**1.INTRODUCTION**

In today’s world technologies are much increasing mostly in the field of artificial intelligence and machine learning. This project aims to the detection of equation and predicts the result using the convolutional neutral network. The system will detect the equation using the neural networks and predicts the result with limited accuracy.

With the advancement in technology, machine learning and deep learning are playing a crucial role in present times. Now, machine learning and deep learning techniques are being employed in handwriting recognition, robotics, artificial intelligence, and many more fields. Developing such system requires training our machines with data, making it capable to learn and make required prediction. Machine Learning is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed.  This project presents a Handwritten Mathematical Equation Solver using Convolutional Neural Network with some image processing techniques to achieve a decent accuracy of 80.46%. A**Handwritten Equation solver using CNN equation** can contain any digit from 0-9 and symbol +,x,- works on image**with** white background and digits/symbols are in black. It can run all the three ipynb files either separately or sequentially. A convolutional [neural network](https://deepai.org/machine-learning-glossary-and-terms/neural-network), or CNN, is a [deep learning](https://deepai.org/machine-learning-glossary-and-terms/deep-learning) neural network designed for processing structured arrays of data such as images. Convolutional neural networks are widely used in [computer vision](https://deepai.org/machine-learning-glossary-and-terms/computer-vision) and have become the state of the art for many visual applications such as image classification, and have also found success in [natural language processing](https://deepai.org/machine-learning-glossary-and-terms/natural-language-processing) for text classification.

**2. SYSTEM ANALYSIS**

**2.1 PROPOSED METHODOLGY**

The model that is proposed by us to classify the equation solver consist of 7. Those phases are

* Data collection
* Image preprocessing
* Selection of classification
* Modeling
* Training
* Testing
* Prediction

**Data Collection:**

Firstly, the images of traffic signs were collected from online sources such as Kaggles. The dataset consists of 39209 images and divided into 43 different classes. This dataset consists of different type images with

**Image Preprocessing:**

In this step images are resized to smaller pixel size in order to speed up the computation. The acquired images contain some noise. This noise is removed using some filtering techniques like Gaussian Blur. After the images are present in RGB format which is not appropriate for further work as RGB format is unable to separate image intensity. Also, RGB color space is noisier than Gray.

**Classifier:**

This is the classification problem as we have to classify the type of disease on the leaf of the plant. So, we have plenty of machine learning as well as deep learning algorithms that we can apply on this dataset. We have decided to start with low complex algorithms and increasing the complexity level order to increase accuracy of the model. We have selected classifier namely – CNN

* **CNN (Convolutional Neural Network)**

A convolutional neural network consists of an input layer, [hidden layers](https://en.wikipedia.org/wiki/Multilayer_perceptron#Layers) and an output layer. In any feed-forward neural network, any middle layers are called hidden because their inputs and outputs are masked by the activation function and final [convolution](https://en.wikipedia.org/wiki/Convolution). In a convolutional neural network, the hidden layers include layers that perform convolutions. Typically, this includes a layer that performs a [dot product](https://en.wikipedia.org/wiki/Dot_product) of the convolution kernel with the layer's input matrix. This product is usually the Fresenius, and its activation function is commonly [ReLU](https://en.wikipedia.org/wiki/Rectifier_(neural_networks)" \o "Rectifier (neural networks)). As the convolution kernel slides along the input matrix for the layer, the convolution operation generates a feature map, which in turn contributes to the input of the next layer. This is followed by other layers such as pooling layers, fully connected layers, and normalization layers.

### Convolutional layers

### Convolutional layers convolve the input and pass its result to the next layer. This is similar to the response of a neuron in the visual cortex to a specific stimulus. Each convolutional neuron processes data only for its [receptive field](https://en.wikipedia.org/wiki/Receptive_field). Although [fully connected feedforward neural networks](https://en.wikipedia.org/wiki/Multilayer_perceptron) can be used to learn features and classify data, this architecture is generally impractical for larger inputs such as high resolution images. It would require a very high number of neurons, even in a shallow architecture, due to the large input size of images, where each pixel is a relevant input feature. For instance, a fully connected layer for a (small) image of size 100 x 100 has 10,000 weights for *each* neuron in the second layer. Instead, convolution reduces the number of free parameters, allowing the network to be deeper. For example, regardless of image size, using a 5 x 5 tiling region, each with the same shared weights, requires only 25 learnable parameters. Using regularized weights over fewer parameters avoids the vanishing gradients and exploding gradients problems seen during [back propagation](https://en.wikipedia.org/wiki/Backpropagation) in traditional neural networks. Furthermore, convolutional neural networks are ideal for data with a grid-like topology (such as images) as spatial relations between separate features are taken into account during convolution and/or pooling.

