## ABSTRACT

With regular manual diagnosis of blood smears, it is an intensive manual process requiring proper expertise in classifying and counting the parasitized and uninfected cells. Typically, this may not scale well and might cause problems if we do not have the right expertise in specific regions around the world. However, these models are not scalable with more data being available for training and given the fact that hand-engineered features take a lot of time. Deep Learning models, or to be more specific, Convolutional Neural Networks (CNNs) have proven to be really effective in a wide variety of computer vision tasks.

**INTRODUCTION**

Deep Learning is a subfield of machine learning concerned with algorithms inspired by the structure and function of the brain called artificial neural networks. Deep learning is the most interesting and powerful machine learning technique. Top deep learning libraries are available on the Python ecosystem like Keras and TensorFlow. Deep learning techniques are so powerful because they learn the best way to represent the problem while learning how to solve the problem. This is called representation learning.

Representation learning is perhaps the biggest differentiation between deep learning models and classical machine learning algorithm. It is the power of representation learning that is spurring such great creativity in the way the techniques are being used.

For example:

* Deep learning models are being used for very difficult problems and making progress, like colorizing image and videos based on the context in the scene.
* Deep learning models are being used in bold new ways, such as cutting the head off a network trained on one problem and tuning it for a completely different problem, and getting impressive results.
* Combinations of deep learning models are being used to both identify objects in photographs and then generate textual descriptions of those objects, a complex multi-media problem that was previously thought to require large artificial intelligence systems.

Deep learning is a tool that you can use on your machine learning projects. It does not have to be a theoretical academic pursuit that you study in gritty detail.

Keras and Tensor Flow are two top numerical libraries for developing deep learning models, but are too technical and complex for the average practitioner. They are intended more for research and development teams and academics interested in developing wholly new deep learning algorithms.

**Objectives of the Project:**

Malaria is a life-threatening disease caused by parasites that are transmitted to people through the bites of infected mosquitoes. Automation of the diagnosis process will enable accurate diagnosis of the disease and hence holds the promise of delivering reliable health- care to resource-scarce areas. Machine Learning technologies have been used for automated diagnosis of malaria. We present some of our recent progresses on highly accurate classification of malariainfected cells using deep convolutional neural networks. First, we describe image processing methods used for segmentation of red blood cells from whole slide images. Convolutional Neural Network (CNN) is a class of deep neural networks, most commonly applied to analysing visual imagery. Convolutional networks were inspired by biological processes in that the connectivity pattern between neurons resembles the organization of the animal visual cortex

**Problem Statement:**

Malaria must be recognized promptly in order to treat the patients in time and to prevent further spread of infection in the community via local mosquitoes.Malaria can be suspected based on the patient’s travel history, symptoms, and the physical findings at examination. However, for a definitive diagnosis to be made, laboratory tests must demonstrate the malaria parasites or their components.

But for this solution has very less accuracy. Malaria can occur if a mosquito infected with the *Plasmodium* parasite bites you. Malaria is transmitted by blood, so it can also be transmitted through:

* an organ transplant
* a transfusion
* use of shared needles or syringes

The symptoms of malaria typically develop within 10 days to 4 weeks following the infection. In some cases, symptoms may not develop for several months. Some malarial parasites can enter the body but will be dormant for long periods of time.Malaria can cause a number of life-threatening complications.

The following may occur:

* swelling of the blood vessels of the brain, or cerebral malaria
* an accumulation of fluid in the lungs that causes breathing problems
* organ failure [of the kidneys,](https://www.healthline.com/health/kidney-failure) [liver,](https://www.healthline.com/health/hepatic-failure) or spleen
* anemia due to the destruction of red blood cells
* low blood sugar

**LITERATURE REVIEW**

Booyens 2013, decribed prompt and accurate diagnosis is critical to the effective management of malaria. The global impact of malaria has spurred interest in developing effective diagnostic strategies not only for resource-limited areas where malaria is a substantial burden on society, but also in developed countries, where malaria diagnostic expertise is often lacking. Malaria diagnosis involves identifying malaria parasites or antigens/products in patient blood. Although this may seem simple, the diagnostic efficacy is subject to many factors. The different forms of the 5 malaria species; the different stages of erythrocytic schizogony, the endemicity of different species, the interrelation between levels of transmission, population movement, parasitemia, immunity, and signs and symptoms; drug resistance, the problems of recurrent malaria, persisting viable or non-viable parasitemia, and sequestration of the parasites in the deeper tissues, and the use of chemoprophylaxis or even presumptive treatment on the basis of clinical diagnosis, can all influence the identification and interpretation of malaria parasitemia in a diagnostic test.

Malaria, sometimes called the "King of Diseases", is caused by protozoan parasites of the genus *Plasmodium*. The most serious and sometimes fatal type of malaria is caused by *Plasmodium falciparum*. The other human malaria species, *P. vivax*, *P. ovale*, *P. malariae*, and sometimes *P. knowlesi* can cause acute, severe illness but mortality rates are low. Malaria is the most important infectious disease in tropical and subtropical regions, and continues to be a major global health problem, with over 40% of the world's population exposed to varying degrees of malaria risk in some 100 countries. It is estimated that over 500 million people suffer from malaria infections annually, resulting in about 1-2 million deaths, of whom 90% are children in sub- Saharan Africa. The number of malaria cases worldwide seems to be increasing, due to increasing transmission risk in areas where malaria control has declined, the increasing prevalence of drugresistant strains of parasites, and in a relatively few cases, massive increases in international travel and migration.

The need for effective and practical diagnostics for global malaria control is increasing, since effective diagnosis reduces both complications and mortality from malaria. Differentiation of clinical diagnoses from other tropical infections, based on patients' signs and symptoms or physicians' findings, may be difficult. Therefore, confirmatory diagnoses using laboratory technologies are urgently needed. This review discusses on the currently available diagnostic methods for malaria in many settings, and assesses their feasibility in resource-rich and resource-poor settings.

With better CNN models and with better epoch values and hidden layers we are proposing a model based on Deep Learning to improve the accuracy level. To know whether a patient exactly has malaria or not based on the samples given by him/her. The proposed system shows better accuracy when compared to the older models. Based on CNN we are doing our predictions to get accurate values.

Technically, deep learning CNN models to train and test, each input image will pass it through a series of convolution layers with filters (Kernal), Pooling, fully connected layers (FC) and apply Softmax function to classify an object with probabilistic values between 0 and 1. A Convolutional Neural Network (ConvNet/CNN) is a Deep Learning algorithm which can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image and be able to differentiate one from the other. CNNs are used for image classification and recognition because of its high accuracy.

