A Novel Method For Handwritten Digit Recognition System

Project Report

TEAM ID: PNT2022TMID45366

S.NO	TEAM MEMBERS	REGISTER NUMBER
1	VINOTH KUMAR. S	812419104075
2	ANTONY RAJA. S	812419104301
3	HASAN UL KHAJA HUSSAIN. S. A (Team Lead)	812419104303
4	Sukumar.R	812419104309

1. INTRODUCTION

HANDWRITTEN digit recognition is the ability of a computer system to recognize the handwritten inputs like digits, characters etc. from a wide variety of sources like emails, papers, images, letters etc. This has been a topic of research for decades. Some of the research areas include signature verification, bank check processing, postal address interpretation from envelopes etc. Here comes the use of Deep Learning. In the past decade, deep learning has become the hot tool for Image Processing, object detection, handwritten digit and character recognition etc. A lot of machine learning tools have been developed like scikit-learn, scipy-image etc. and pybrains, Keras, Theano, Tensorflow by Google, TFLearn etc. for Deep Learning. These tools make the applications robust and therefore more accurate. The Artificial Neural Networks can almost mimic the human brain and are a key ingredient in image processing field. For example, Convolutional Neural Networks with Back Propagation for Image Processing, Deep Mind by Google for creating Art by learning from existing artist styles.

Handwriting Recognition has an active community of academics studying it. The biggest conferences for handwriting recognition are the International Conference on Frontiers in Handwriting Recognition (ICFHR), held in even-numbered years, and the International Conference on Document Analysis and Recognition (ICDAR), held in odd-numbered years. Both of these conferences are endorsed by the IEEE. Active areas of research include: Online Recognition, OfflineRecognition, Signature Verification, Postal-Address Interpretation,Bank-Check Processing, Writer Recognition. Classification of images and patterns has been one of the major implementation of Machine Learning and Artificial Intelligence. People are continuously

trying to make computers intelligent so that they can do almost all the work done by humans Handwriting recognition system is the most basic and an important step towards this huge and interesting area of Computer Vision. Handwritten digit recognition has recently been of very interest among the researchers because of the evolution of various Machine Learning, Deep Learning and Computer Vision algorithms.

1.1 Project Overview

In this report, We compare the results of some of the most widely used Machine Learning Algorithms like CNN- convolution neural networks and with Deep Learning algorithm like multilayer CNN using Keras with Theano and Tensorflow. MNIST is a dataset which is widely used for handwritten digit recognition. The dataset consist of 60,000 training images and 10,000 test images.The artificial neural neworks can all most mimic the human brain and are a key ingredient in image processing field. The advance of handwriting processing results from a combination of various elements, for example: improvements in the recognition rates, the use of complex systems to integrate various kinds of information, and new technologies such as high quality high speed scanners and cheaper and more powerful CPUs. Some handwriting recognition system allows us to input our handwriting into the system

1.2 PURPOSE

Handwritten digits can be done either by controlling a mouse or using a third-party drawing tablet. The input can be converted into typed text or can be left as an "ink object" in our own handwriting. We can also enter the text we would like the system to recognize into any Microsoft Office program file by typing. We can do this by typing 1s and 0s. This works as a Boolean variable. Handwriting recognition [4] is not a new technology, but it has not gained public attention until recently. The ultimate goal of designing a handwriting recognition system with an accuracy rate of 100% is quite illusionary, because even human beings are not able to recognize every handwritten text without any doubt. For example, most people can not even read their own notes.

2. LITERATURE SURVEY

In today's society, character recognition is becoming increasingly vital. It facilitates human work and aids with the resolution of more difficult issues. One illustration is handwritten character recognition, which is extensively used worldwide. This technique was created to recognise zip codes or postal codes for use in mail sorting. This can aid people in the difficult-to-read postal code mail sorting process. Researchers have been working on handwriting recognition for more than thirty years. The number of firms participating in handwriting recognition research has steadily expanded over the last several years. Handwriting processing has advanced due to a mix of factors such as improved recognition rates and the usage of complicated systems. We can enter our handwriting into some handwriting recognition systems. Either using a mouse or a third-party drawing tablet, you can accomplish this. We have the option of typing the input or leaving it as an "ink object" with our own handwriting. Additionally, we can manually type the content into any Microsoft Office software file that we want the system to identify. Typing 1s and 0s will allow us to do this. As a Boolean variable, this operates.