### Pooling layers

### Convolutional networks may include local and/or global pooling layers along with traditional convolutional layers. Pooling layers reduce the dimensions of data by combining the outputs of neuron clusters at one layer into a single neuron in the next layer. Local pooling combines small clusters, tiling sizes such as 2 x 2 are commonly used. Global pooling acts on all the neurons of the feature map. There are two common types of pooling in popular use: max and average. Max pooling uses the maximum value of each local cluster of neurons in the feature map, while average pooling takes the average value.

### Activation layer

### In any neural network activation layer plays in important role as it is responsible for non-linear learning of the network. There are different type of activation function such as sigmoid, tanh, ReLu, LeakyReLu, In our model we have used ReLu foe all the layer except output layer for which we have used softmax.

### Fully connected layers

[Fully connected layers connect every neuron in one layer to every neuron in another layer](https://en.wikipedia.org/wiki/Layer_(deep_learning)#Dense_layer). It is the same as a traditional [multi-layer perceptron](https://en.wikipedia.org/wiki/Multi-layer_perceptron) neural network (MLP). The flattened matrix goes through a fully connected layer to classify the images.

**NOTE:** The CNN model was able to give accuracy of 98% on testing set that is so far best among all classifier with 0.033 loss

|  |  |
| --- | --- |
| **CLASSIFIER** | **ACCURACY** |
| **CNN** | **88.0** |

**Modelling:**

 Machine learning is an offshoot of artificial intelligence, which analyzes data that automates analytical model building. Machine learning tells us that systems can, if trained, identify patterns, learn from data, and make decisions with little or no human intervention. On the other hand, machine learning models are files trained to recognize particular pattern types. Models are also known as the resulting output of the training process and are considered the mathematical representation of real-world processes.

**Training:**

A training model is a dataset that is used to train an ML algorithm. It consists of the sample output data and the corresponding sets of input data that have an influence on the output. The training model is used to run the input data through the algorithm to correlate the processed output against the sample output. The result from this correlation is used to modify the model.

**Testing:**

In machine learning, model testing is referred to as the process where the performance of a fully trained model is evaluated on a testing set. The testing set consisting of a set of testing samples should be separated from the both training and validation sets, but it should follow the same probability distribution as the training set.

**Prediction:**

Prediction refers to the output of an algorithm after it has been trained on a historical dataset and applied to new data when forecasting the likelihood of a particular outcome

### 2.2 WORK FLOW DIAGRAM

### PREPROCESSING OF IMAGE

SEGMENTATION AND CHARACTERISATION OF EQUATION

CLASSIFICATION OF CHARACTERS USING PRETRAINED CNN AND MAKE EQUATION LIKE STRING

### ANALYSING RESULTS

DETECTED EQUATION AND ROOTS

### 3. SOFTWARE REQUIREMENTS

### The language is used to develop Traffic sign detection is based on digital image processing. The system is used to predict the sign boards in python. Python is an interpreted high level programming language for general purpose programming.

### PYTHON

### Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance.

**INTERNET BROWSER**

A **web browser** (commonly referred to as a **browser**) is [application software](https://en.wikipedia.org/wiki/Application_software) for accessing the [World Wide Web](https://en.wikipedia.org/wiki/World_Wide_Web). When a [user](https://en.wikipedia.org/wiki/User_(computing)) requests a [web page](https://en.wikipedia.org/wiki/Web_page) from a particular [website](https://en.wikipedia.org/wiki/Website), the web browser retrieves the necessary content from a [web server](https://en.wikipedia.org/wiki/Web_server) and then displays the page on the user's device.

A web browser is not the same thing as a [search engine](https://en.wikipedia.org/wiki/Search_engine), though the two are often confused. A search engine is a website that provides [links](https://en.wikipedia.org/wiki/Hyperlink) to other websites. However, to connect to a website's server and display its web pages, a user must have a web browser installed.