The CNN follows a hierarchical model which works on building a network, like a funnel, and finally gives out a fully-connected layer where all the neurons are connected to each other and the output is processed. The trained model is deployed in the IBM cloud and UI is developed using the Node-RED flow. Node-RED provides a web browser-based flow editor, which can can be used to create JavaScript functions. Elements of applications can be saved or shared for re-use. The runtime is built on Node.js. The flows created in NodeRED are stored using JSON. Node-RED can used in cloud services like Cisco, Nokia Innovation Platform, AT&T, AT&T IOT Platform, IBM Cloud and other platforms.

Malaria is a major cause of death in tropical and sub-tropical countries, killing each year over 1 million people globally; 90% of fatalities occur in African children. Although effective ways to manage malaria now exist, the number of malaria cases is still increasing, due to several factors. In this emergency situation, prompt and effective diagnostic methods are essential for the management and control of malaria. Traditional methods for diagnosing malaria remain problematic; therefore, new technologies have been developed and introduced to overcome the limitations. This review details the currently available diagnostic methods for malaria.

According to Kevin et al. 2014 a clinical diagnosis of malaria is traditional among medical doctors. This method is least expensive and most widely practiced. Clinical diagnosis is based on the patients' signs and symptoms, and on physical findings at examination. The earliest symptoms of malaria are very nonspecific and variable, and include fever, headache, weakness, myalgia, chills, dizziness, abdominal pain, diarrhea, nausea, vomiting, anorexia, and pruritus. A clinical diagnosis of malaria is still challenging because of the non-specific nature of the signs and symptoms, which overlap considerably with other common, as well as potentially life-threatening diseases, e.g. common viral or bacterial infections, and other febrile illnesses.

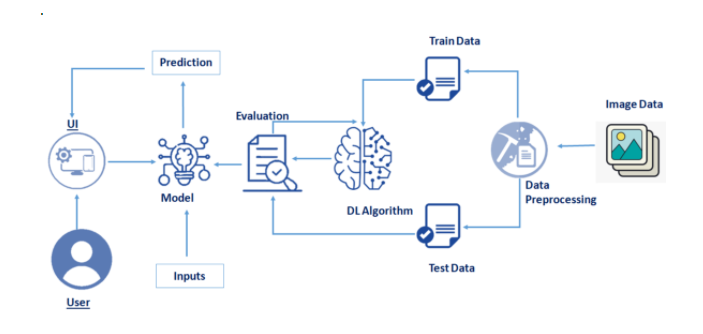
The overlapping of malaria symptoms with other tropical diseases impairs diagnostic specificity, which can promote the indiscriminate use of antimalarials and compromise the quality of care for patients with non-malarial fevers in endemic areas. The Integrated Management of Children Illness (IMCI) has provided clinical algorithms for managing and diagnosing common childhood illnesses by minimally trained healthcare providers in the developing world having inappropriate equipment for laboratory diagnosis.

A widely utilized clinical algorithm for malaria diagnosis, compared with a fully trained pediatrician with access to laboratory support, showed very low specificity (0-9%) but 100% sensitivity in African settings. This lack of specificity reveals the perils of distinguishing malaria from other causes of fever in children on clinical grounds alone. Recently, another study showed that use of the IMCI clinical algorithm resulted in 30% over- diagnosis of malaria. Therefore, the accuracy of malaria diagnosis can be greatly enhanced by combining clinical-and parasite-based findings.

Rapid and effective malaria diagnosis not only alleviates suffering, but also decreases community transmission. The nonspecific nature of the clinical signs and symptoms of malaria may result in over-treatment of malaria or non-treatment of other diseases in malaria- endemic areas, and misdiagnosis in non-endemic areas. In the laboratory, malaria is diagnosed using different techniques, e.g. conventional microscopic diagnosis by staining thin and thick peripheral blood smears , other concentration techniques, e.g. quantitative buffy coat (QBC) method, rapid diagnostic tests e.g., OptiMAL, ICT, Para-HIT-f, ParaScreen, SD Bioline , Paracheck , and molecular diagnostic methods, such as polymerase chain reaction (PCR) .

Some advantages and shortcomings of these methods have also been described, related to sensitivity, specificity, accuracy, precision, time consumed, cost-effectiveness, labor intensiveness, the need for skilled microscopists, and the problem of inexperienced technicians.Malaria is conventionally diagnosed by microscopic examination of stained blood films using Giemsa, Wright's, or Field's stains. This method has changed very little since Laverran's original discovery of the malaria parasite, and improvements in staining techniques by Romanowsky in the late 1,800s. More than a century later, microscopic detection and identification of *Plasmodium* species in Giemsa-stained thick blood films (for screening the presenting malaria parasite), and thin blood films (for species' confirmation) remains the gold standard for laboratory diagnosis . Malaria is diagnosed microscopically by staining thick and thin blood films on a glass slide, to visualize malaria parasites. Briefly, the patient's finger is cleaned with 70% ethyl alcohol, allowed to dry and then the side of fingertip is picked with a sharp sterile lancet and two drops of blood are placed on a glass slide.

**Methodology:**



Classification techniques are an essential part of machine learning and data mining applications. Approximately 70% of problems in Data Science are classification problems. There are lots of classification problems that are available, but the logistic regression is common and is a useful regression method for solving the binary classification problem. Another category of classification is Multinomial classification, which handles the issues where multiple classes are present in the target variable.

Feature engineering is the process of using domain knowledge of the data to create features that make machine learning algorithms work."When working on a machine learning problem, feature engineering is manually designing what the input x's should be.”

Machine-learning technology powers many aspects of modern society: from web searches to content filtering on social networks to recommendations on e-commerce websites, and it is increasingly present in consumer products such as cameras and smartphones.Machine-learning systems are used to identify objects in images, transcribe speech into text, match news items, posts or products with users’ interests, and select relevant results of search. Increasingly, these applications make use of a class of techniques called deep learning.

Conventional machine-learning techniques were limited in their ability to process natural data in their raw form. For decades, constructing a patternrecognition or machine- learning system required careful engineering and considerable domain expertise to design a feature extractor that transformed the raw data (such as the pixel values of an image) into a suitable internal representation or feature vector from which the learning subsystem, often a classifier, could detect or classify patterns in the input.

Deep learning is making major advances in solving problems that have resisted the best attempts of the artificial intelligence community for many years. It has turned out to be very good at discovering intricate structures in highdimensional data and is therefore applicable to many domains of science, business and government. In addition to beating records in image recognition and speech recognition, it has beaten other machine-learning techniques at predicting the activity of potential drug molecules, analysing particle accelerator data reconstructing brain circuits, and predicting the effects of mutations in noncoding DNA on gene expression and disease. Perhaps more surprisingly, deep learning has produced extremely promising results for various tasks in natural language understanding, particularly topic classification, sentiment analysis, question answering and language translation.

We think that deep learning will have many more successes in the near future because it requires very little engineering by hand, so it can easily take advantage of increases in the amount of available computation and data. New learning algorithms and architectures that are currently being developed for deep neural networks will only accelerate this progress.

