MID TERM REPORT

On

Hand Gesture Recognition

Submitted by

Vanshika Shivani
Roll No: 171500373
Harshit Rai
Roll No: 171500127
Keshavi Aggarwal
Roll No: 171500160

Department of Computer Engineering & Applications

Institute of Engineering & Technology

GLA University
Mathura- 281406, INDIA
2019

Summary of the Project Work

This project entitled Hand Gesture Recognition was completed successfully. The system has been developed with much care and free of errors and at the same time it is efficient and less time consuming.

The purpose of this project was to study the Convolutional Neural Network and with the help of that identify different fashion items.

This project helped us gaining valuable information and practical knowledge on several objects like creating CNN models and Deep Learning using Python, TensorFlow and Keras, usage of functions of keras. Also the project helped us to understand about the development phases of the project and the software development lifecycle of the project. We learnt to test different features of the project.

This project has given us great satisfaction in building a model which can be implemented by anyone by simple modifications.

There is a scope of further development in our project that is we can add more hand gestures.

ACKNOWLEDGEMENT

The project work in this report is an outcome of continuous work over a period and drew intellectual support from various sources. I would like to articulate our profound gratitude and to all those people who extended their wholehearted co-operation and have helped me in completing this project successfully.

I am thankful to Mr. Mandeep Singh for teaching and assisting me in making the project successful. I would also like to thank our parents & other fellow mates for guiding and encouraging me throughout the duration of the project.

Department of computer Engineering and Applications

GLA University, Mathura

17 km. Stone NH#2, Mathura-Delhi Road, P.O. – Chaumuhan, Mathura – 281406

Declaration

I hereby declare that the work which is being presented in this project of "**Hand Gesture Recognition",** in partial fulfillment of the requirements for Python developer is an authentic record of my own work carried under the supervision of "Mr. Mandeep Singh".

Signature of Candidate:

Name of Candidate: Vanshika Shivani

Roll. No.: 171500373

Name of Candidate: Harshit Rai

Roll. No.: 171500127

Name of Candidate: Keshavi Aggarwal

Roll. No.: 171500160

Course: B-Tech

Year: IIIrd

Semester: Vth

ABSTRACT

Mini Project is the requirement for all engineering students in order to complete their Bachelor of Engineering degree at the GLA University, Mathura. Mini Project is a very important program, since it complements both the academic and professional aspects of the engineering education. Exposing the students to the practical experience and actual working environment shall open the avenues for developing their skills and capabilities, as well as enhancing their intellectual and emotional persona. The Mini Project also can provide strong linkages between university-industries that shall pave opportunities for "smart partnerships" and industrially driven research. The outcomes of the EIT that are mainly based on the assessment covering the company's and university's evaluation will provide the feedback for student's performance after 75% completion of their engineering study. The remarks from the companies on the students will very much helpful for the university to have a continuous quality improvement especially on curriculum practiced.

TABLE OF CONTENTS

INTRODUCTION TO DEEP LEARNING

Deep learning is a branch of machine learning which is completely based on artificial neural networks, as neural network is going to mimic the human brain so deep learning is also a kind of mimic of human brain. In deep learning, we don't need to explicitly program everything. The concept of deep learning is not new. It has been around for a couple of years now. It's on hype nowadays because earlier we did not have that much processing power and a lot of data. As in the last 20 years, the processing power increases exponentially, deep learning and machine learning came in the picture.

A formal definition of deep learning is- neurons, in human brain approximately 100 billion neurons all together this is a picture of an individual neuron and each neuron is connected through thousands of their neighbors.

The question here is how we recreate these neurons in a computer. So, we create an artificial structure called an artificial neural net where we have nodes or neurons. We have some neurons for input value and some for output value and in between, there may be lots of neurons interconnected in the hidden layer.

Installation of Jupyter Notebook on window

Instructions tested with Windows 10 64-bit and Continuum's Anaconda 5.20

Install GNU on Windows

1.Download and install GNU on Windows (Gow) from

https://github.com/bmatzelle/gow/releases/download/v0.8.0/Gow-0.8.0.exe.select The default options when prompted during the installation of Gow.

Install Anaconda and Jupyter Notebook

1.Downloads and install Anaconda from https://repo.anaconda.com/archive/Anaconda3-2019.07-Windows-x86_64.exe.