2.1 Existing Problem

Handwriting number recognition is a challenging problem researchers had been research into this area for so long especially in the recent years. In our study there are many fields concern with numbers, for example, checks in banks or recognizing numbers in car plates, the subject of digit recognition appears. A system for recognizing isolated digits may be as an approach for dealing with such application. In other words, to let the computer understand the Arabic numbers that is written manually by users and views them according to the computer process. Scientists and engineers with interests in image processing and pattern recognition have developed various approaches to deal with handwriting number recognition problems such as, minimum distance, decision tree and statistics.

2.2 References

Year: 2021

Authors: Ayush Kumar Agrawal and Vineet Kumar Awasthi

An artificial neural network has one hidden layer between the input and output layers, whereas a deep neural network has numerous hidden layers with input and output layers. Deep neural networks use several hidden layers to increase model performance and achieve higher accuracy compared to accuracy of machine learning models. Most researchers do their research in the area of pattern recognition. In the field of pattern recognition, there are many patterns that can be used, including handwritten numbers, characters, pictures, faces, sounds, and speech. This study focuses on the classification and recognition of handwritten digits.1000 were utilized as test samples and 1000 were training samples.10000 picture samples make up the USPS dataset, of which 7291 serve as training samples and 2007 serve as testing samples. We've used the proposed deep neural network technique in this paper to classify and identify data from the ARDIS and USPS datasets. The suggested model consists of six layers with softmax and relu activation functions. After model implementation, accuracy for ARDIS samples reached 98.70% testing and 99.76% training, which is greater than accuracy from prior research. Additionally, using the USPS samples dataset, 98.22% training accuracy and 93.01% testing accuracy were attained. When compared to earlier methodologies, the data show that deep neural networks perform incredibly well.

2.3 Problem Statement Definition

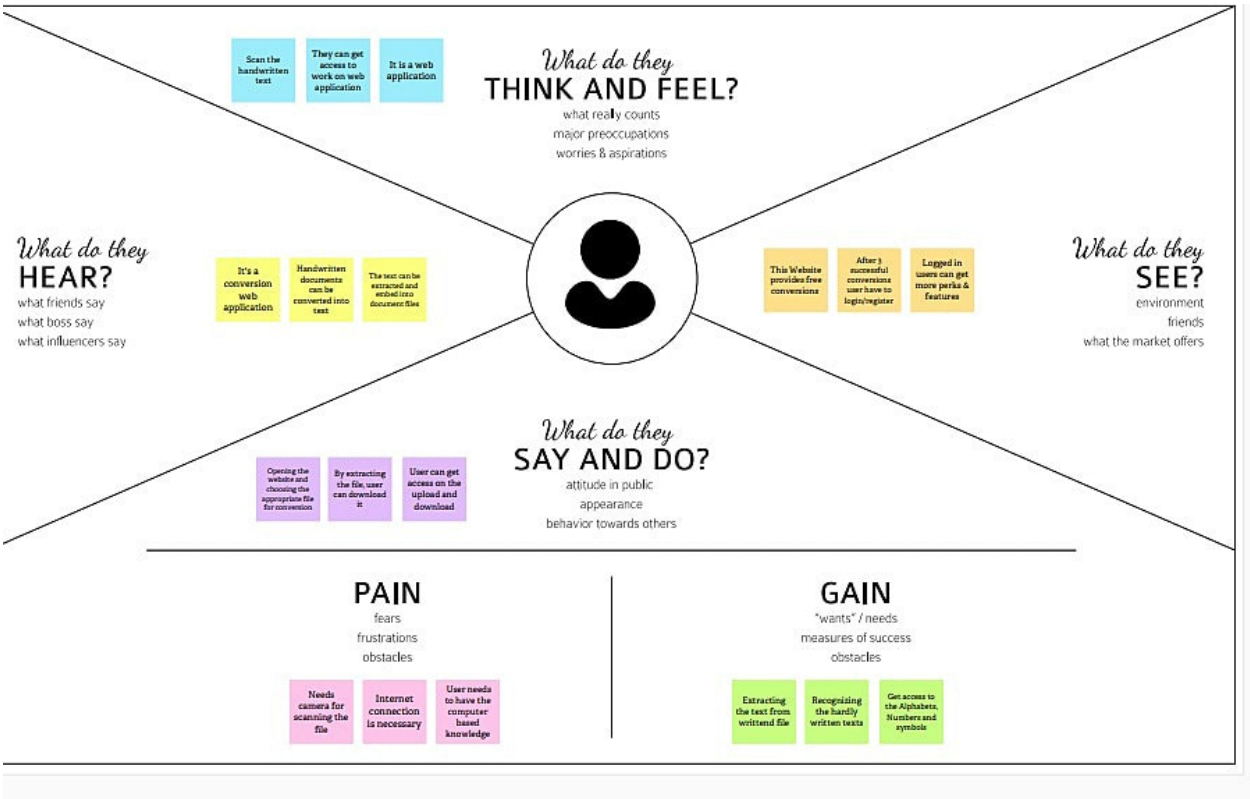
Different users had their own handwriting styles where here the main challenge falls to let computer system understand these different handwriting styles and recognize them as standard writing. We presented a system for dealing with such problem. The system started by acquiring an image containing digits, this image was digitized using some optical devices and after applying some enhancements and

