**A**

**PROJECT**

**REPORT ON**

**DOG BREED CLASSIFICATION**

**By Vivek Parmar (16CEUSG030) - CE 084**

**Hardik Parmar (16CEUSG003) - CE 081**

**B.Tech CE SEMESTER-VI**

**Subject: System Design Practice**

**Guided By:**

Professor M. S. Bhatt

Associate Professor

Dept. of Comp. Engg.



**Faculty of Technology**

**Dharmsinh Desai University, Nadiad**

**Department of Computer Engineering**

**DHARMSINH DESAI UNIVERSITY**

NADIAD 387001, GUJARAT



**Certificate**

This is to certify that the project carried out in the subject of System Design Practice titled “Dog Breed Classification” and recorded in this report is a work of

|  |  |  |
| --- | --- | --- |
| Vivek Parmar | CE – 084 | 16CEUSG030 |
| Hardik Parmar | CE – 081 | 16CEUSG003 |

**Department of Computer Engineering, semester VI. He was involved in Project developing during the period Dec-2018 to April- 2019.**

**Guide:**

Prof. Malay S. Bhatt Dr. C. K. Bhensdadia

Associate Professor Head, CE Dept

CE Dept, Faculty of Technology

Faculty of Computer Engineering ‘ Dharmsinh Desai University

Dharmsinh Desai University

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  | **TABLE OF CONTENTS** |  |
| **Chapter** |  | **Page** |
| **1.** | **Introduction** | **1** |
| **2.** | **About the System** | **2** |
|  | Overall description | 2 |
| **3.** | **Algorithm and its working** | **4** |
|  | Libraries and Technologies used |  |
| **4.** | **Implementation** | **11** |
|  | Modules and Brief Description | 11 |
| **5.** | **Test Case Design** | **12** |
| **6.** | **Conclusion and Future Extension** | **13** |
|  | **Bibliography** |  |

|  |  |  |
| --- | --- | --- |
|  | **List of Figures** |  |
|  |  |  |
| **figures** |  | **Page** |
| **1.** | **Perceptron** | **6** |
| **2.** | **Sigmoid function** | **7** |
| **3.** | **Gradient Descent** | **8** |
| **4.** | **Dataflow diagram** | **9** |
| **5.** | **Layout [ Initial Page]** | **13** |
| **6.** | **Layout [predicted breed output]** | **13** |
|  |  |  |

**Chapter 1**

Introduction



This system which is named Dog Detection, is a research and analysis project. As the name suggests, the system let user know the particular breed of the dog from the image entered by user. Given an image of a dog, our algorithm will identify an estimate of the dog’s breed. This project is made with the use of PyTorch deep learning library.

**Chapter 2**

About the System



**2.1** **Overall Description**

**2.1.1** **Product Perspective**

Dog breed detection system was developed for knowing the certain breed of dog by creating deep learning neural networks. Any user can use it. It is a free and open source application.

**2.1.2** **Product Functions**

* Input Image: Users can give image to application for further breed detection.

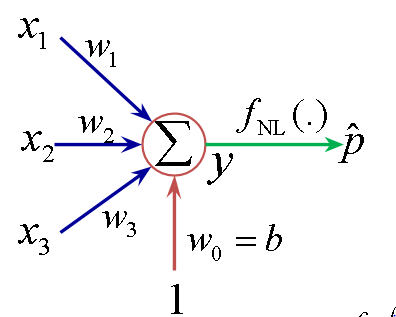
**Chapter 3**

Algorithm and its working



**Perceptron:**

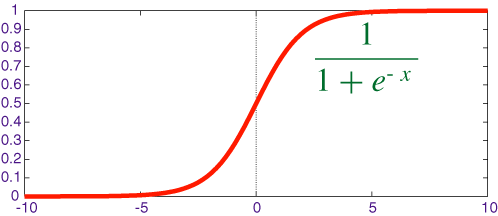
The architecture and behavior of a perceptron is very similar to biological neurons, and is often considered as the most basic form of neural network. Other kinds of neural networks were developed after the perceptron, and their diversity and applications continue to grow. It is easier to explain the constitutes of a neural network using the example of a single layer perceptron.



A single layer perceptron works as a linear binary classifier. Consider a feature vector [x1, x2, x3] that is used to predict the probability (p) of occurrence of a certain event.