### IMPORTANT PYTHON LIBRARIES

### Tensor Flow

### Tensor Flow is an open source framework developed by Google researchers to run [machine learning](https://www.techtarget.com/searchenterpriseai/definition/machine-learning-ML), [deep learning](https://www.techtarget.com/searchenterpriseai/definition/deep-learning-deep-neural-network) and other statistical and predictive analytics workloads. Like similar platforms, it's designed to streamline the process of developing and executing advanced analytics applications for users such as data scientists, statisticians and predictive modelers. The Tensor Flow software handles data sets that are arrayed as computational nodes in graph form. The edges that connect the nodes in a graph can represent multidimensional vectors or matrices, creating what are known as tensors. Because Tensor Flow programs use a data flow architecture that works with generalized intermediate results of the computations, they are especially open to very large-scale parallel processing applications, with [neural networks](https://www.techtarget.com/searchenterpriseai/definition/neural-network).

### Open CV

OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products. Being a BSD-licensed product, OpenCV makes it easy for businesses to utilize and modify the code. The library has more than 2500 optimized algorithms, which includes a comprehensive set of both classic and state-of-the-art computer vision and machine learning algorithms. These algorithms can be used to detect and recognize faces, identify objects, classify human actions in videos, track camera movements, track moving objects, extract 3D models of objects, produce 3D point clouds from stereo cameras, stitch images together to produce a high resolution image of an entire scene, find similar images from an image database, remove red eyes from images taken using flash, follow eye movements, recognize scenery and establish markers to overlay it with augmented reality,

* **Keras:**

**Keras is a neural network Application Programming Interface (API) for Python that is tightly integrated with TensorFlow, which is used to build machine learning models. Keras’ models offer a simple, user-friendly way to define a neural network, which will then be built for you by TensorFlow.**

* **Numpy:**

NumPy is the fundamental package for scientific computing in Python. It is a Python library that provides a multidimensional array object, various derived objects (such as masked arrays and matrices), and an assortment of routines for fast operations on arrays, including mathematical, logical, shape manipulation, sorting, selecting, I/O, discrete Fourier transforms, basic linear algebra, basic statistical operations, random simulation and much more. At the core of the NumPy package, is the ndarray object. This encapsulates n-dimensional arrays of homogeneous data types, with many operations being performed in compiled code for performance.

* **OS**

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

* **Matplotlib:**

Matplotlib is an amazing visualization library in Python for 2D plots of arrays. Matplotlib is a multi-platform data visualization library built on NumPy arrays and designed to work with the broader SciPy stack. It was introduced by John Hunter in the year 2002. One of the greatest benefits of visualization is that it allows us visual access to huge amounts of data in easily digestible visuals. Matplotlib consists of several plots like line, bar, scatter, histogram etc.

* **Flask**

Flask is a micro web framework written in Python. It is classified as a micro framework because it does not require particular tools or libraries. ... Extensions exist for object-relational mappers, form validation, upload handling, various open authentication technologies and several common frame work related tools.

### 4.SYSTEM REQUIREMENTS

|  |  |
| --- | --- |
| SOFTWARE | HARDWARE |
| OPERATING SYSTEM : WINDOWS 8 | PROCESSOR : PENTIUM 3 AND ABOVE, AMD, A6.. |
| SOFTWARE USED : INTERNET BROWSER | SPEED : 2GHZ AND ABOVE |
| DATA SCIENCE TOOLS: NUMPY, KERAS, MATPLOTLIB, PANDAS, CV2,TENSORFLOW,FLASK | RAM : 2GB |
| SCRIPT: PYTHON | DISK SPACE : 50GB |
| INTERPRETER : PYTHON 3.8 | KEYBOARD : STANDARD WINDOWS KEYBOARD |

