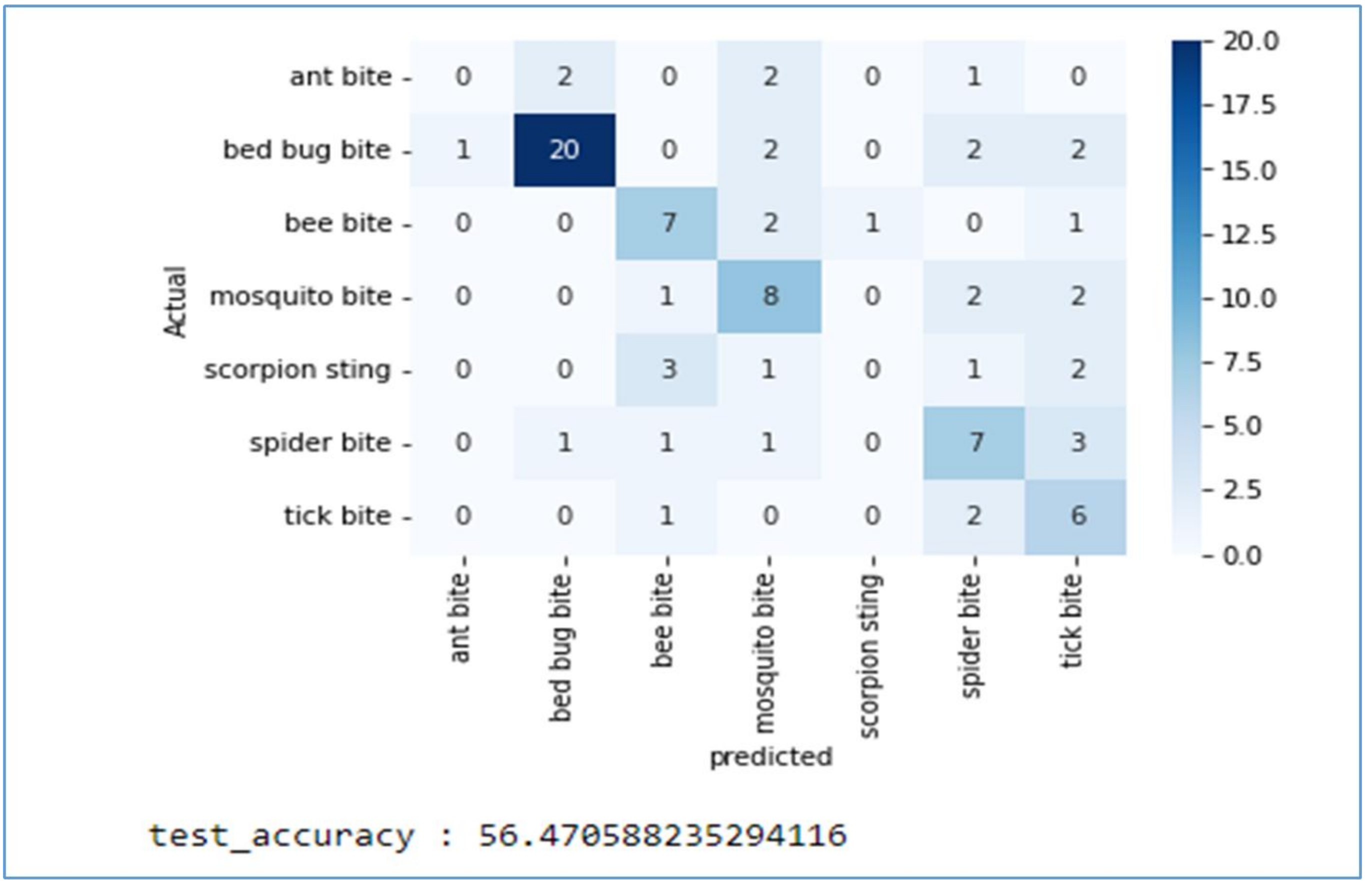


Figure 20: train accuracy and test loss (ResNet50)

- VGG16 model:

The figure (**Figure 23**) below displays the confusion matrix of the (**VGG 16**) model. We note that the model was able to classify **5** categories relatively correctly, and the remaining **2**were not able to classify them clearly and accurately.



	ant bite	bed bug bite	bee bite	mosquito bite	scorpion sting	spider bite	tick bite
ant bite	0	2	0	2	0	1	0
bed bug bite	1	20	0	2	0	2	2
bee bite	0	0	7	2	1	0	1
mosquito bite	0	0	1	8	0	2	2
scorpion sting	0	0	3	1	0	1	2
spider bite	0	1	1	1	0	7	3
tick bite	0	0	1	0	0	2	6

```
test_accuracy : 56.470588235294116
```

Figure 21 :confusion matrix (VGG16)

The curve below **(Figure 24)** shows the train value (orange) and the test loss value (blue) in relation to the number of epochs.