Select the default options when prompted during the installation of Anaconda

1. Open "Anaconda Prompt" by finding it in the window (start) menu.

2. Type the command in red to verified Anaconda was installed.

>python --version

Python 3.7.3

3. Type the command in red to update Anaconda >conda update --all --yes

Start Jupyter Notebook

1. Type the command in red to start jupyter Notebook

> jupyter notebook

SYSTEM AND TOOLS

Operating System

The programming work was carried out on one computer which ran the window 10 system.

The model was tested on two computers which ran window 10 and Ubuntu.

Python

Python is a widely used general-purpose, high level programming language. It was initially designed by Guido van Rossum in 1991 and developed by Python Software Foundation. It was mainly developed for emphasis on code readability, and its syntax allows programmers to express concepts in fewer lines of code.

Python is a programming language that lets you work quickly and integrate systems more efficiently.

There are two major Python versions- Python 2 and Python 3. Both are quite different.

Beginning with Python programming:

1) Finding an Interpreter:

Before we start Python programming, we need to have an interpreter to interpret and run our programs. There are certain online interpreters like https://ide.geeksforgeeks.org/, http://ideone.com/ or http://codepad.org/ that can be used to start Python without installing an interpreter.

Windows: There are many interpreters available freely to run Python scripts like IDLE (Integrated Development Environment) which is installed when you install the python software from.

Jupyter Notebook

The Jupyter Notebook is an interactive computing environment that enables users to author notebook documents that include: - Live code - Interactive widgets - Plots - Narrative text Equations - Images - Video

These documents provide a complete and self-contained record of a computation that can be converted to various formats and shared with others using email, Drop box, version control systems (like git/GitHub) or nbviewer.jupyter.org.

Keras Module

What is Keras?

Keras is an Open Source Neural Network library written in Python that runs on top of Theano or

Tensorflow. It is designed to be modular, fast and easy to use. It was developed by François Chollet, a Google engineer.

Keras doesn't handle low-level computation. Instead, it uses another library to do it, called the "Backend. So Keras is high-level API wrapper for the low-level API, capable of running on top of TensorFlow, CNTK, or Theano.

Keras High-Level API handles the way we make models, defining layers, or set up multiple input-output models. In this level, Keras also compiles our model with loss and optimizer functions, training process with fit function. Keras doesn't handle Low-Level API such as making the computational graph, making tensors or other variables because it has been handled by the "backend" engine.

What is Backend?

Backend is a term in Keras that performs all low-level computation such as tensor products, convolutions and many other things with the help of other libraries such as Tensorflow or Theano. So, the "backend engine" will perform the computation and development of the models. Tensorflow is the default "backend engine" but we can change it in the configuration.

Theano is an open source project that was developed by the MILA group at the University of Montreal, Quebec, Canada. It was the first widely used Framework. It is a Python library that helps in multidimensional arrays for mathematical operations using Numpy or Scipy. Theano can use GPUs for faster computation, it also can automatically build symbolic graphs for computing gradients. On its website, Theano claims that it can recognize numerically unstable expressions and compute them with more stable algorithms, this is very useful for our unstable expressions.

On the other hand, Tensorflow is the rising star in deep learning framework. Developed by Google's Brain team it is the most popular deep learning tool. With a lot of features, and researchers contribute to help develop this framework for deep learning purposes.

Advantages of Keras

Fast Deployment and Easy to understand

Keras is very quick to make a network model. If you want to make a simple network model with a few lines, Keras can help you with that. Look at the example below:

```
from keras.models import
Sequential from keras.layers import
Dense, Activation

model = Sequential() model.add(Dense(64, activation='relu',
input_dim=50)) #input shape of 50 model.add(Dense(28,
activation='relu')) #input shape of 50

model.add(Dense(10, activation='softmax'))
```

Because of friendly the API, we can easily understand the process. Writing the code with a simple function and no need to set multiple parameters.

Large Community Support:

There are lots of AI communities that use Keras for their Deep Learning framework. Many of them publish their codes as well tutorial to the general public.

Have multiple Backends

You can choose Tensorflow, CNTK, and Theano as your backend with Keras. You can choose a different backend for different projects depending on your needs. Each backend has its own unique advantage.

Cross-Platform and Easy Model Deployment

With a variety of supported devices and platforms, you can deploy Keras on any device like iOS with CoreML
Android with Tensorflow Android,

Web browser with .js support

Cloud engine

Raspberry Pi

Multi GPUs Support

You can train Keras with on a single GPU or use multiple GPUs at once. Because Keras has a built-in support for data parallelism so it can process large volumes of data and speed up the time needed to train it.

