Error Detecting Pen

Prateek Bansal[1], Ritvik Singhal[2]

[1]Electronics and Communication Engineering, Jaypee Institute of Information Technology, NOIDA, India
b.prateek25[at]gmail.com

[2]Computer Science Engineering, Jaypee Institute of Information Technology, NOIDA, India
singhal.ritvik[at]gmail.com

Abstract: ***In this era where everything is based on touch technology, people prefer touch based computing rather than writing by hands on paper using a pen. Smart electronic gadgets of modern era such as tablets, laptops and smartphones have become handy. People prefer typing over handwriting as it is very user friendly and a lot of smart inbuilt features such an auto-correction which can help them fix their spellingmistakes instantly. But what if this feature of error detection comes in your pen? A pen you can write with without any spellingerrors. In this paper, we have introduced an idea where we have developed a pen that will check the spellings of the hand written text and minimize all the spelling error while writing.***

Keywords: Arduino, Smart pen, MPU6050, NRF24LO1, Dyslexia, Python, Data modeling

1. Introduction

In the age of digital communication, where short messaging, texting and chatting platforms are changing the writing style, only a few care about knowing the correct spelling of words.

Teaching grammar is not currently part of official language courses and only a few care about knowing the correct spelling of the words they use in their daily composition. If a person misspelled a word, dictionaries are computerized and the computer gives the instruction. When you use pen-and-paper and set up writing by hand, such a feature is not accessible.

This might be viewed as an issue and may not be viewed as an issue, as well. In the event that it is an issue, at that point do we have an advanced method of inciting an essayist in the event that a person has erroneously spelled a word in their manually written bit of work.

In this Digital age individuals will in general be easygoing and excessively careless about utilizing the English language. All the more significantly, formal correspondence turns into an issue on the off chance that one can't spell words effectively and accessing advanced methods for correspondence by school-going youngsters irritates thepower of this problem. Some of the time is irritating, yet honestly computer's help has saved billions of typographical blushes. Have you ever thought of a pen that could assist you with revising mistakes while writing? Our main vision behind this project was to learn and try to make something that can be of assistance to others. We can later on introduce technology like speech recognition so that the device becomes more practically usable. We can implement this by using speech recognition sensors like grove or robodo and then we can work on this using speech recognition API.

2. Problem Statement

We precisely aim at making a Digi-pen (as we have named it) which will be an advancement of the normal pen. A pen that would enable user to verify correctness of each and every word written by him/her and also notify user in case of any error detectedin the spelling of the word via LED lights.

3. Hardware and Software Used

A. Softwares

Python IDE: Python is a high-level programming language and is open source that is, it is free to use.

Arduino IDE: The Arduino IDE is an open source application available for Windows/MacOS/Linux. It is used to write and upload programs to the Arduino board microcontroller. It has a compiler for compiling the program and creates a hex file ready to be loaded into Arduino board. The coding of Arduino UNO is done in Arduino IDE inEmbedded C language

Anaconda IDE: Anaconda is a free and open source distribution of Python and R resources for logical subscriptions (data science, AI applications, big data preparation, advanced testing and so on.), Which means improving the package and organization. Package types are controlled by the anaconda package management system. Anaconda distribution includes heaps of logical scientific data for Windows, Linux, and MacOS. there are many python libraries with standard language editing libraries just as with various external libraries such as numpy, pandas, scikit-read.

B. Hardware

Arduino UNO: Arduino UNO (as shown in Fig. 1) is a micro-controller based on ATmega328P and belongs to 8 bit Alf and Vegard's RISC processorfamily. The board is equipped with a 16MHz quartz crystal and thereforeworks at 16 MHz clock speed. In total, there are 14 input and output pins. Reads data from NRF module on the system side and send the received data to the system for possessing.

Arduino NANO: The Arduino [10]Nano is a small, full and breadboard-adjusted board based on the ATmega328 (Arduino [10] Nano 3.0) or ATmega168 (Arduino [10] Nano xx). It has a great deal of functionality similar to Arduino [10] Dumilanov, and is still in the optional bundle. It requires a DC power jack, and works with the mini-B USB cable instead of the standard ones.Processing the data from MPU6050 and sends it to the NRF module.