- We note that the value of the test loss gradually decreases with the increase in the number of courses, in return the value of training increases with the increase in the number of epochs.
- We notice in each of them a turbulence of increasing and decreasing instantaneously.
- We conclude from this analysis that the training went well in general

since the value of the loss is decreasing and the accuracy is increasing in this model

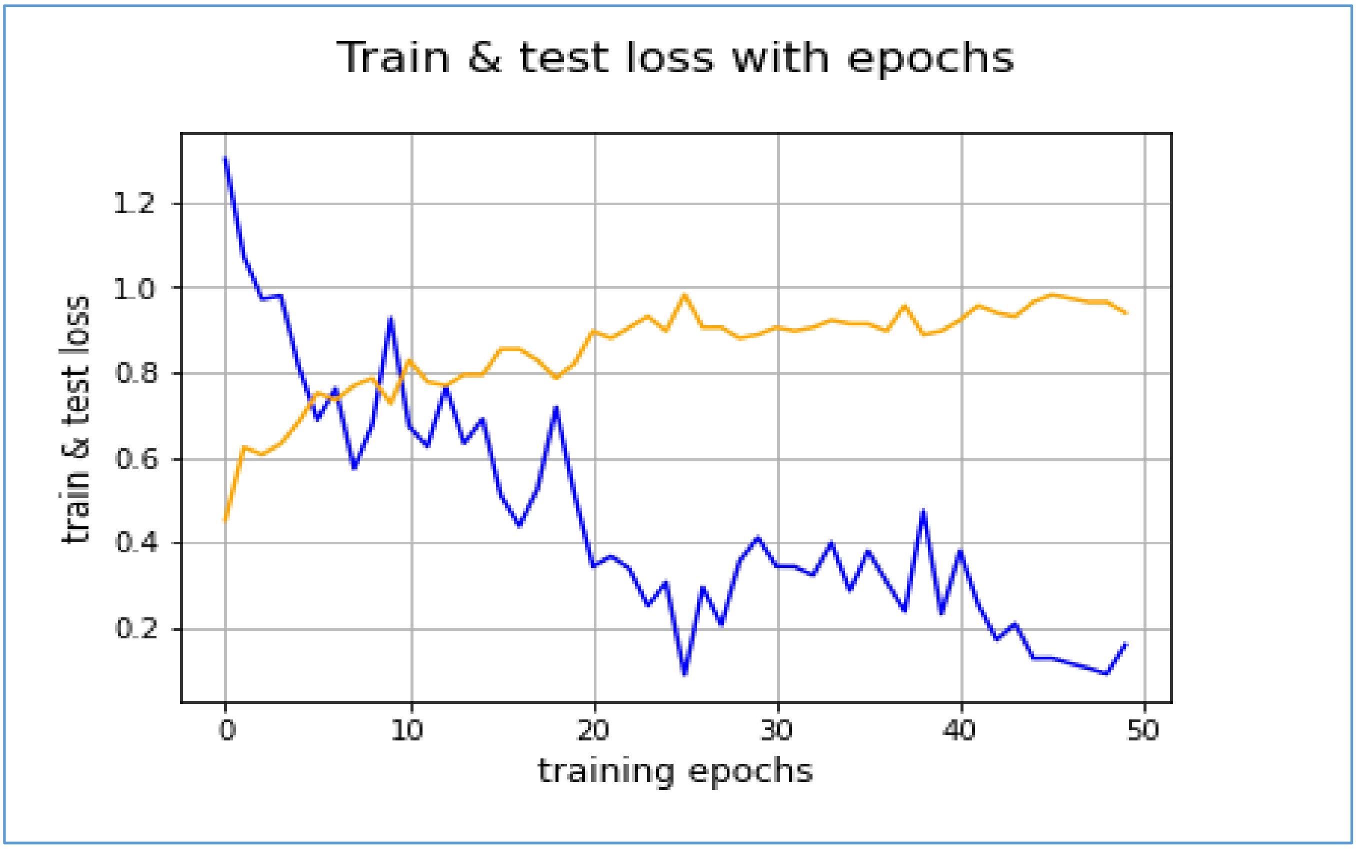
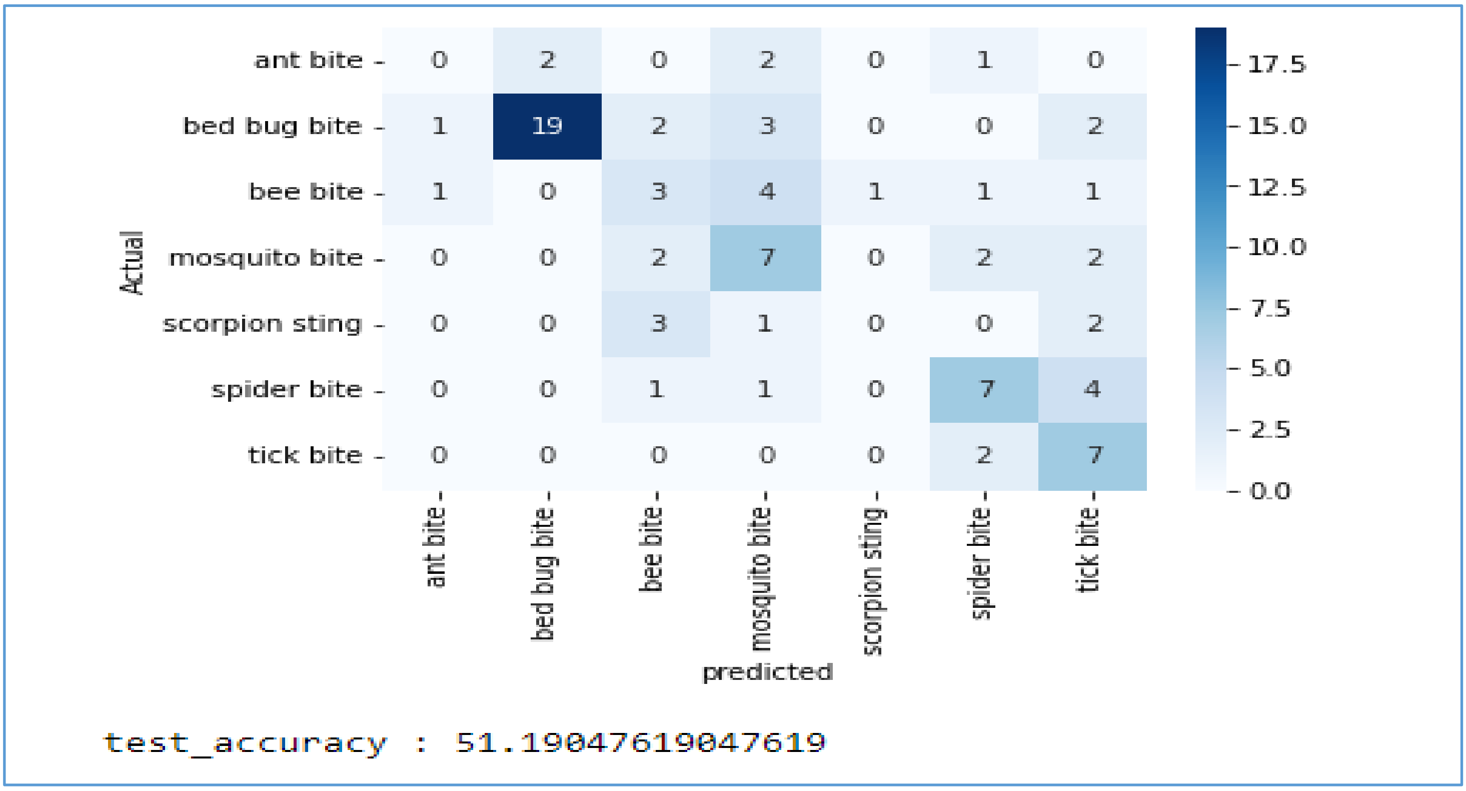


Figure 22 : train accuracy and test loss (VGG16)

- InceptionV3 model:

The figure (**Figure25**) below displays the confusion matrix of the (**InceptionV3**) model. We note that the model was able to classify four categories relatively correctly, and the remaining three were not able to classify them clearly and accurately.



	ant bite	bed bug bite	bee bite	mosquito bite	scorpion sting	spider bite	tick bite
ant bite	0	2	0	2	0	1	0
bed bug bite	1	19	2	3	0	0	2
bee bite	1	0	3	4	1	1	1
mosquito bite	0	0	2	7	0	2	2
scorpion sting	0	0	3	1	0	0	2
spider bite	0	0	1	1	0	7	4
tick bite	0	0	0	0	0	2	7

```
test_accuracy : 51.19047619047619
```

Figure 23:confusion matrix (InceptionV3)

The curve below **(Figure 26)** shows the train value (orange) and the test loss value (blue) in relation to the number of epochs.