**Supervised Learning**

The most common form of machine learning, deep or not, is supervised learning. Imagine that we want to build a system that can classify images as containing, say, a house, a car, a person or a pet. We first collect a large data set of images of houses, cars, people and pets, each labelled with its category. During training, the machine is shown an image and produces an output in the form of a vector of scores, one for each category. We want the desired category to have the highest score of all categories, but this is unlikely to happen before training.

We compute an objective function that measures the error (or distance) between the output scores and the desired pattern of scores. The machine then modifies its internal adjustable parameters to reduce this error. These adjustable parameters, often called weights, are real numbers that can be seen as ‘knobs’ that define the input–output function of the machine. In a typical deep-learning system, there may be hundreds of millions of these adjustable weights, and hundreds of millions of labelled examples with which to train the machine.

To properly adjust the weight vector, the learning algorithm computes a gradient vector that, for each weight, indicates by what amount the error would increase or decrease if the weight were increased by a tiny amount. The weight vector is then adjusted in the opposite direction to the gradient vector.

**Convolutional Neural Networks**:

ConvNets are designed to process data that come in the form of multiple arrays, for example a colour image composed of three 2D arrays containing pixel intensities in the three colour channels. Many data modalities are in the form of multiple arrays: 1D for signals and sequences, including language; 2D for images or audio spectrograms; and 3D for video or volumetric images. There are four key ideas behind ConvNets that take advantage of the properties of natural signals: local connections, shared weights, pooling and the use of many layers.

The architecture of a typical ConvNet is structured as a one consists series of stages. The first few stages are composed of two types of layers: convolutional layers and pooling layers. Units in a convolutional layer are organized in feature maps, within which each unit is connected to local patches in the feature maps of the previous layer through a set of weights called a filter bank. The result of this local weighted sum is then passed through a non-linearity such as a ReLU. All units in a feature map share the same filter bank. Different feature maps in a layer use different filter banks.

The reason for this architecture is twofold. First, in array data such as images, local groups of values are often highly correlated, forming distinctive local motifs that are easily detected. Second, the local statistics of images and other signals are invariant to location. In other words, if a motif can appear in one part of the image, it could appear anywhere, hence the idea of units at different locations sharing the same weights and detecting the same pattern in different parts of the array. Mathematically, the filtering operation performed by a feature map is a discrete convolution, hence the name.

Although the role of the convolutional layer is to detect local conjunctions of features from the previous layer, the role of the pooling layer is to merge semantically similar features into one. Because the relative positions of the features forming a motif can vary somewhat, reliably detecting the motif can be done by coarse-graining the position of each feature. A typical pooling unit computes the maximum of a local patch of units in one feature map (or in a few feature maps).

Neighbouring pooling units take input from patches that are shifted by more than one row or column, thereby reducing the dimension of the representation and creating an invariance to small shifts and distortions. Two or three stages of convolution, non-linearity and pooling are stacked, followed by more convolutional and fully-connected layers. Backpropagating gradients through a ConvNet is as simple as through a regular deep network, allowing all the weights in all the filter banks to be trained.

Deep neural networks exploit the property that many natural signals are compositional hierarchies, in which higher-level features are obtained by composing lower- level ones. In images, local combinations of edges form motifs, motifs assemble into parts, and parts form objects. Similar hierarchies exist in speech and text from sounds to phones, phonemes, syllables, words and sentences. The pooling allows representations to vary very little when elements in the previous layer vary in position and appearance.

**Image understanding with Deep Convolutional Networks:**

Since the early 2000s, ConvNets have been applied with great success to the detection, segmentation and recognition of objects and regions in images. These were all tasks in which labelled data was relatively abundant, such as traffic sign recognition, the segmentation of biological images54 particularly for connect atomics, and the detection of faces, text, pedestrians and human bodies in natural images.

A major recent practical success of ConvNets is face recognition. Importantly, images can be labelled at the pixel level, which will have applications in technology, including autonomous mobile robots and self-driving cars. Companies such as Mobile eye are using such ConvNet-based methods in their upcoming vision systems for cars. Other applications gaining importance involve natural language understanding and speech recognition.

There are multiple equivalent ways to describe the mathematical model undelying multinomial logistic regression. This can make it difficult to compare different treatments of the subject in different texts. The idea behind all of them, as in many other staistical classification techniques, is to construct a linear predictor function that constructs a score from a set of weights that are linearly combined with the explonatory variables of a given observation using a dot product.

The difference between the multinomial logit model and numerous other models is the procedure for determining the optimal weights/ coefficients and the way that the score is interpreted. In particular, in the multinomial logit model, the score can directly be converted to probability value, indicating the probability of observation i choosing outcome k given the measured characteristics of the observation.

Deep-learning theory shows that deep nets have two different exponential advantages over classic learning algorithms that do not use distributed representations. Both of these advantages arise from the power of composition and depend on the underlying data- generating distribution having an appropriate componential structure. First, learning distributed representations enable generalization to new combinations of the values of learned features beyond those seen during training (for example, 2n combinations are possible with n binary features). Second, composing layers of representation in a deep net brings the potential for another exponential advantage70 (exponential in the depth).

The represented as hidden layers of multilayer neural network learn represent the network’s inputs in a way that makes it easy to predict the target outputs. This is nicely demonstrated by training a multilayer neural network to predict the next word in a sequence from a local from image to text. Captions generated by a recurrent neural network (RNN) taking, as extra input, the representation extracted by a deep convolution neural network (CNN) from a test image, with the RNN trained to ‘translate’ high-level representations of images into captions (top).

Reproduced with permission from reference. When the RNN is given the ability to focus its attention on a different location in the input image (middle and bottom; the lighter patches were given more attention) as it generates each word (bold), we found that it exploits this to achieve better ‘translation’ of images into captions. Vision Deep CNN Language Generating RNN A group of people shopping at an outdoor market. There are many vegetables at the fruitstand. A woman is throwinga frisbee in a park. A little girl sitting on a bed with a teddy bear. A group of people sitting on a boat in the water. A girl standing in a forest with trees in the background. A dog is standing on a hardwood. A stop sign is on a road with a mountain in the background context of earlier words.

Each word in the context is presented to the network as a one-of-N vector, that is, one component has a value of 1 and the rest are 0. In the first layer, each word creates a different pattern of activations, or word vectors. In a language model, the other layers of the network learn to convert the input word vectors into an output word vector for the predicted next word, which can be used to predict the probability for any word in the vocabulary to appear as the next word.

The network learns word vectors that contain many active components each of which can be interpreted as a separate feature of the word, as was first demonstrated in the context of learning distributed representations for symbols. These semantic features were not explicitly present in the input. They were discovered by the learning procedure as a good way of factorizing the structured relationships between the input and output symbols into multiple ‘micro-rules’.