### 5.SYSTEM DESIGN

### 5.1 DATA FLOW DIAGRAM

### LEVEL 0

### 

### LEVEL 1

### 

### LEVEL 1.1

### 

### LEVEL 1.2

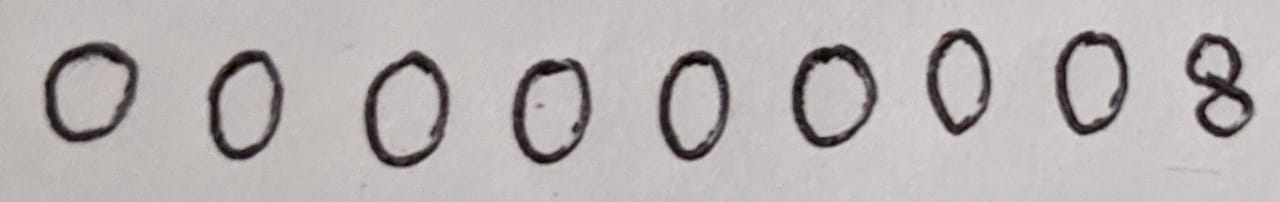
### 

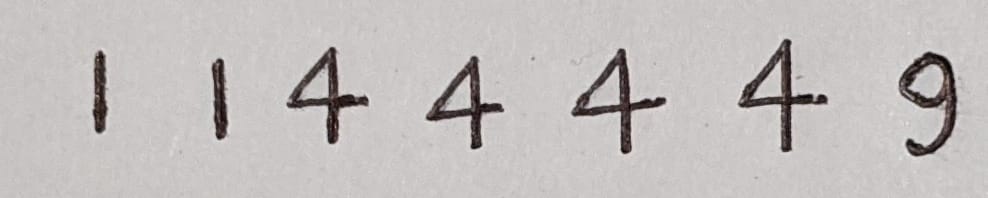
### LEVEL 1.3

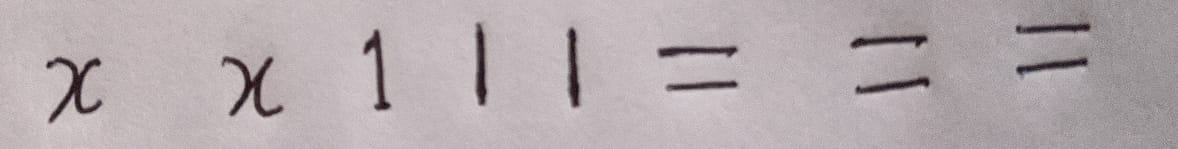
### 

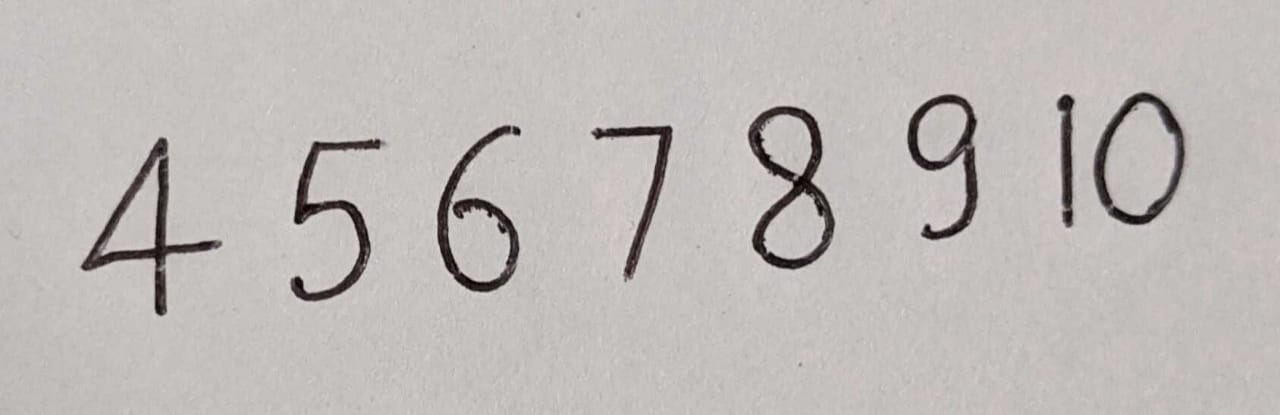
**6.DATASET COLLECTION**

In this system the dataset collected from the Kaggles and MNIST dataset.