- We notice a turbulence of an instantaneous increase and decrease in the value of the test score and the training accuracy
- We conclude from this analysis that the training is generally good
- Since the value of the loss is decreasing and the accuracy is increasing in this model

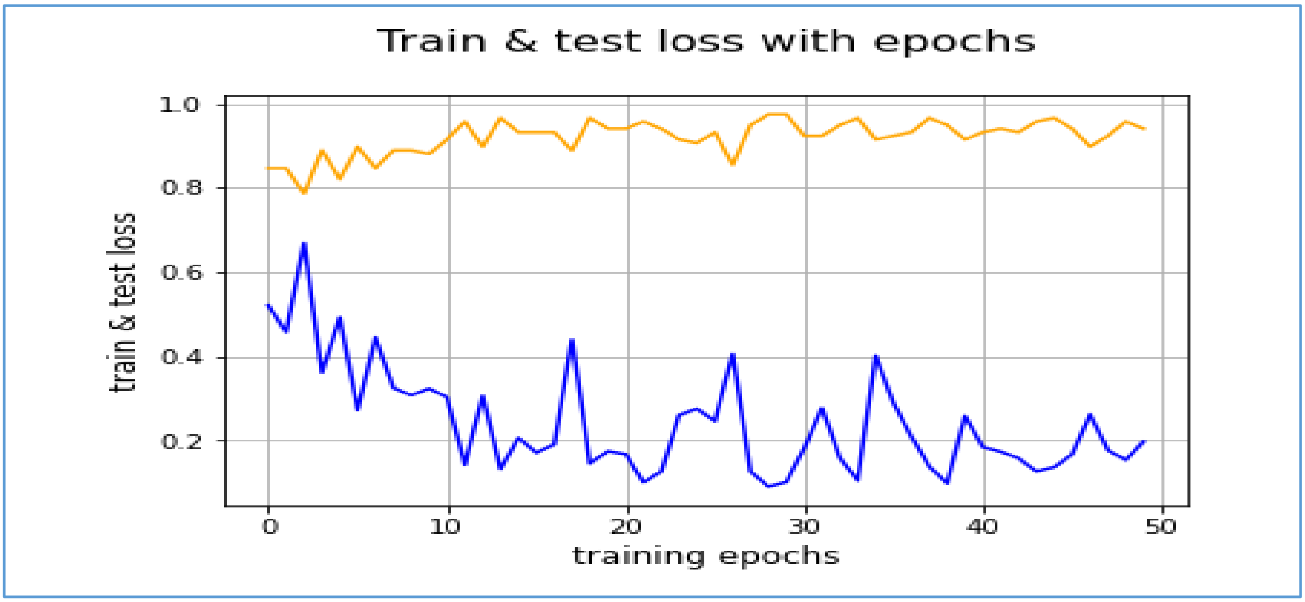


Figure 24 : train accuracy and test loss (InceptionV3)

6. Ensemble predictions:

- Prediction 1(ResNet50)

Tableau 3 : prediction of resnet50

classes	1	2	3	4	5	6	7
predict	0	0.74	0.40	0.76	0.28	0.53	0.77

- Prediction 2(VGG 16)

Tableau 4 : prediction of VGG16

Classes	1	2	3	4	5	6	7
predict	0	0.74	0.63	0.61	0	0.53	0.66

- Prediction 3(InceptionV3)

Tableau 5 :prediction of inceptionV3

classes	1	2	3	4	5	6	7
predict	0	0.70	0.27	0.53	0	0.53	0.77

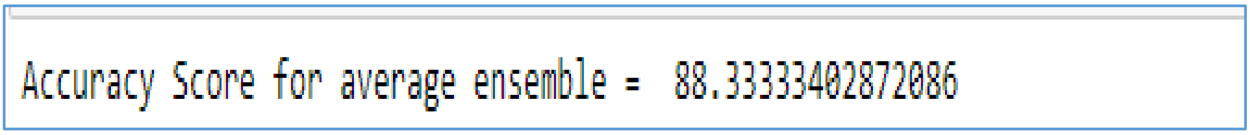
- Ensemble predictions :