Learning word vectors turned out to also work very well when the word sequences come from a large corpus of real text and the individual micro-rules are unreliable. When trained to predict the next word in a news story, for example, the learned word vectors for Tuesday and Wednesday are very similar, as are the word vectors for Sweden and Norway. Such representations are called distributed representations because their elements (the features) are not mutually exclusive and their many configurations correspond to the variations seen in the observed data. These word vectors are composed of learned features that were not determined ahead of time by experts, but automatically discovered by the neural network. Vector representations of words learned from text are now very widely used in natural language applications.

**Characteristics of Deep Learning:**

* Supervised, Semi-Supervised or Unsupervised. When the category labels are present while you train the data then it is Supervised learning.
* Huge Amount of Resources.
* Large Amount of Layers in Model.
* Optimizing Hyper-parameters.
* Cost Function.

**Advantages:**

Microscopy is an established, relatively simple technique that is familiar to most laboratories in endemic countries. In such areas, microscopy is a standard technique used for diagnosing other diseases (such as tuberculosis), often by the same laboratories using the same facilities and equipment. Blood slide microscopy makes it possible to count the number of parasites and is more useful than rapid diagnostic tests for monitoring the effectiveness of malaria treatment.

## Disadvantages:

## Microscopy requires a level of skill often not available in many health facilities in several malaria-endemic countries, especially in remote rural areas, where most malaria transmission occurs. In addition, lack of functional microscopes or electricity to run them, lack of or sub-standard reagents such as stains, and high workloads may affect the quality of results.

**IMPLEMENTATION:**

**3.2.1 Deep Learning Using Python:**

The data collected consists of about 1470 record attributes and target attribute that is Attrition of an employee. To predict the attrition of an employee we need to build a machine learning model. For that we have used the Jupyter notebook and IBM cloud for training the model and deploying it. Building a machine learning includes the following steps.

Data Preprocessing

1. Feature Extraction
2. Model Training
3. Prediction
4. Deployment to Prediction

5**.**  Data Preprocessing

**Data Pre-processing**: Data pre-processing includes 5 steps they are:

• Importing Libraries and Reading the Dataset

• Check the missing values and considering the Co relational heat map

.

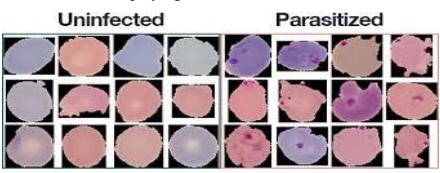
• Separating the independent and dependent variables.

• Converting the data into numpy array and perform Encoding on categorical variables.

• Splitting the dataset for training and testing.

**Step 1:**

Read the dataset and displaying the dataset.



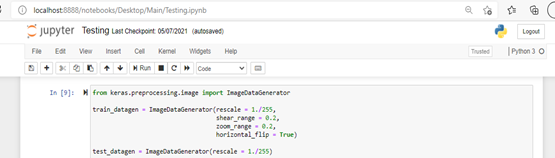
**Step 2:**

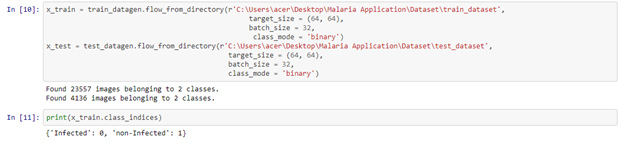
Importing Data Generator Library



**Step 3:**

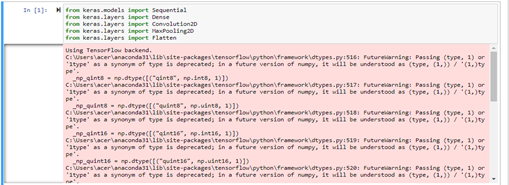
Testing and training the data





**Step 4:**

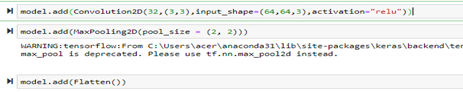
Initialize the Model





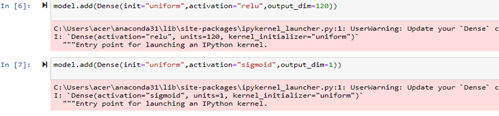
**Step 5:**

Adding Convolution layers



**Step 6:**

Relu Activation Function



**Step 7:**

Adding Adam Optimiser



**Step 8:**

Saving the model



Deep learning is a tool that you can use on your machine learning projects. It does not have to be a theoretical academic pursuit that you study in gritty detail.

Develop and evaluate deep learning models in Python. The platform for getting started in applied deep learning is Python. Python is a fully featured general purpose programming language, unlike R and Matlab. It is also quick and easy to write and understand, unlike C++ and Java. The SciPy stack in Python is a mature and quickly expanding platform for scientific and numerical computing. The platform hosts libraries such as scikit-learn the general-purpose machine learning library that can be used with your deep learning models.

It is because of these benefits of the Python ecosystem that two top numerical libraries for deep learning were developed for Python, Theano and the newer TensorFlow library released by Google (and adopted recently by the Google DeepMind research group). Keras and TensorFlow are two top numerical libraries for developing deep learning models, but are too technical and complex for the average practitioner. They are intended more for research and development teams and academics interested in developing wholly new deep learning algorithms.

The saving grace is the Keras library for deep learning, that is written in pure Python, wraps and provides a consistent agnostic interface to Theano and TensorFlow and is aimed at machine learning practitioners that are interested in creating and evaluating deep learning models. It is a little over one year old and is clearly the best-of-breed library for getting started with deep learning because of both the speed at which you can develop models and the numerical power it is built upon.

**3.2.2: Sigmoid function:**

The sigmoid function, also called logistic function gives an ‘S’ shaped curve that can take any real-valued number and map it into a value between 0 and 1. If the curve goes to positive infinity, y predicted will become 1, and if the curve goes to negative infinity, y predicted will become 0. If the output of the sigmoid function is more than 0.5, we can classify the outcome as 1 or YES, and if it is less than 0.5, we can classify it as 0 or NO.

For example: If the output is 0.75, we can say in terms of proability as: there is a 75 percent chance of winning probability. On applying the sigmoid function on above linear regression we get

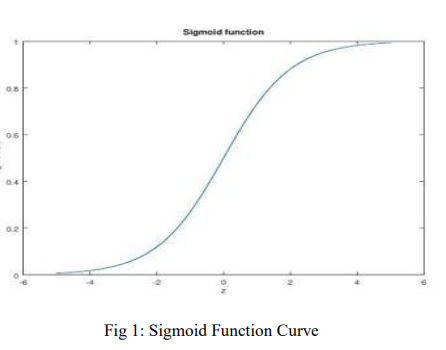
p = 1 / 1 + e^-y

e^y = (p / p-1)

y =log (p / p-1)

log (p / p-1) = β0 + β1X1 + β2X2 +...... + βnXn

The sigmoid curve for this is shown below:



**3.3 Data Preparation :**

The data was sourced from IBM HR Analytics Detecting Malaria with Deep Learning which contains employee data for 1470 infected images and 1350 uninfected images employees with various information about the employees of effecting malaria. It is collected from Kaggle[1]. We used this dataset to predictwhen employees are going to quit by understanding the main drivers of employee chrun. Firstly we need to import the dataset source file for analysis. The dataset consists of 1470 infected images and 1350 uninfected images. The dataset contains several numerical and categorical columns providingvarious information on employee’s personal and employement details.The data provided has no missing values. In HR Analytics, employee data is unlikely to feature large ratio of missing values as HR Departments typically have all personal and employement data on-file.