Introduction to Convolutional neural network

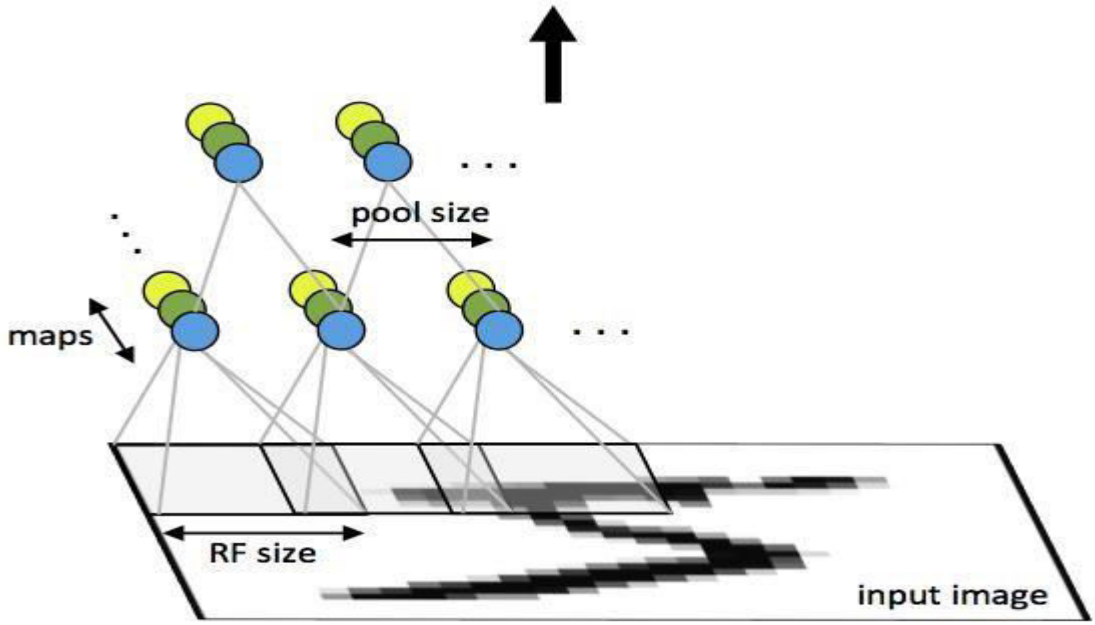
Overview

A Convolutional Neural Network (CNN) is comprised of one or more convolutional layers (often with a subsampling step) and then followed by one or more fully connected layers as in a standard multilayer neural network. The architecture of a CNN is designed to take advantage of the 2D structure of an input image (or other 2D input such as a speech signal). This is achieved with local connections and tied weights followed by some form of pooling which results in translation invariant features. Another benefit of CNNs is that they are easier to train and have many fewer parameters than fully connected networks with the same number of hidden units. In this article we will discuss the architecture of a CNN and the back propagation algorithm to compute the gradient with respect to the parameters of the model in order to use gradient based optimization. See the respective tutorials on convolution and pooling for more details on those specific operations.

Architecture

A CNN consists of a number of convolutional and subsampling layers optionally followed by fully connected layers. The input to a convolutional layer is a m x m x r image where m is the height and width of the image and r is the number of channels, e.g. an RGB image has r=3. The convolutional layer will have k filters (or kernels) of size n x n x q where n is smaller than the dimension of the image and q can either be the same as the number of channels r or smaller and may vary for each kernel. The size of the filters gives rise to the locally connected structure which are each convolved with the image to produce k feature maps of size m−n+1. Each map is then subsampled typically with mean or max pooling over p x p contiguous regions where p ranges between 2 for small images (e.g. MNIST) and is usually not more than 5 for larger inputs. Either before or after the subsampling layer an additive bias and sigmoidal nonlinearity is applied to each feature map. The figure below illustrates a full layer in a CNN

consisting of convolutional and subsampling sublayers. Units of the same color have tied weights.