MPU6050 Gyroscope Module: MPU6050 [6] is a micro-electro-mechanical systems (MEMS) equipped with a 3-axis accelerometer and a 3-axis gyroscope inside it. This encourages us to consider the acceleration, velocity, orientation, displacement and many other motion parameters of a frame or item. Similarly there is a digital motion processor inside it (DMP) which is unreliable to create complex predictions and consequently slows down the work of microcontrollers.Traces the coordinates of the movement of the pen.

NRF24LO1 Module: Nrf24l01Solitary Chip RF nrf24l01is a wireless handset module, which means that each module can send data as well. They operate on 2.4GHz frequency, which falls within the ISM band, and as a result is valid for use in practically all countries for building applications. While working modules can be separated by 100 meters (200 feet), this is an unusual decision for all wireless remote controlled ventures like ours. Sends the data serially to the other NRF module connected to the system.

Figure 1: Arduino Uno microcontroller

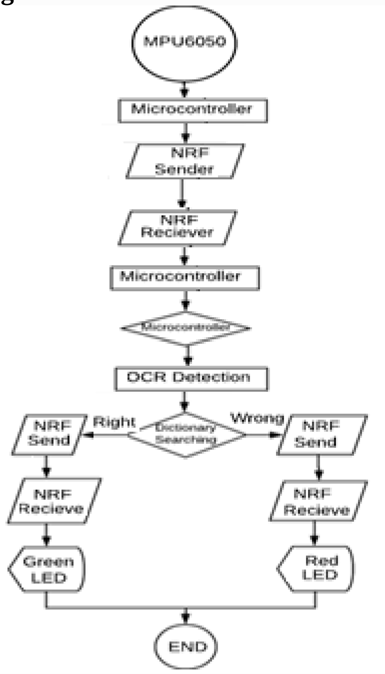

Figure 2: Flowchart diagram of system

4. Implementation

A. System Architecture

The flowchart diagram of the system is shown in Fig. 2and consists of a micro-controller interfaced with MPU6050 and NRF module.

The output from MPU determines the direction in which the user is writingand is serially transmitted from the pen to the laptop via microcontroller's and NRF modules on both sides. Further, the written text is converted to string and then searched in a virtual dictionary using pythons library and finally the result of it is send back to the pen which indicated the user the correctness of the written word via LED lights on the pen.

B. System Working

Converting Hand Gesture to Text in Png Format

Initial steps include making all the connections starting with connecting MPU6050 with Arduino. Functions used in accelerometer sketch are mpu.readRawAccel() which is used for the decimal equivalent of the 16-bit values of the acceleration in the X, Y and Z directions but This does not provide the acceleration values in g units, for that we'll use mpu.readNormalizeAccel() which gives the values of the acceleration in the X, Y and Z directions in g units and also takes into consideration the range setting chosen for the accelerometer and the corresponding sensitivity. Using this it gives acceleration in g units which can vary from 0 to the rangechosen and then finally we use mpu.begin(gyro_scale,accelo_range) to set the range of the accelerometer(Shown in Table I) and the scale of the gyroscope. After this connection the connection between NRF24l01 with Arduino will be made.

Now the data will be send to Arduino IDE serially and can be seen on the serial monitor of Arduino IDE.(Shown in Fig. 3). After sending the data to the Arduino we will take 30 readings of the initially positioning of the pen and then take an average of those reading to make an accurate estimate of the pens initial position.(shown in Fig. 4). Having the pen calibrated and after setting ranges for the movement for left, right, up and down(shown in Fig. 5), now we'll move the pen to actually write a word. This word will now show on the screen and a screenshot of this written word will be taken for us to work on it further.(shown in Fig. 5)

Figure 3: Process of serially sending the data

International Journal of Science and Research (IJSR)
ISSN: 2319-7064
ResearchGate Impact Factor (2018): 0.28 | SJIF (2019): 7.583

Figure 4: Calibrating to determine the initial position of the device

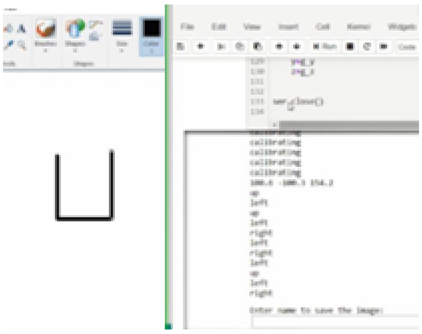

Figure 5: Drew the letter 'U' and asked to enter a name so that an image file is generated with that name.