**3.3.1 BLOCK DIAGRAM:**

Convolutional Neural Networks, or CNNs, were designed to map image data to an output variable. They have proven so effective that they are the go-to method for any type of prediction problem involving image data as an input.

CNNs are used in variety of areas, including image and pattern recognition, speech recognition, natural language processing, and video analysis. There are a number of reasons that convolutional neural networks are becoming important. In traditional models for pattern recognition, feature extractors are hand designed.

One of the main parts of Neural Networks is Convolutional neural networks (CNN). They are made up of neurons with learnable weights and biases. Each specific neuron receives numerous inputs and then takes a weighted sum over them, where it passes it through an activation function and responds back with an output.

• Layers in CNN

• Input layer.

• Convo layer (Convo + ReLU)

• Pooling layer.

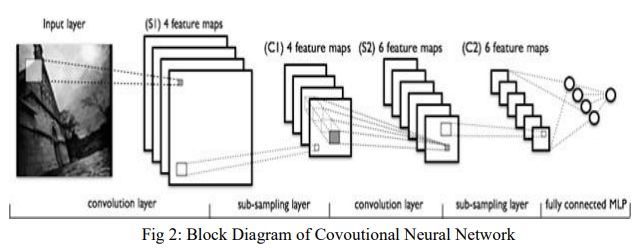
• Fully connected (FC) layer.

• Softmax/logistic layer.

• Output layer.

The term convolution refers to the mathematical combination of two functions to produce a third function. It merges two sets of information. In the case of a CNN, the convolution is performed on the input data with the use of a filter or kernel (these terms are used interchangeably) to then produce a feature map.

CNN is designed to automatically and adaptively learn spatial hierarchies of features through back propagation by using multiple building blocks, such as convolution layers, pooling layers, and fully connected layers.



ReLu refers to the Rectifier Unit, the most commonly deployed activation function for the outputs of the CNN neurons. Mathematically, it's described as: Unfortunately, the ReLu function is not differentiable at the origin, which makes it hard to use with back propagation training. There is a layer of input nodes, a layer of output nodes, and one or more intermediate layers.

The interior layers are sometimes called “hidden layers” because they are not directly observable from the systems inputs and outputs. Back propagation is the central mechanism by which neural networks learn. It is the messenger telling the network whether or not the net made a mistake when it made a prediction.

Forward propagation is when a data instance sends its signal through a network's parameters toward the prediction at the end. Classification neural networks used for feature categorization are very similar to fault-diagnosis networks, except that they only allow one output response for any input pattern, instead of allowing multiple faults to occur for a given set of operating conditions.

To prepare a thick blood film, a blood spot is stirred in a circular motion with the corner of the slide, taking care not make the preparation too thick, and allowed to dry without fixative. After drying, the spot is stained with diluted Giemsa (1 : 20, vol/vol) for 20 min, and washed by placing the film in buffered water for 3 min. The slide is allowed to air-dry in a vertical position and examination using a light microscope. As they are unfixed, the red cells lyse when a water-based stain is applied.

A thin blood film is prepared by immediately placing the smooth edge of a spreader slide in a drop of blood, adjusting the angle between slide and spreader to 45° and then smearing the blood with a swift and steady sweep along the surface. The film is then allowed to air-dry and is fixed with absolute methanol. After drying, the sample is stained with diluted Giemsa (1 : 20, vol/vol) for 20 min and washed by briefly dipping the slide in and out of a jar of buffered water (excessive washing will decolorize the film). The slide is then allowed to air-dry in a vertical position and examined under a light microscope. The wide acceptance of this technique by laboratories all around the world can be attributed to its simplicity, low cost, its ability to identify the presence of parasites, the infecting species, and assess parasite density-all parameters useful for the management of malaria.

Recently, a study showed that conventional malaria microscopic diagnosis at primary healthcare facilities in Tanzania could reduce the prescription of anti malarial drugs, and also appeared to improve the appropriate management of non-malarial fevers. However, the staining and interpretation processes are labor intensive, time consuming, and require considerable expertise and trained healthcare workers, particularly for identifying species accurately at low parasitemia or in mixed malarial infections.

The most important shortcoming of microscopic examination is its relatively low sensitivity, particularly at low parasite levels. Although the expert microscopist can detect up to 5 parasites/µl, the average microscopist detects only 50-100 parasites/µl. This has probably resulted in underestimating malaria infection rates, especially cases with low parasitemia and asymptomatic malaria.

**3.3.2 Malaria attacks in India:**

The major human malaria species in India are Plasmodium falciparum and P. vivax; P. malaria has been reported in the eastern India state of Orissa (Sharma et al., 2006), while P. ovale appears to be extremely rare if not absent. Malaria was nearly eradicated from India in the early 1960s but the disease has reemerged as a major public health problem. As a result, in 1976, 6.45 million cases were recorded by the National Malaria Eradication Programme (NMEP), highest since resurgence.

Some of the most common methods to prevent malaria are:

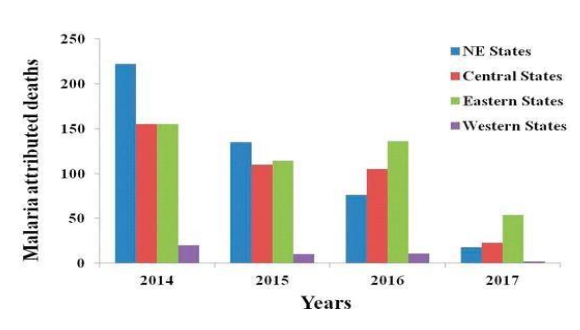
• Applying mosquito repellents.

• Always using mosquito nets over the bed.

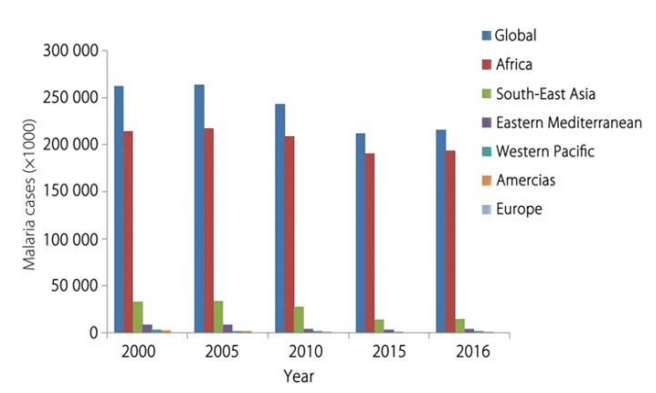
• Wear long sleeves clothes that cover your arms and legs completely.