**Activation function:**

The result of the summation function, that is the weighted sum, is transformed to a desired output by employing a nonlinear function (fNL), also known as activation function. Since the desired output is probability of an event in this case, a sigmoid function can be used to restrict the results (y) between 0 and 1.

****

**[sigmoid function]**

**Error and Loss Function:**

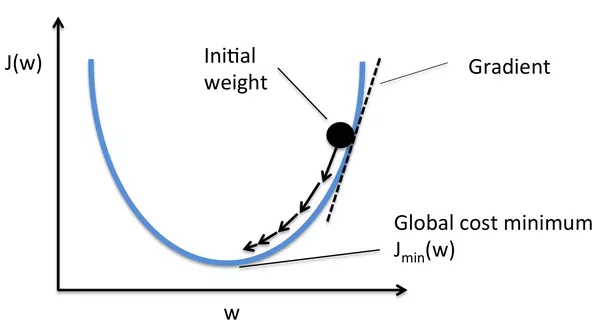
In most learning networks, error is calculated as the difference between the actual output and the predicted output.

https://cdn-images-1.medium.com/max/1000/1*KeDofaLByCl99Z8PpVHgiw.gif

The function that is used to compute this error is known as Loss Function J(.). Different loss functions will give different errors for the same prediction, and thus have a considerable effect on the performance of the model. One of the most widely used loss function is mean square error, which calculates the square of difference between actual value and predicted value. Different loss functions are used to deal with different type of tasks, i.e. regression and classification.

**Back Propogation and Optimisation Function:**

Error J(w) is a function of internal parameters of model i.e. weights and bias. For accurate predictions, one needs to minimize the calculated error. In a neural network, this is done using back propagation. The current error is typically propagated backwards to a previous layer, where it is used to modify the weights and bias in such a way that the error is minimized. The weights are modified using a function called Optimization Function.



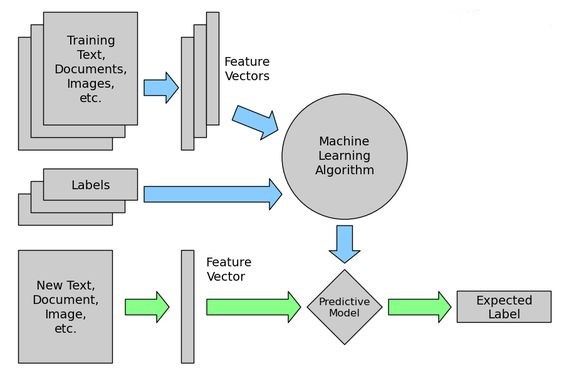
Optimisation functions usually calculate the **gradient** i.e. the partial derivative of loss function with respect to weights, and the weights are modified in the opposite direction of the calculated gradient. This cycle is repeated until we reach the minima of loss function.

Thus, the components of a neural network model i.e. the activation function, loss function and optimization algorithm play a very important role in efficiently and effectively training a Model and produce accurate results. Different tasks require a different set of such functions to give the most optimum results.

**Gradient Descent**

Calculates gradient for the whole dataset and updates values in direction opposite to the gradients until we find a local minima. Stochastic Gradient Descent performs a parameter update for each training example unlike normal Gradient Descent which performs only one update. Thus, it is much faster. Gradient Decent algorithms can further be improved by tuning important parameters like momentum, learning rate.

**DATA FLOW DIAGRAM**



**Chapter 4**

Implementation



**4.1 Modules and brief description**

* **Import dataset**

We have a dataset of dogs consisting total 133 breeds each having more than 50 images.

* **Create a CNN**

We use VGG-16 model that has been trained on ImageNet. We divide our dataset for three data loaders for the training, validation and testing. We have center crop the image to make it 224\*224 size. We have used reLU activation function, loss function and optimizer in order to create CNN.

* **Train and validate model**

We have used training and validation dataset for model and give stochastic gradient descent to optimize the model.

* **Image Selector**

Image selector is the windows form-based file selector module where user will browse the image to send or user will give the image paths to identify dog breeds. This module is consisting a browse button, image box and and a label to print the predicted breed.

**4.2 Technologies**

* **PythonNet**

It is a nuget package having the purpose to give programmers nearly seamless integration with .net clr and provides a powerful application scripting too for developers. It allows python code to interact with CLR and also used to embed Python into a .NET application.

* **ASP.NET**

To build user interface and to give image as an input.