Table I: Location and Direction of Movement

Left Greater than 1600	Up Greater than 1200	Down Greater than 1400	Right Greater than 700

Converting Image to Text String

Hand-written recognition or ID framework is recognized into four unique stages; preprocessing, feature extraction, classification and post processing. Data is retrieved from the tablet and it sets up stroke functions with event, time, power and speed. Data planning prior to the construction of parts of the stroke is heavily burdened with online character variation, contrary evidence identifying offline physical identity. Speed and force data are helpful for personality analysis in the form of independent analysis and so on. Given. Strokes are evaluated between up and down occasions during pencil development. Such raw data strokes have many breaks and manipulations and huge amounts of data that cannot be handled specifically by human information. Therefore, preprocessing and example extraction can be achieved against crude stroke. A block diagram model for the same is represented below(shown in Fig.6) along with the flowchart of the entire process(shown in Fig. 7)

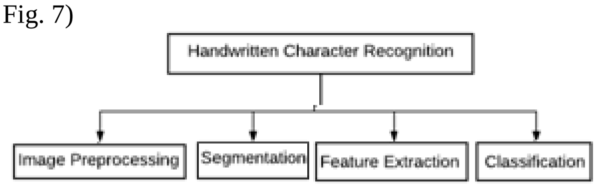

Figure 6: Model Breakdown

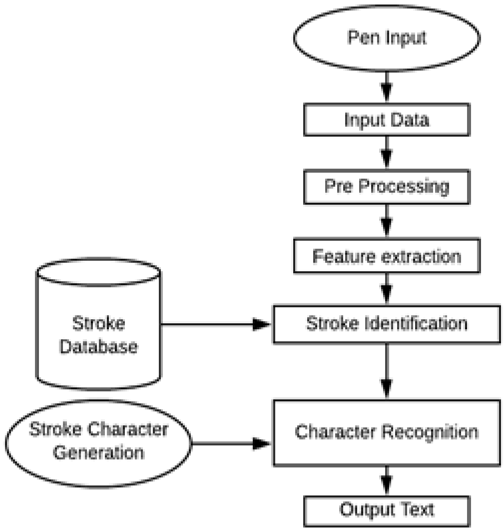

Figure 7: Sample flowchart of processes

Preprocessing

Preprocessing is known as an essential step of character recognition and it is typically known as a critical stage for an extraordinary recognition rate.

The main purpose of preprocessing in any framework configuration is to validate information and exclude variations such as noise and so on, because in this type of view, identification rates are reduced. For character recognition, the framework of the preprocessing stage is raw data and the output is standard hunger-free data.

Some basic techniques used during preprocessing are generalization, filtering, distortion detection, noise removal, distortion correction, and so on. If there is no noise reduction during preprocessing, this may indicate poor separation, durability. Since the detection rate is high. Less. In addition, various preprocessing steps are performed in the main stage to standardize writing by hand strokes / letters. These include noise assessment as well as baseline assessment and specialties.

During the character latching phase, the handwriting bitmap is subtracted as a series of pixels in picture format. The drawn character has a higher resolution; It should be cut first, where the null field is expelled and the information is set to the limit.

The partitioning is implemented by alphabetical order on an unusual vertical node, the character matches the direction. Histogram-based segmentation is difficult to implement but gives excellent results. Characters can have different heights and sizes.

These can be broken down by the establishment of local maxima and local minima. They can be clustered based on similar local minimum and maximum values.