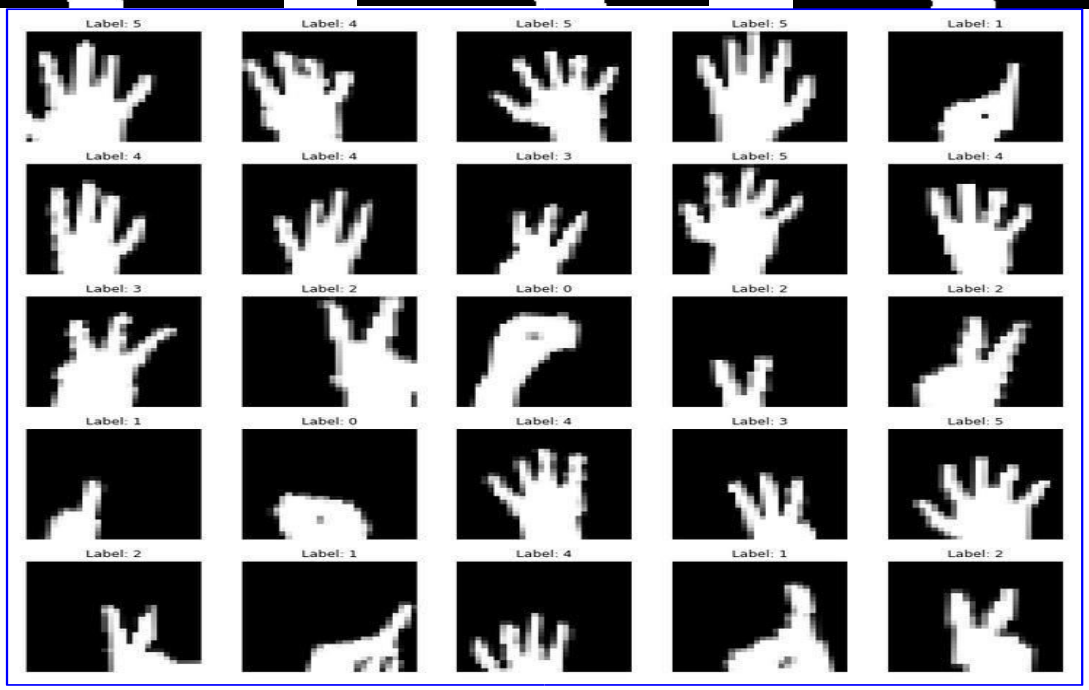
First layer of a convolutional neural network with pooling. Units of the same color have tied weights and units of different color represent different filter maps.

After the convolutional layers there may be any number of fully connected layers. The densely connected layers are identical to the layers in a standard multilayer neural network.

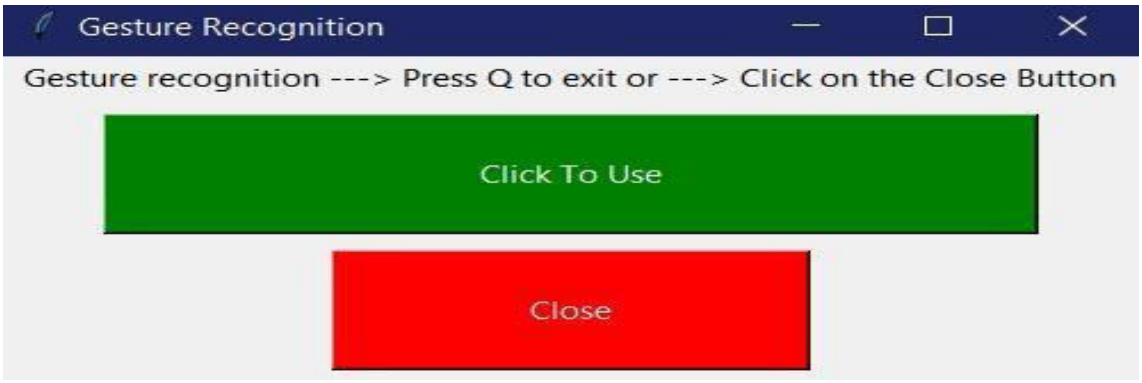
Introduction To Hand Gesture Dataset

Hand gesture Dataset which is used in this project is created by us and images—consisting of a training set of 1800 examples. Each example is a 50x50 grayscale image, associated with a label from 5 classes. I used this dataset to serve as a direct drop-in replacement for the original traditional dataset for benchmarking machine learning algorithms. It shares the same image size and structure of training and testing splits.

Here's an example how the data looks (each class takes three-rows):



Graphical User Interface of Project


Gesture Recognition
Gesture recognition ---> Press Q to exit or ---> Click on the Close Button
Click To Use
Close

REMAINING WORK

Approximately 40% of the work is done and 60% is still left which will be completed in 3-4 weeks.
The virtual mouse part of the project is left and its GUI part too.
It is will be followed by testing the algorithm and optimizing it further if needed to improve accuracy.

REFRENCES

https://github.com/zalandoresearch/fashion-mnist#why-we-made-fashion-mnist

https://www.pyimagesearch.com/2019/02/11/fashion-mnist-with-keras-and-deeplearning/

https://research.zalando.com/welcome/mission/research-projects/fashionmnist/