****

****

****

****

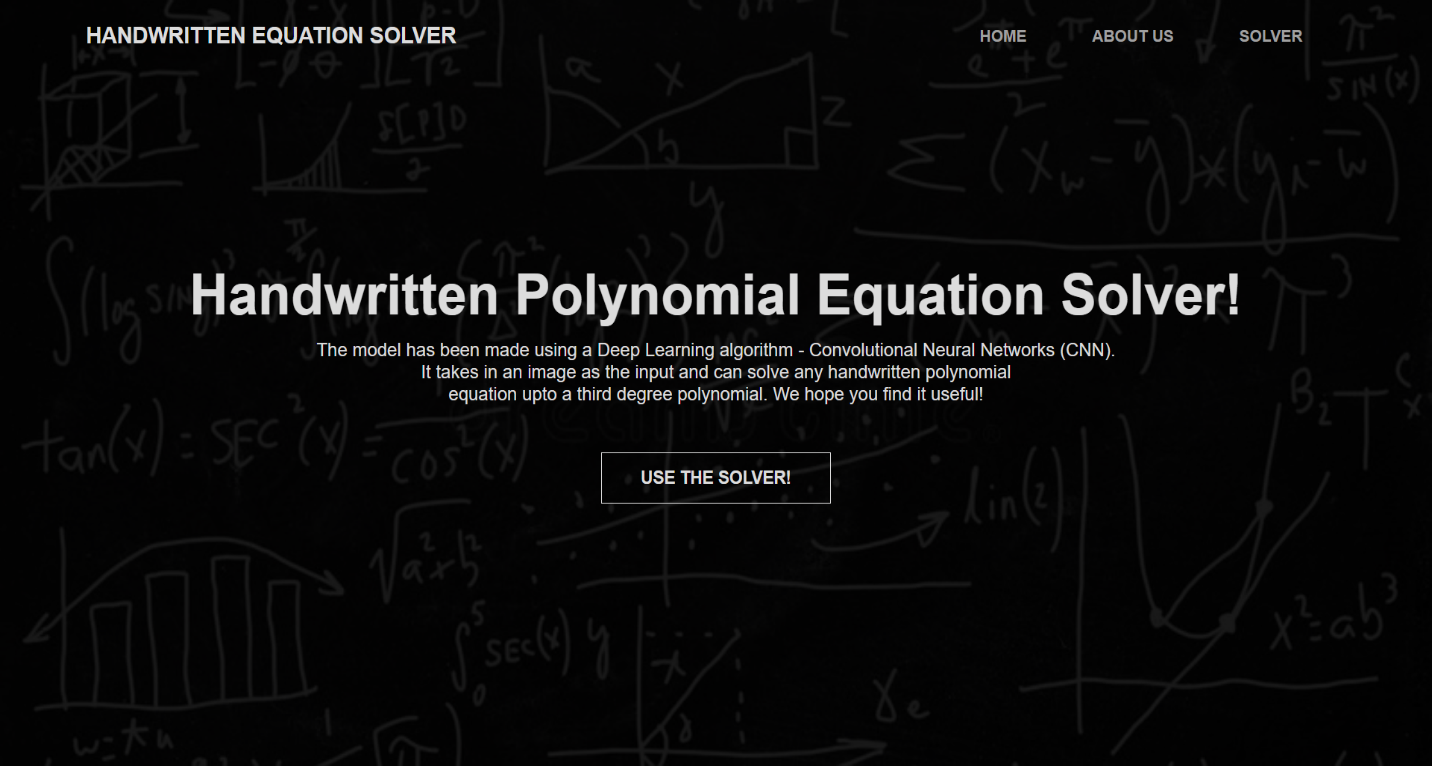
**7.PRODUCT BACKLOG**

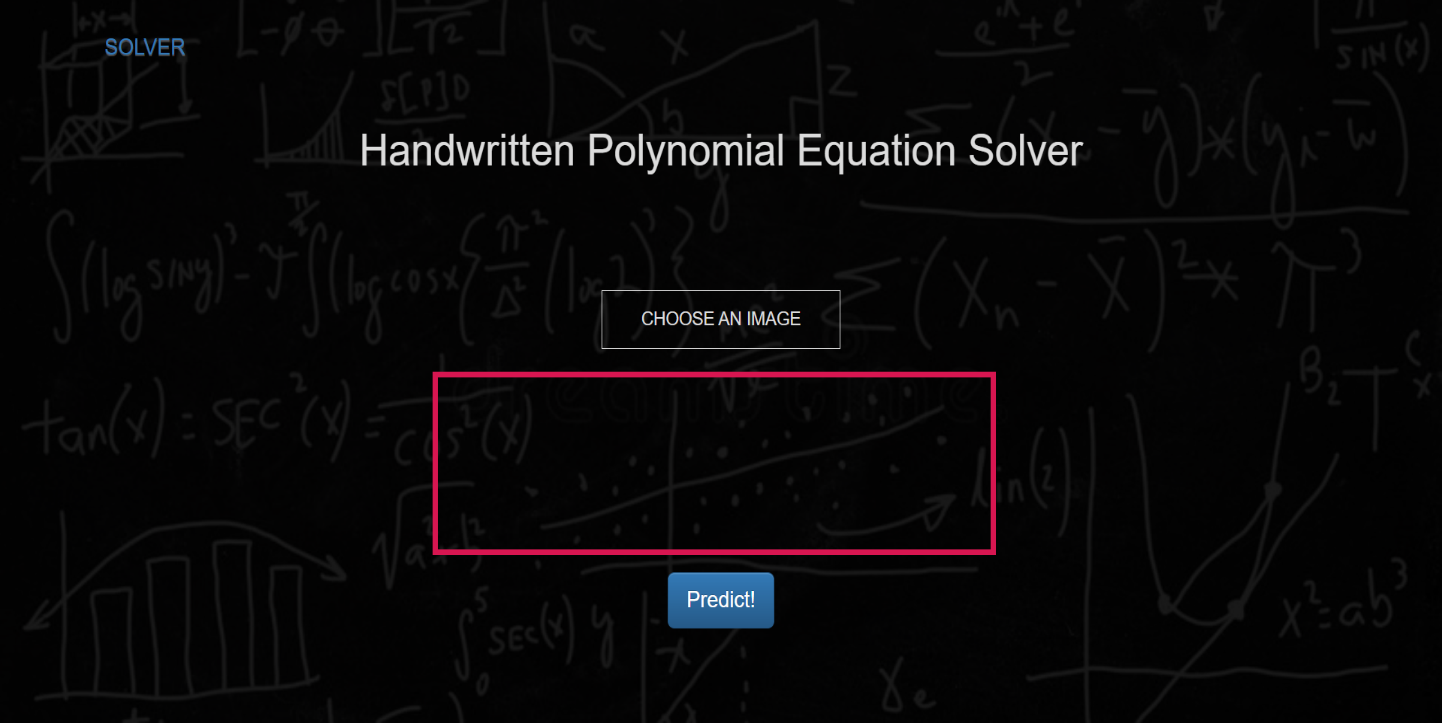
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | USER STORIES | PRIORITY | COMMENT FROM SCRUM MASTER | COMMENT FROM PRODUCT OWNER |
| 1 | Equation Detection and Finding result | Very high |  |  |
| 2 | Data collections | Very high |  | Kaggles, MNIST |
| 3 | Requirements | High |  | Python3.8 |
| 4 | Splitting Data | High |  | Train test split |
| 5 | Feature extraction | High |  | Based on color and shape |
| 6 | Pre-processing image | High |  | Convert all images in same size |
| 7 | Image data generation | High |  | Images in real time |
| 8 | Model choosing | High |  | CNN |
| 9 | Save model | High |  | Save model using keras |
| 10 | Training model | High |  | Using saved model |
| 11 | Upload Image | High |  | Upload folder in jpeg |
| 12 | UI Design | High |  | HTML |
| 13 | Result | High |  | Based on input |
| 14 | Documentation | High |  |  |