• Screen your doors and windows, especially during the evenings.

• opt for loose-fitted clothes instead of tight ones



**Malaria cases across the world:**



**3.3.3: FLOWCHART**

**Different Levels of Detecting Malaria:**

A flowchart is simply a graphical representation of steps. It shows steps in sequential order and is widely used in presenting the flow of algorithms, workflow or processes. Typically, a flowchart shows the steps as boxes of various kinds, and their order by connecting them with arrows.

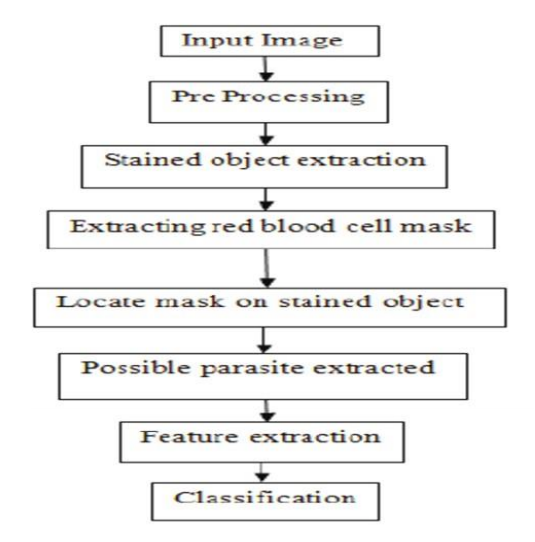
Image pre-processing is the name for operations on images at the lowest level of abstraction whose aim is an improvement of the image data that suppress undesired distortions or enhances some image features important for further processing. Image pre?processing use the redundancy in images. To make the process easier, data pre-processing is divided into four stages: data cleaning, data integration, data reduction, and data transformation.

The reason why a user transforms existing files into a new one is because of many reasons. Data pre-processing has the objective to add missing values, aggregate information, label data with categories (Data binning) and smooth a trajectory. Python becomes an apt choice for such Image processing tasks. This is due to its growing popularity as a scientific programming language and the free availability of many states of the art Image Processing tools in its ecosystem.

MATLAB is a scientific programming language and provides strong mathematical and numerical support for the implementation of advanced algorithms. It is for this reason that MATLAB is widely used by the image processing and computer vision community. By reading the image as a NumPy array, various image processing can be performed using NumPy functions. By the operation of NumPy array, you can get and set (change) pixel values, trim images, concatenate images, etc.

Classification refers to a predictive modeling problem where a class label is predicted for a given example of input data. Examples of classification problems include: Given an example, classify if it is spam or not.

Given a handwritten character, classify it as one of the known characters.

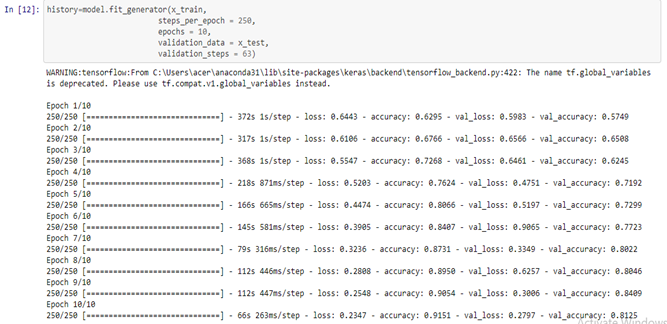


**CHAPTER 4**

**RESULTS AND DISCUSSION**

**Accuracy:**

The accuracy can be termed as closeness of measurements in statistical measures; however, it is also used in classifications. In classifications accuracy is the proportion of true results among the total number of cases. And to our model we got an accuracy of about 85% which indicates a good classification accuracy rate.



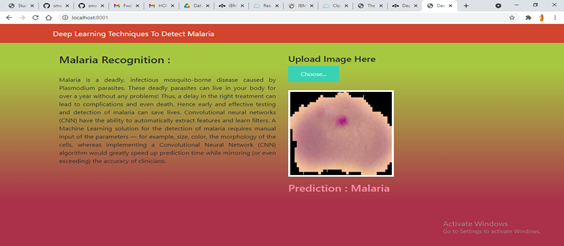
**RESULTS:**

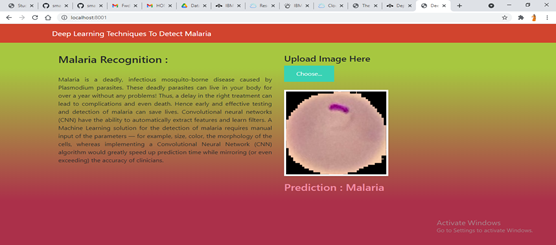


**case 1:** Malaria is a life-threatening disease caused by parasites that are transmitted to people through the bites of infected female Anopheles mosquitoes. It is preventable and curable. In 2019, there were an estimated 229 million cases of malaria worldwide. The estimated number of malaria deaths stood at 409 000 in 2019.

Expected Result is Malaria

Prediction is shown below:

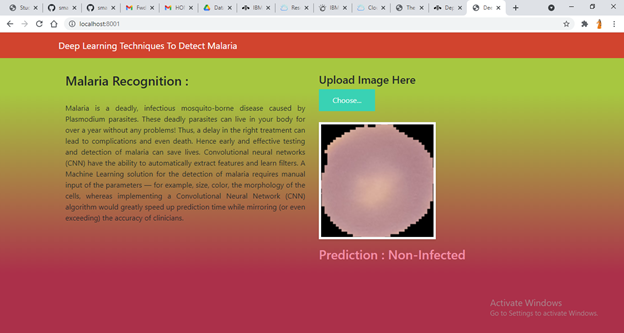


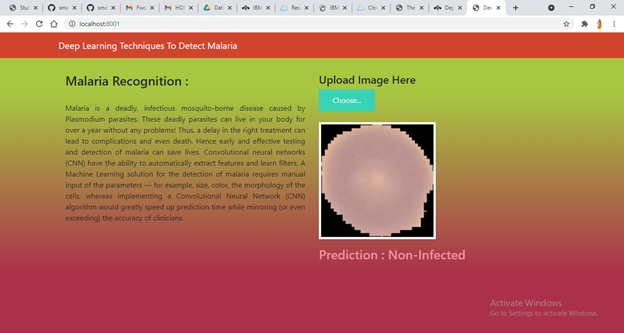


**case 2:** People receive medical treatment to maintain the balance, when necessary. Physical well-being involves pursuing a healthful lifestyle to decrease the risk of disease.