Below is the example on how preprocessing works with it's basic steps.(shown in Fig. 8)

International Journal of Science and Research (IJSR)
ISSN: 2319-7064
ResearchGate Impact Factor (2018): 0.28 | SJIF (2019): 7.583

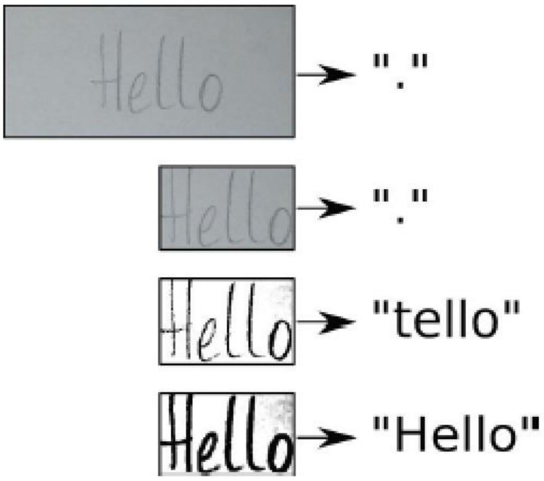

Figure 8: An example showing basic steps of preprocessing

Feature Extraction

Features are the most relevant arrangement of attributes which characterize character images. Rather than utilizing the whole arrangement of pixels of images, just a couple of basic pixels are extricated from images and utilized as features. The essential purpose of the feature extraction step is to eliminate a more appropriate example of classification. These features are of various types, e.g., horizontal features, Vertical features, texture based properties, etc. The selective proof selection of each component depends on the arc type, characteristic angle, relative position, length ratio and connection angle.

Another approach to detecting flexibility (according to biological visualization) is to differentiate basic cells 9 and develop these cells based on the concept of connected component. Some other supporting features such as shape, size, direction of writing, slope and chart and end coordinates may be directional properties. The process of the feature extraction algorithm is clear from its allocation. This involves identifying letters or images based on their characteristics or inseparable elements. This idea is similar to the way people identify roles based on their characteristics or attitude.

The literature reading review provides an investigation into character recognition, with little attention coordinated for facility progress. The validation process is fundamentally enhanced by feature updates because it exposes almost inextricable misconceptions. This method may give the software engineer additional authority over the features used for identification or ID. This method takes more time to lead, yet gives more accurate results.A decent feature set encourages grouping step. It is likewise a sort of dimensionality decrease which efficiently represents critical parts of the image and hides irrelevant details of image.

Below is the flowchart on how preprocessing works along with its basic steps.(shown in Fig. 9)

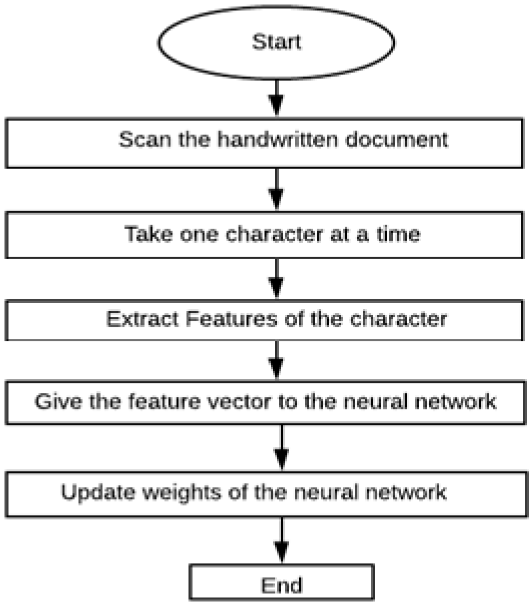

Figure 9: Flowchart showing basic steps in feature extraction

Classification and Post Processing

The classification is called the function of assigning labels (classes, categories) to the ump halves (data occurrence) that have not yet been found. In AI, this happens on the basis of preparing training on most ready-made models. Classification is a managed learning problem where the "teacher" adds a symbol to each occurrence of data. The label is a discrete number that identifies the square where there is space for a particular event. This is usually referred to as a nonaggressive total number.

There are many machine learning models that perform classification; These are called classifiers. The point of classification is to set the selection threshold in the feature-space that separates the manufacturing model with the goal of effectively naming another class perception phenomenon. All in all, the electoral boundary is a hyper-surface that divides n dimensional space into two allocations that are also n. 1 dimensional. Classification is the problem of identifying which class is in many other ump halves, based on a set of ump halves that contain class anonymity known data. Different assumptions enter many quantitative features. There are no best classifiers, however, the use of the classifier depends on many variables, such as accessible read sets and the number of free parameters. In our project we used a nervous system strategy for classification.