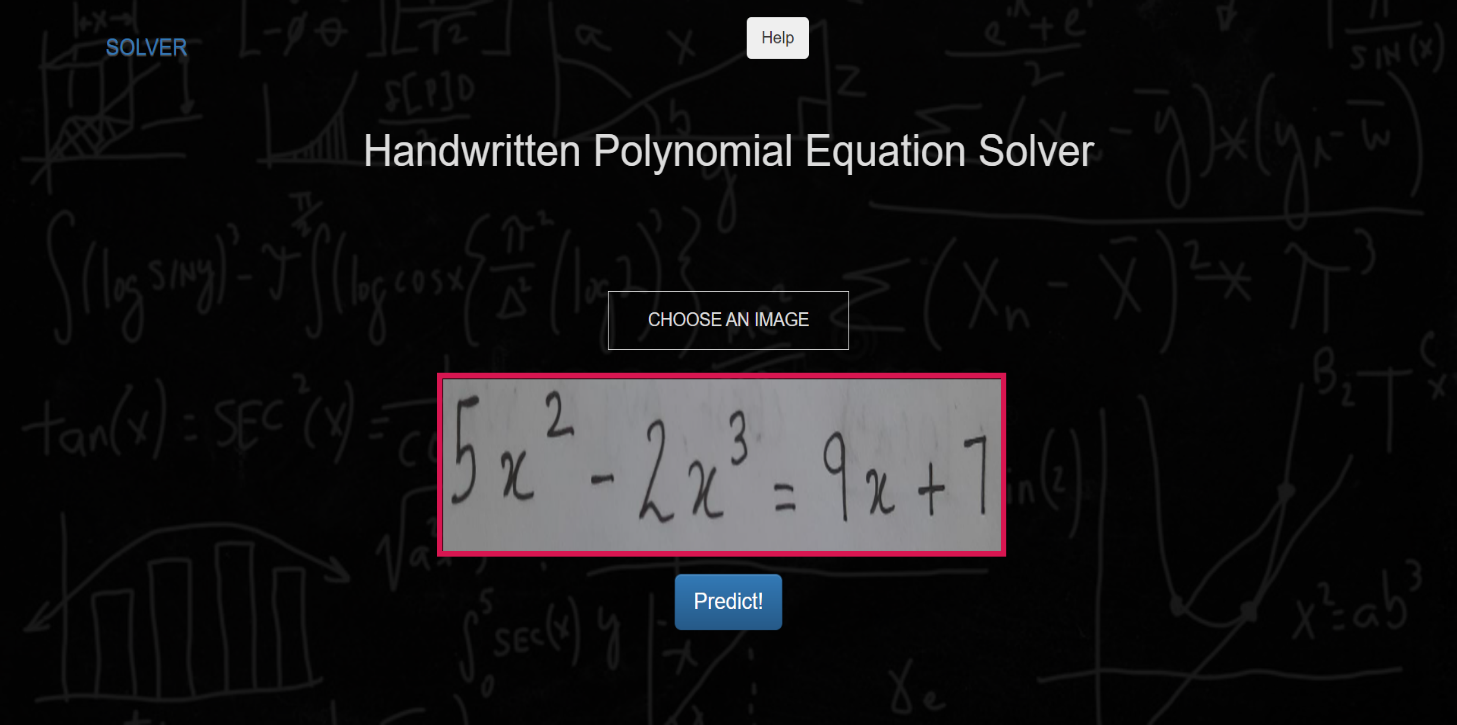
**8. SPRINT BACKLOG**

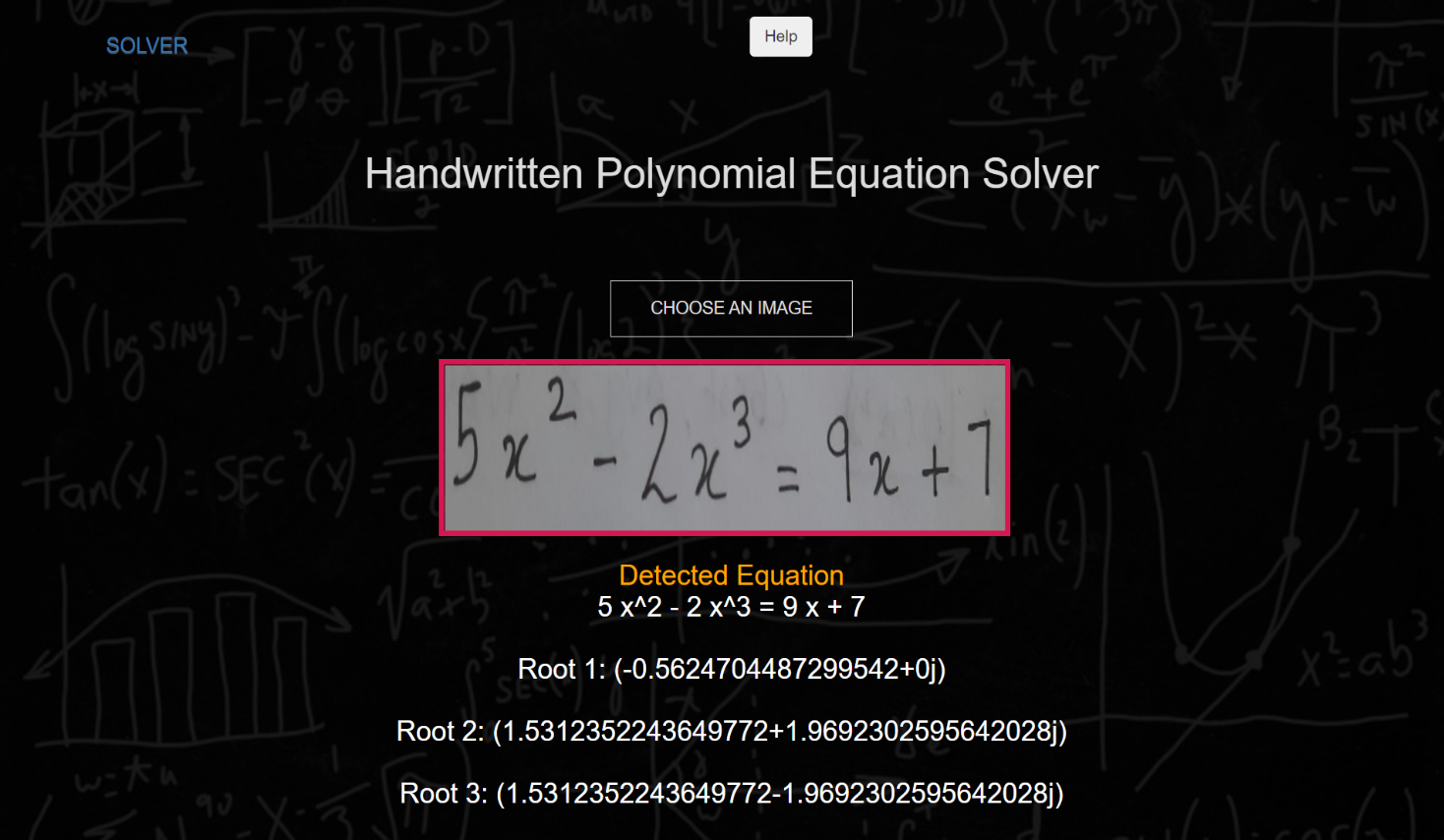
|  |  |  |  |
| --- | --- | --- | --- |
| **USER STORIES** | **NOT STARTED** | **IN PROGRESS** | **COMPLETED** |
| Data collection |  |  | Completed |
| Data preparation |  |  | Completed |
| Model Identification |  |  | Completed |
| Different Images Output |  |  | Completed |
| Training Model |  |  | Completed |
| Testing Model |  |  | Completed |
| Testing Model |  |  | Completed |
| Evaluation Model |  |  | Completed |
| Prediction |  |  | Completed |
| UI Design |  |  | Completed |
| Documentation |  |  | Completed |

**9. SCREENSHOTS**

****

****

****

****

**10. CONCLUSION**

The main scope this project is to give a very accurate artificial intelligence solution for classifying different mathematical equations which makes use of convolutional neural network for classifying purpose. The presented model used the dataset that consist of more than 42000 images with 43 classes. Each image is classifying using convolutional neural network and predict the correct class. The model then can be developed on Android and as well as iOS platform to reach out the driver who can make the actual use of the proposed system. This model detects the equation and predicts the result with the limited accuracy.

**11.REFERNCES**

1. https://www.kdnuggets.com
2. https://www.reasearchgate.com
3. SolveIt: An Application for Automated Recognition and Processing of Handwritten Mathematical Equations. Publisher: IEEE
4. Handwritten Math Problem Solver Using Convolutional Neural Network by Mohini Lokhande, Kriti Murudkar, Arbaaz Nadaf, Niyati Shah
5. https://www.irjet.com