Tableau 6 : ensemble predictions

classes	1	2	3	4	5	6	7
predict	0	0.72	0.43	0.63	0.09	0.53	0.73

7. Ensemble accuracy:

This isthe final result (**Figure 25**)of ensemblelearning, which is a good experimental result, and we note that the ensemble learning method improved the accuracy compared to each model alone.



```
Accuracy Score for average ensemble =  88.33333402872086
```

Figure 26: Ensemble accuracy

8. Conclusion:

In this chapter we provide some details for the implementation of the proposed method, such as the libraries used and the programming language used, then we report on the experimental results of the experiments performed on the dataset and we have given some ideas about the results. Obtained to facilitate understanding of those results.

GENERAL CONCLUSION:

With our first experience in the subject, the results are considered acceptable and can be developed in the future by providing a large and accurate data set to work on until the system becomes more accurate

Through this system we aim to reduce the damage of bites that harm humans, because often it is not possible to identify the insect until after a long time, and now through this system it will be easy for people to know the type of insect that bit them during their sleep or omission, and to know what Whether it is dangerous or not, and also as an assistant in the field of ambulance and keeping pace with technological development

And also it can be developed and become a mobile application for everyone and it can be useful for people who travel in remote and isolated places and hikers in forests, mountains and deserts

This system can become a basic medical assistant if we can collect a huge data set that includes all kinds of insects and stinging animals

We aspire in the future, with the technological development taking place in artificial intelligence, which this system will become an assistant robot in the field of medicine in the interest of urgency, reducing the fatigue of doctors and nurses on duty, especially since most of the stings occur at night.

REFERENCES

1. Gonzalez, R. C., & Woods, R. E. (2002). Digital Image Processing, New Jersey.
2. Khaldi, B., O. Aiadi, and M.L. Kherfi, Combining colour and grey-level co-occurrence matrix features: a comparative study. IET Image Processing, 2019. 13(9): p. 1401-1410
3. Mitchell, T. M., & Mitchell, T. M. (1997). Machine learning (Vol. 1, No. 9). New York: McGraw-hill.
4. Rungta, K. (2018). TensorFlow in 1 Day: Make your own Neural Network. Amazon Digital Services LLC-Kdp Print Us.
5. Sairamya, N. J., Susmitha, L., George, S. T., & Subathra, M. S. P. (2019). Hybrid approach for classification of electroencephalographic signals using time–frequency images with wavelets and texture features. In Intelligent Data Analysis for Biomedical Applications (pp. 253-273). Academic Press.
6. Ghosh, A., Sufian, A., Sultana, F., Chakrabarti, A., & De, D. (2020). Fundamental concepts of convolutional neural network. In Recent trends and advances in artificial intelligence and Internet of Things (pp. 519-567). Springer, Cham.
7. Ghosh, A., Sufian, A., Sultana, F., Chakrabarti, A., & De, D. (2020). Fundamental concepts of convolutional neural network. In Recent trends and advances in artificial intelligence and Internet of Things (pp. 519-567). Springer, Cham.
8. Ghosh, A., Sufian, A., Sultana, F., Chakrabarti, A., & De, D. (2020). Fundamental concepts of convolutional neural network. In Recent trends and advances in artificial intelligence and Internet of Things (pp. 519-567). Springer, Cham.
9. Abadi, M. (2016, September). TensorFlow: learning functions at scale. In Proceedings of the 21st ACM SIGPLAN International Conference on Functional Programming (pp. 1-1).
10. Ketkar, N. (2017). Introduction to keras. In Deep learning with Python (pp. 97-111). Apress, Berkeley, CA.
11. Nelli, F. (2018). Python data analytics. Apress Media, California.
12. Nelli, F. (2018). Python data analytics. Apress Media, California.
13. Kavitha, D., Kiran, B. R., Niteesh, B., & Praveen, S. (2021). Multiple Object Recognition Using OpenCV. REVISTA GEINTEC-GESTAO INOVACAO E TECNOLOGIAS, 11(2), 1736-1747.

14. Saabith, A. S., Fareez, M. M. M., & Vinothraj, T. (2019). Python current trend applications-an overview. International Journal of Advance Engineering and Research Development, 6(10).
15. McKinney, W. (2012). Python for data analysis: Data wrangling with Pandas, NumPy, and IPython. " O'Reilly Media, Inc.".
16. Chebbi, I. (2021). VGG16: Visual Generation of Relevant Natural Language Questions from Radiology Images for Anomaly Detection.