The classification of each image is one of the known characters in view of the characteristics of those images in the classification. In this method, each character image is mapped to a text representation.

Post Processing

The final stage of character recognition is post-processing. This is the process of correcting the wrong product using natural language. Once the shape is identified, it is processed by obtaining the product.

In the event that the shape is perceived simply, at that point the precision can be improved as per the information on

International Journal of Science and Research (IJSR)
ISSN: 2319-7064
ResearchGate Impact Factor (2018): 0.28 | SJIF (2019): 7.583

language. Shape recognizers carry on in an unexpected way for various handwriting input.

Data Analysis and Training

The dataset utilized in our undertaking is the IAM data set. The undertaking of extracting and finding the right data set which could be utilized to prepare our framework was a troublesome task. Be that as it may, after numerous endeavors we were at last ready to get the necessary data set for our task. The database contains unpublished handwritten material types, tested at 300dpi targets and remains PNG images with 256 gray levels. All structures and additionally customizable content lines, words and sentences are accessible to download as PNG documents, embedding XML meta-data into a picture record. All contributions in the IAM database are created using sentences provided by LOB Corpus.

The IAM Handwriting Database 3.0 is organized as follows - 657 writers contributed samples of their handwriting, 1539 pages of scanned text, 5685 isolated and labeled sentences, 13353 isolated and labeled text lines and 115'320 isolated and labeled words.

The words have been removed from the pages of filtered content utilizing a programmed division conspire and were confirmed physically. The division plot has been created at our foundation. All structure, line and word images are given as PNG documents and the comparing structure mark records, counting division data and are included in the image files as meta-information in XML format which is XML file and XML file format (DTD). 100% data is trained, since real time input is given.

Fig. 10 below shows some samples of the data set:

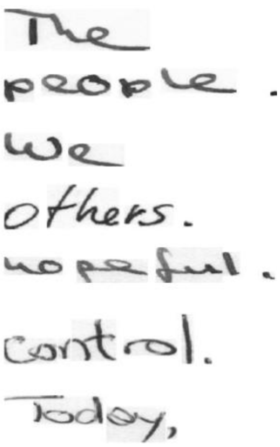

Figure 10: Examples of IAM dataset

Training

To train our model, use the (fit () 'function in our model with the following parameters: training data (train_x), target data (train_y), validation data, and number of ages. For each model, we have set the number of epochs to 10 model. X_test, y_test), Ages = 10).

Model

The CNN model for perceiving handwritten characters is made using python and tensor flow[7]. The most broadly utilized sort of model is SEQUENTIAL. The successive model comprises a direct pile of layers. The equivalent has been utilized for building up the handwritten text recognition model. Characterizing a model was our essential undertaking. Characterizing a model fundamentally implies adding various layers to the stack.

Layer 1: The re-hap design layer, taking the contribution of the primary layer we added (784,1) and converting it (28,28,1).

Layer 2 and Layer 3: Concentrated Layer - The ability of the fusion layer to search for specific features from a given matrix is now an image of a generalized handwritten material. The solid layer uses a layer layer channel network, which is a mixture of zeros and theirs. The channel network slips into the communications framework and places the components that coordinate with the components of the channel grid. Structures constructed in this way are called feature maps.This solid layer takes the layer layer input grid (28, 28, 1). This layer uses a channel of size (5,5) and a step size of 1. It generates 32 feature maps using 32 different channels. The purpose of using the convolution layer twice is to remove more features to improve it
Model accuracy.

Layer 4: MAX POOLING Layer. Maxpooling is eliminated by applying the uncovered sub-regions of the underlying description of the maximum channel (usually). When the images are too large at this point, we need to reduce the amount of parameters we can train. Pooling is accomplished with the sole reason of reducing the spatial size of the photo.