Expected Result is Non-Infected

Prediction is shown below:





**CHAPTER 5**

**CONCLUSION**

A major finding of the committee is the need to increase donor and public awareness of the growing risk presented by the resurgence of malaria. Overall, funding levels are not adequate to meet the problem. The committee believes that funding in the past focused too sharply on specific technologies and particular control strategies (e.g., indiscriminate use of insecticide spraying). Future support must be balanced among the needs outlined in this report. The issue for prioritization is not whether to select specific technologies or control strategies, but to raise the priority for solving the problem of malaria. This is best done by encouraging balanced research and control strategies and developing a mechanism for periodically adjusting support for promising approaches.

**REFERENCES** https://towardsdatascience.com/detecting-malaria-with-deep-learning?9e45c1e34b60 https://machinelearningmastery.com/inspirational-applications-deep-learning/ https://machinelearningmastery.com/deep-learning-with-python/ https://www.healthline.com/health/malaria#\_noHeaderPrefixedContent https://www.investopedia.com/terms/d/deeplearning.asp#:~:text=Deep%20learn ing%20is%20an%20AI,is%20both%20unstructured%20and%20unlabeled. https://machinelearningmastery.com/what-is-deep-learning/ https://machinelearningmastery.com/tutorial-first-neural-network-python-keras/ https://machinelearningmastery.com/convolutional-layers-for-deep-learningneural-networks/ https://www.osapublishing.org/oe/fulltext.cfm?uri=oe-25-15-17150&id=369137 https://www.sciencedirect.com/science/article/pii/S1877050918308019 https://www.manning.com/books/deep-learning-with-python https://www.tensorflow.org/resources/learn-ml https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2688806/

1. **Source Code**
2. **from keras.models import Sequential**
3. **from keras.layers import Dense**
4. **from keras.layers import Convolution2D**
5. **from keras.layers import MaxPooling2D**
6. **from keras.layers import Flatten**
7. **model=Sequential()**
8. **model.add(Convolution2D(32,(3,3),input\_shape=(64,64,3),activation="relu"))**
9. **model.add(MaxPooling2D(pool\_size = (2, 2)))**
10. **model.add(Flatten())**
11. **model.add(Dense(init="uniform",activation="relu",output\_dim=120))**
12. **model.add(Dense(init="uniform",activation="sigmoid",output\_dim=1))**
13. **model.compile(loss="binary\_crossentropy",optimizer="adam",metrics=["accuracy"])**
14. **from keras.preprocessing.image import ImageDataGenerator**
15. **train\_datagen = ImageDataGenerator(rescale = 1./255,**
16. **shear\_range = 0.2,**
17. **zoom\_range = 0.2,**
18. **horizontal\_flip = True)**
19. **test\_datagen = ImageDataGenerator(rescale = 1./255)**
20. **x\_train = train\_datagen.flow\_from\_directory(r'C:\Users\acer\Desktop\dataset\train\_dataset',**
21. **target\_size = (64, 64),**
22. **batch\_size = 32,**
23. **class\_mode = 'binary')**
24. **x\_test = test\_datagen.flow\_from\_directory(r'C:\Users\acer\Desktop\dataset\test\_dataset',**
25. **target\_size = (64, 64),**
26. **batch\_size = 32,**
27. **class\_mode = 'binary')**
28. **history=model.fit\_generator(x\_train,**
29. **steps\_per\_epoch = 250,**
30. **epochs = 10,**
31. **validation\_data = x\_test,**
32. **validation\_steps = 63)**
33. **model.save("mymodel.h5")**
34. **import numpy as np**
35. **from skimage.transform import resize**
36. **def detect(frame):**
37. **try**
38. **img = resize(frame,(64,64))**
39. **img = np.expand\_dims(img,axis=0)**
40. **if(np.max(img)>1):**
41. **img = img/25.0**
42. **prediction = model.predict(img)**
43. **print(prediction)**
44. **prediction = model.predict\_classes(img)**
45. **print(prediction)**
46. **except AttributeError:**
47. **print("shape not found")**
48. **import cv2**
49. **frame=cv2.imread(r" C:\Users\acer\Desktop\Malaria Application\Dataset\train\_dataset\Infected\C60P21thinF\_IMG\_20150804\_104919\_cell\_140.png")**
50. **data = detect(frame)**