**Chapter 5**

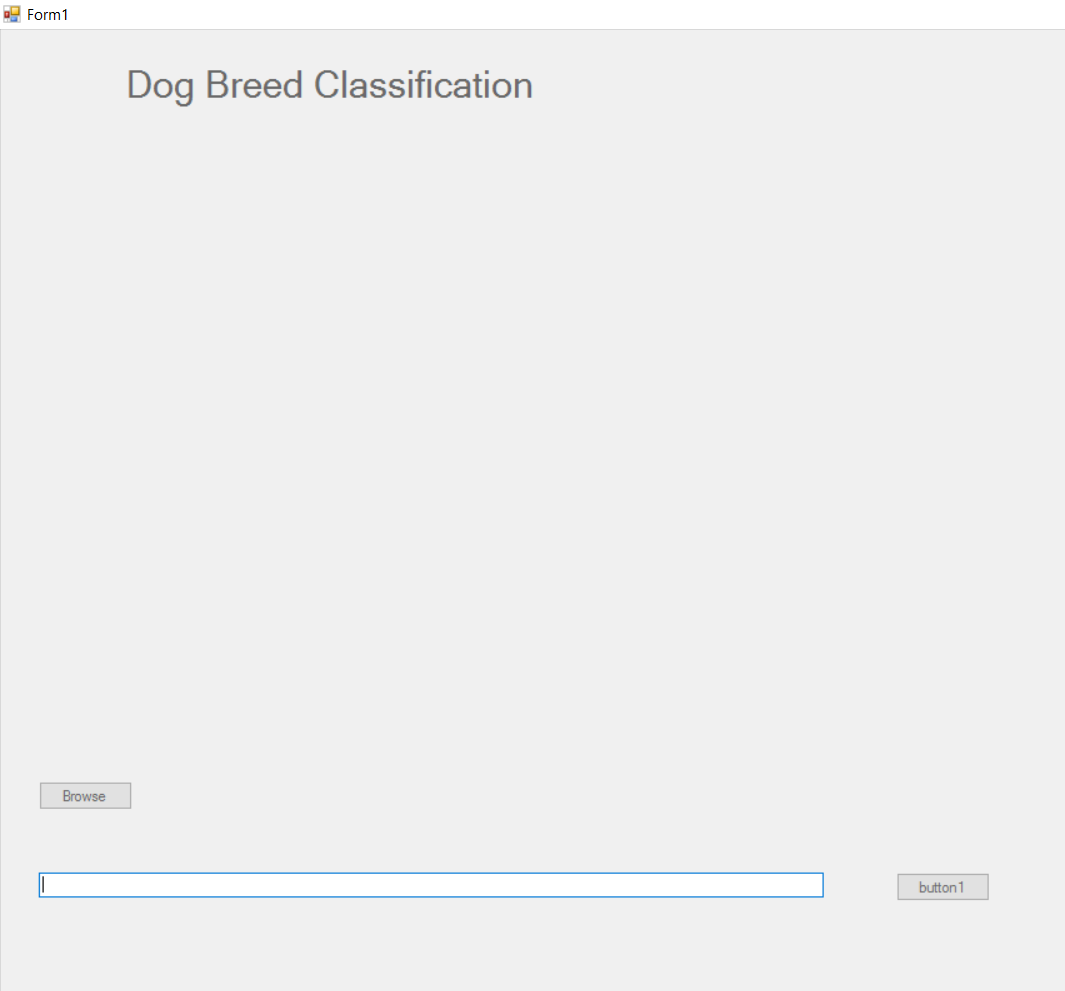
Test Case Design



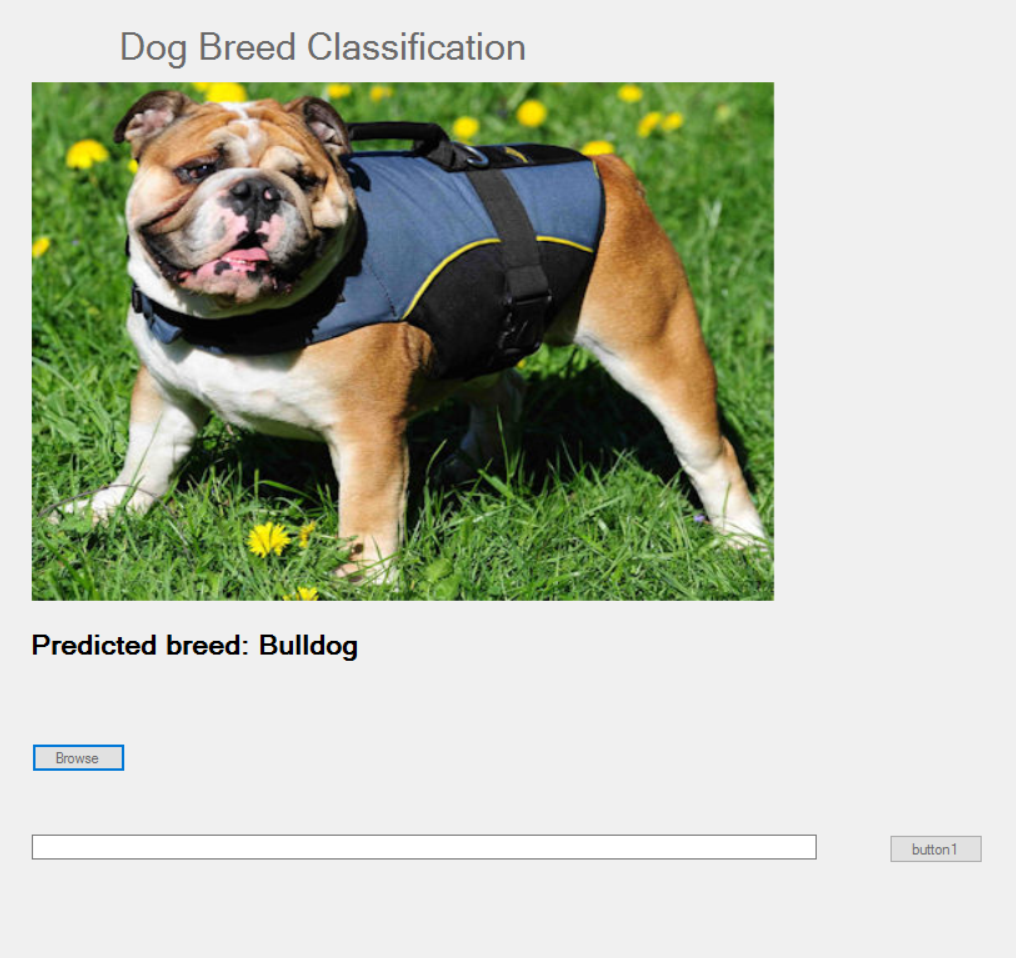
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Module** | **Test Case** | **Expected** | **Actual** | **Result** |
|  | **Input** | **Output** | **Output** |  |
|  |  |  |  |  |
| Detection | Image from Image loader | Dog’s predicted  Breed name  Along with its  image | Predicted breed name along with its image | Pass |
|  |  |  |  |  |
|  |  |  |  |  |
| Image Loader | Provide a path of  Image from local machine | Dog’s predicted  Breed name along with its image | Message displayed “image loaded” | Pass |
|  |  |  |  |  |
|  |  |  |  |  |

**LAYOUTS:**

1. Initial Page



**2**.Predicted breed(output)



**Chapter 7**

Conclusion and Future Extensions



**Conclusion**

* In development processes of this project, in first step all the requirements and user stories are gathered. After that technologies to use were decided. Then actual implementation of the project was collaboratively conducted. After implementation of major functions, user interface was improved. And at end the testing of modules was done. We’re able to get 81.56 accuracy on testing dataset of dog breed.

|  |  |  |  |
| --- | --- | --- | --- |
| **Dataset Type** | **Total Images** | **Prediction** | **Highest Predicted Breed** |
| Testing | 1064 | 40.34 % (Lowest) | Collie |
| Testing | 1064 | 81.56 % (Highest) | Beagle |

**Future Extensions**

* Future version of this application will be deployed on Azure or AWS cloud
* Also, the next version may include vast variety of new dog breeds which are not part of the current dataset.
* Our application’s next version will cover major other animals also.

Other functionality to be added in this application is a mobile app consisting real time lens which uses this model and will predict breeds.

**Bibliography**

[1] “Udacity”

Link: <https://classroom.udacity.com/courses/ud188>

[2]” PyTorch”

Link: <https://pytorch.org/>

[3] “Azure”

Link: <https://portal.azure.com/>

[4] “NVidia”

Link: <https://developer.nvidia.com/cuda-zone>

[5]” VGG16”

Link: <https://www.kaggle.com/keras/vgg16>

[6] “GitHub”

Link: <https://github.com/udacity/deep-learning-v2-pytorch>