Analyzing the Model

We are running Python Analysis. P with arguments to break picture record data / analysis. Ground-Truth Content "are". Pixel disturbance shows how the pixel affects the right square score. The red pixels will vote in favor of the right square while the blue pixels will vote against the right square. Subsequent plotlines show how the Earth's chances of reading the truth change when the material is moved. As can be seen, the model is not intelligible, since all preparation drawings from the IAM are left-adjustable.(Shown in Fig. 11)

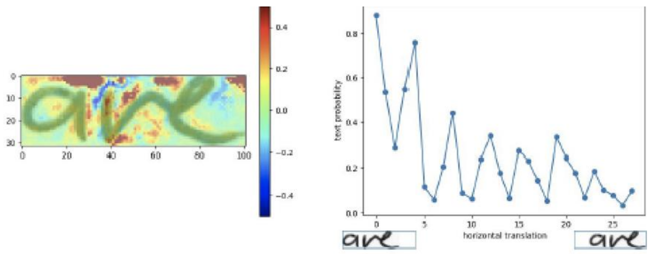

Figure 11: Plots showing the probability score of a certain handwritten word "are"

5. Results

After analyzing the model created for the obtained dataset. The dataset was trained and all the modules were put together in the form of a python code to obtain the desired result i.e. recognizing handwritten words or characters.

The observations and results of some of the examples performed.(shown in Fig. 12 and Fig. 13).

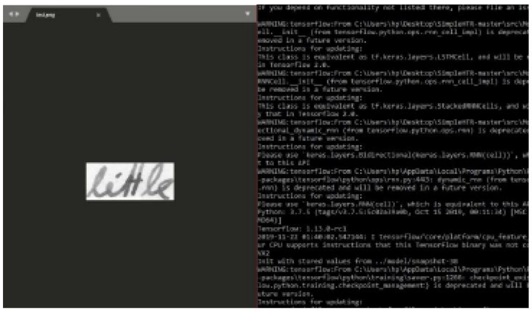

Figure 12: Result of recognition of the word 'are'

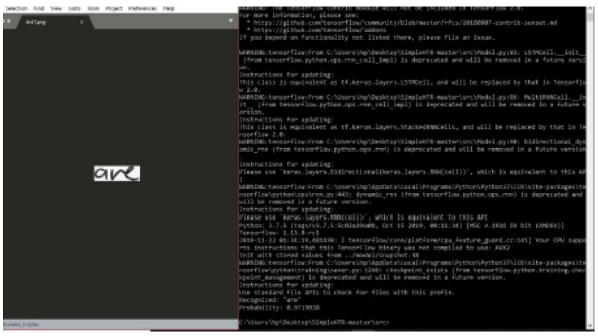

Figure 13: Result of recognition of the word 'little'

After extracting the word by 0CR [8], we wrote the Python code to check the word written in the dictionary. Pyroid is a dictionary module for Python 2/3 that combines word meanings, translations and synonyms. It uses WordNet to get implications, Google for comments and thhtml.com to get synonyms and negative words.

However, for more force and adaptability, we utilized a devoted spell checking library like PyEnchant. Captivate is utilized to check the spelling of words and recommend amendments for words that are miss-spelled. It can utilize numerous well known spell checking bundles to play out this errand, including ispell, aspell and MySpell. It is very adaptable at handling different word references and numerous dialects.

Lastly after having completed the word extraction through 0CR [8] and then checking it in the dictionary we simply used an if else statement to pass the result to the LED's. If the word was found in the dictionary, we stored the "true" value in a variable and passed it to the Arduino[10] which gave the command to green LED. Subsequently for the word not found in the dictionary we created another variable and stored the value "false" and passed it to Arduino[10] which gave the command to red LED.

6. Conclusion

Through this project we have successfully created the prototype model of a "wireless pen" which can be used to write on a paper as well as on the screen directly. We were able to take the input from the user in the form of normal written text and give an output in the form of a LED signal i.e. green for the spelling being right and red for it being wrong.

7. Future Scope

Lot of work and further extensive research needs to be done before making this concept commercially procurable and marketable as a product. Technology like speech recognition can be added so that the device becomes more practically usable. We can implement this by using speech recognition sensors like grove or robodo and then we can work on this using speech recognition API. Future models will also check for grammatical errors.