1. **FLASK CODE:**
2. from \_\_future\_\_ import division, print\_function
3. from flask import Flask, redirect, url\_for, request, render\_template, jsonify
4. import json
5. from werkzeug.utils import secure\_filename
6. from keras.models import load\_model
7. from keras.preprocessing import image
8. import numpy as np
9. import sys
10. import os
11. app = Flask(\_\_name\_\_, static\_url\_path='')
12. def model\_predict(img\_path, model):
13. img = image.load\_img(img\_path, target\_size=(64, 64))
14. x = image.img\_to\_array(img)
15. x = np.expand\_dims(x, axis=0)
16. preds = model.predict\_classes(x)
17. return preds
18. @app.route('/', methods=['GET'])
19. def index():
20. return render\_template('index.html')
21. @app.route('/predict', methods=['GET', 'POST'])
22. def upload():
23. if request.method == 'POST':
24. f = request.files['image']
25. basepath = os.path.dirname(\_\_file\_\_)
26. file\_path = os.path.join(
27. basepath, 'uploads', secure\_filename(f.filename))
28. f.save(file\_path)
29. model = load\_model('mymodel.h5')
30. preds = model\_predict(file\_path, model)
31. print("preds : "+str(preds))
32. ls=["Malaira","Not-Infected"]
33. p=preds.flatten()
34. result = ls[p[0]]
35. print(result)
36. return result
37. return None
38. if \_\_name\_\_ == '\_\_main\_\_':
39. port = int(os.getenv('PORT', 8001))
40. app.run(host='0.0.0.0', port=port, debug=True)
41. **Templates**
42. **<!DOCTYPE html>**
43. **<html>**
44. **<head>**
45. **<meta charset="utf-8">**
46. **<meta http-equiv="X-UA-Compatible" content="IE=edge">**
47. **<meta name="viewport" content="width=device-width, initial-scale=1">**
48. **<title>Breast Cancer</title>**
49. **<!-- Bootstrap -->**
50. **<link href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css" rel="stylesheet">**
51. **<link href="styles.css" rel="stylesheet">**
52. **<!-- jQuery (necessary for Bootstrap's JavaScript plugins) -->**
53. **<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.4/jquery.min.js"></script>**
54. **<!-- Include all compiled plugins (below), or include individual files as needed -->**
55. **<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>**
56. **<script src="js/lib/jquery.i18n/jquery.i18n.js"></script>**
57. **<script src="js/lib/jquery.i18n/jquery.i18n.messagestore.js"></script>**
58. **<script src="js/lib/jquery.i18n/jquery.i18n.fallbacks.js"></script>**
59. **<script src="js/lib/jquery.i18n/jquery.i18n.language.js"></script>**
60. **<script src="js/lib/jquery.i18n/jquery.i18n.parser.js"></script>**
61. **<script src="js/lib/jquery.i18n/jquery.i18n.emitter.js"></script>**
62. **<script src="js/lib/jquery.i18n/jquery.i18n.emitter.bidi.js"></script>**
63. **<script src="antixss.js" type="text/javascript"></script>**
64. **<script>**
65. **$( document ).ready(function() {**
66. **$.i18n().load( {**
67. **en: {**
68. **"welcome": "Welcome.",**
69. **"name": "name",**
70. **"what\_is\_your\_name": "What is your name?",**
71. **"hello": "Hello $1",**
72. **"added\_to\_database": "Hello $1, I've added you to the database!",**
73. **"database\_contents": "Database contents: "**
74. **},**
75. **ja: {**
76. **"welcome": "ようこそ。",**
77. **"name": "名前",**
78. **"what\_is\_your\_name": "お名前を教えてください。",**
79. **"hello": "こんにちは $1",**
80. **"added\_to\_database": "こんにちは $1 さん、あなたをデータベースに追加しました。",**
81. **"database\_contents": "データベースの内容: "**
82. **}**
83. **} );**
84. **$('body').i18n();**
85. **$('#user\_name').attr("placeholder", $.i18n('name') );**
86. **});**
87. **</script>**
88. **</head>**
89. **<body>**
90. **<div class="container" id="container">**
91. **<h1 data-i18n="welcome"></h1> <!- Welcome ->**
92. **<div id="nameInput" class="input-group-lg center-block helloInput">**
93. **<p class="lead" data-i18n="what\_is\_your\_name"></p>**
94. **<input id="user\_name" type="text" class="form-control" aria-describedby="sizing-addon1" value="" />**
95. **</div>**
96. **<p id="response" class="lead text-center"></p>**
97. **<p id="databaseNames" class="lead text-center"></p>**
98. **</div>**
99. **<footer class="footer">**
100. **<div class="container">**
101. **<span><a href="https://console.bluemix.net/docs/tutorials/index.html" target="\_blank">Looking for more tutorials?</a></span>**
102. **</div>**
103. **</footer>**
104. **</body>**
105. **</html>**
106. **<script>**
107. **//Submit data when enter key is pressed**
108. **$('#user\_name').keydown(function(e) {**
109. **var name = $('#user\_name').val();**
110. **if (e.which == 13 && name.length > 0) { //catch Enter key**
111. **//POST request to API to create a new visitor entry in the database**
112. **$.ajax({**
113. **method: "POST",**
114. **url: "./api/visitors",**
115. **contentType: "application/json",**
116. **data: JSON.stringify({name: name })**
117. **})**
118. **.done(function(data) {**
119. **if(data && data.name){**
120. **if(data.\_id)**
121. **$('#response').html($.i18n('added\_to\_database', AntiXSS.sanitizeInput(data.name)));**
122. **else**
123. **$('#response').html($.i18n('hello', AntiXSS.sanitizeInput(data.name)));**
124. **}**
125. **else {**
126. **$('#response').html(AntiXSS.sanitizeInput(data));**
127. **}**
128. **$('#nameInput').hide();**
129. **getNames();**
130. **});**
131. **}**
132. **});**
133. **//Retrieve all the visitors from the database**
134. **function getNames(){**
135. **$.get("./api/visitors")**
136. **.done(function(data) {**
137. **if(data.length > 0) {**
138. **data.forEach(function(element, index) {**
139. **data[index] = AntiXSS.sanitizeInput(element)**
140. **});**
141. **$('#databaseNames').html($.i18n('database\_contents') + JSON.stringify(data));**
142. **}**
143. **});**
144. **}**
145. **//Call getNames on page load.**
146. **getNames();**
147. **</script>**

1. **Setup**
2. **"""**
3. **Hello World app for running Python apps on Bluemix**
4. **"""**
5. **# Always prefer setuptools over distutils**
6. **from setuptools import setup, find\_packages**
7. **# To use a consistent encoding**
8. **from codecs import open**
9. **from os import path**
10. **here = path.abspath(path.dirname(\_\_file\_\_))**
11. **# Get the long description from the README file**
12. **with open(path.join(here, 'README.md'), encoding='utf-8') as f:**
13. **long\_description = f.read()**
14. **setup(**
15. **name='python-hello-world-flask',**
16. **version='1.0.0',**
17. **description='Hello World app for running Python apps on Bluemix',**
18. **long\_description=long\_description,**
19. **url='https://github.com/IBM-Bluemix/python-hello-world-flask',**
20. **license='Apache-2.0'**
21. **)**

1. **JS**
2. /\*
3. \* Copyright IBM Corporation 2017
4. \*
5. \* Licensed under the Apache License, Version 2.0 (the "License");
6. \* you may not use this file except in compliance with the License.
7. \* You may obtain a copy of the License at
8. \*
9. \* http://www.apache.org/licenses/LICENSE-2.0
10. \*
11. \* Unless required by applicable law or agreed to in writing, software
12. \* distributed under the License is distributed on an "AS IS" BASIS,
13. \* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
14. \* See the License for the specific language governing permissions and
15. \* limitations under the License.
16. \*/
17. "use strict";
18. class AntiXSS {
19. static sanitizeInput(str) {
20. return String(str).replace(/&(?!amp;|lt;|gt;)/g, '&amp;').replace(/</g, '&lt;').replace(/>/g, '&gt;');
21. }
22. }
23. $(document).ready(function () {
24. // Init
25. $('.image-section').hide();
26. $('.loader').hide();
27. $('#result').hide();
28. // Upload Preview
29. function readURL(input) {
30. if (input.files && input.files[0]) {
31. var reader = new FileReader();
32. reader.onload = function (e) {
33. $('#imagePreview').css('background-image', 'url(' + e.target.result + ')');
34. $('#imagePreview').hide();
35. $('#imagePreview').fadeIn(650);
36. }
37. reader.readAsDataURL(input.files[0]);
38. }
39. }
40. $("#imageUpload").change(function () {
41. $('.image-section').show();
42. $('#btn-predict').show();
43. $('#result').text('');
44. $('#result').hide();
45. readURL(this);
46. });
47. // Predict
48. $('#btn-predict').click(function () {
49. var form\_data = new FormData($('#upload-file')[0]);
50. // Show loading animation
51. $(this).hide();
52. $('.loader').show();
53. // Make prediction by calling api /predict
54. $.ajax({
55. type: 'POST',
56. url: '/predict',
57. data: form\_data,
58. contentType: false,
59. cache: false,
60. processData: false,
61. async: true,
62. success: function (data) {
63. // Get and display the result
64. $('.loader').hide();
65. $('#result').fadeIn(600);
66. $('#result').text('Prediction : '+data);
67. console.log('Success!');
68. },
69. });
70. });
71. });