8. Acknowledgment

The author, thankfully and gratefully acknowledge the sincerecontributions of our faculty and collegewho have been our constant support during the entire span of our project and research. We sincerely thank them for their guidance and encouragement which helped us in carrying out this project so efficiently.

References

[1] Varsha Suresh Jagtap, Prof.Mrs.K. Sujatha, "AN Accelerometer Based Digital Pen with Trajectory Recognition Algorithm For Handwritten Digit AND Gesture Recognition", International Journal of Engineering Sciences & Research Technology Jagtap, vol. 4, no. 5, pp. 1-9, May2015

[2] Palekar, R. R., Parab, S. U., Parikh, D. P., &Kamble, V. N. (2017, April). Real time license plate detection using openCV and tesseract. In Communication and Signal Processing (ICCSP), 2017 International Conference on (pp. 2111-2115). IEEE.

[3] Yi, Chucai, and Yingli Tian. "Scene text recognition in mobile applications by character descriptor and structure configuration." IEEE transactions on image processing 23, no. 7 (2016): 2972-2982.

[4] Z. Lu and L. Wang, "Learning descriptive visual representation for image classification and annotation," Google ML Kit, vol. 48, no. 2, pp. 498–508, 2017.

[5] R. Smith, "An Overview of the Tesseract 0CR[8] Engine", Ninth International Conference on Document Analysis and Recognition (ICDAR 2007), vol. 2, pp. 629-633, 2017.

[6] MPU-6050 Datasheet (PDF) - List of Unclassifed Manufacturers. – URL: http://www.alldatasheet.com/datasheetpdf/pdf/517744/ETC1/MPU-6050.html

[7] M. Agarwal, Shalika, V. Tomar, P. Gupta, "Handwritten Character Recognition using Neural Network and TensorFlow ", International Journal of Innovative Technology and Exploring Engineering (IJITEE), ISSN: 2278-3075, vol. 8, Issue- 6S4, April 2019.

[8] P. K. Charles, V. Harish, M. Swathi, CH. Deepthi, "A Review on the Various Techniques used for Optical Character Recognition[8] " International Journal of Engineering Research and Applications (IJERA), ISSN: 2248-9622, vol. 2, pp.659-662, Issue 1, Jan-Feb 2016.

[9] S. S. Mor, S. Solanki, S. Gupta, S. Dhingra, M. Jain, R. Saxena, "Handwritten Text Recognition: with Deep

Learning and Android ", International Journal of Engineering and Advanced Technology (IJEAT), ISSN: 2249 – 8958, vol. 8, Issue-3S, February 2019.

[10] R. Vaidya , D. Trivedi1 , S. Satra, Prof. M. Pimpale, Design for DC Motor Speed Control Using Arduino[10] Microcontroller, Proceedings of the 2nd International Conference on Inventive Communication and Computational Technologies (ICICCT 2018) IEEE Xplore Compliant , Part Number: CFP18BAC-ART; ISBN:978-1-5386-1974-2.

[11] Ma Yuekun Ying Shiyan Xiao Linrong;Design for One-to-many Wireless Data Transmission System Based on nRF24LE1 Chip[J];Radio Frequency Identification Technologies and Applications;2009-03

[12] Yuelong Li Jinping Li Li Meng, "Character Recognition Based on Hierarchical RBF Neural Networks" Intelligent Systems Design and Applications, 2006. ISDA '06. Sixth International Conference, 1, On Page(s): 127-132, 2006.

[13] G. S. Budhi and R. Adipranata, "Handwritten Javanese Character Recognition Using Several Artificial Neural Network Methods", J.ICT Res. Appl., vol. 8, no. 3, pp. 195–212, 2015.

[14] Y. Liang, J. Wang, S. Zhou, Y. Gong, and N. Zheng, "Incorporating image priors with deep convolutional neural networks for image super resolution", Neurocomputing, Vol. 194, pp. 340- 347, 2016.

[15] O. Vinyals, A. Toshev, S. Bengio, and D. Ethan, "Show and tell: A neural image caption generator", Proceedings of the IEEE Conference Computer Vision and Pattern Recognition, pp. 3156–3164, 2015.