modifications to the digits within the image it can be recognized using feed forward back propagation algorithm.

3.IDEATION AND PROPOSED SOLUTION

3.1 Empathy Map canvas

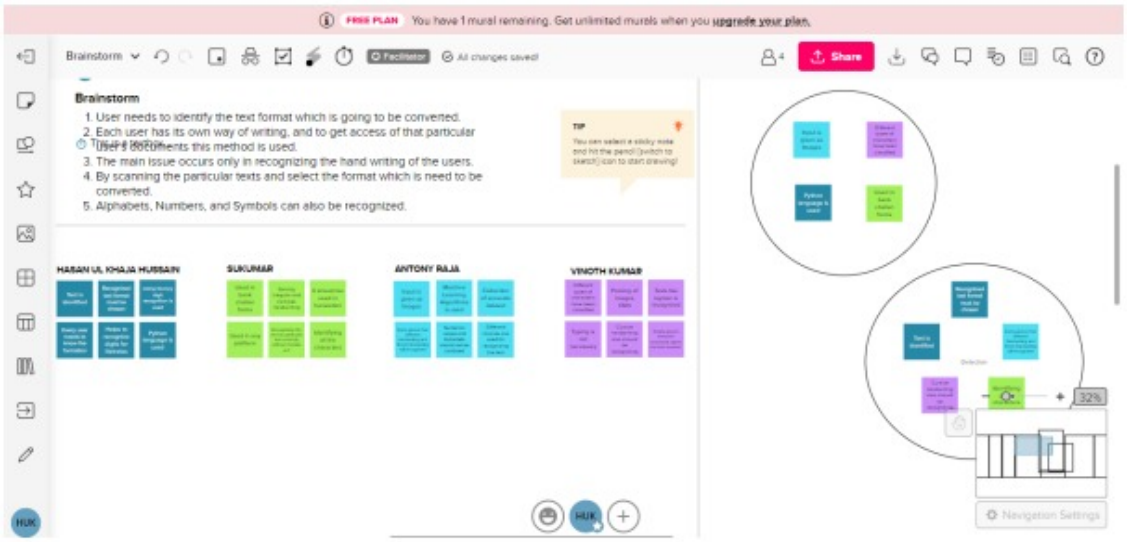
An empathy map is a simple, easy-to-digest visual that captures knowledge about a user's behaviours and attitudes. It is a useful tool to helps teams better understand their users. Creating an effective solution requires understanding the true problem and the person who is experiencing it. The exercise of creating the map helps participants consider things from the user's perspective along with his or her goals and challenges.



3.2 Ideation and Brainstorming

Brainstorming provides a free and open environment that encourages

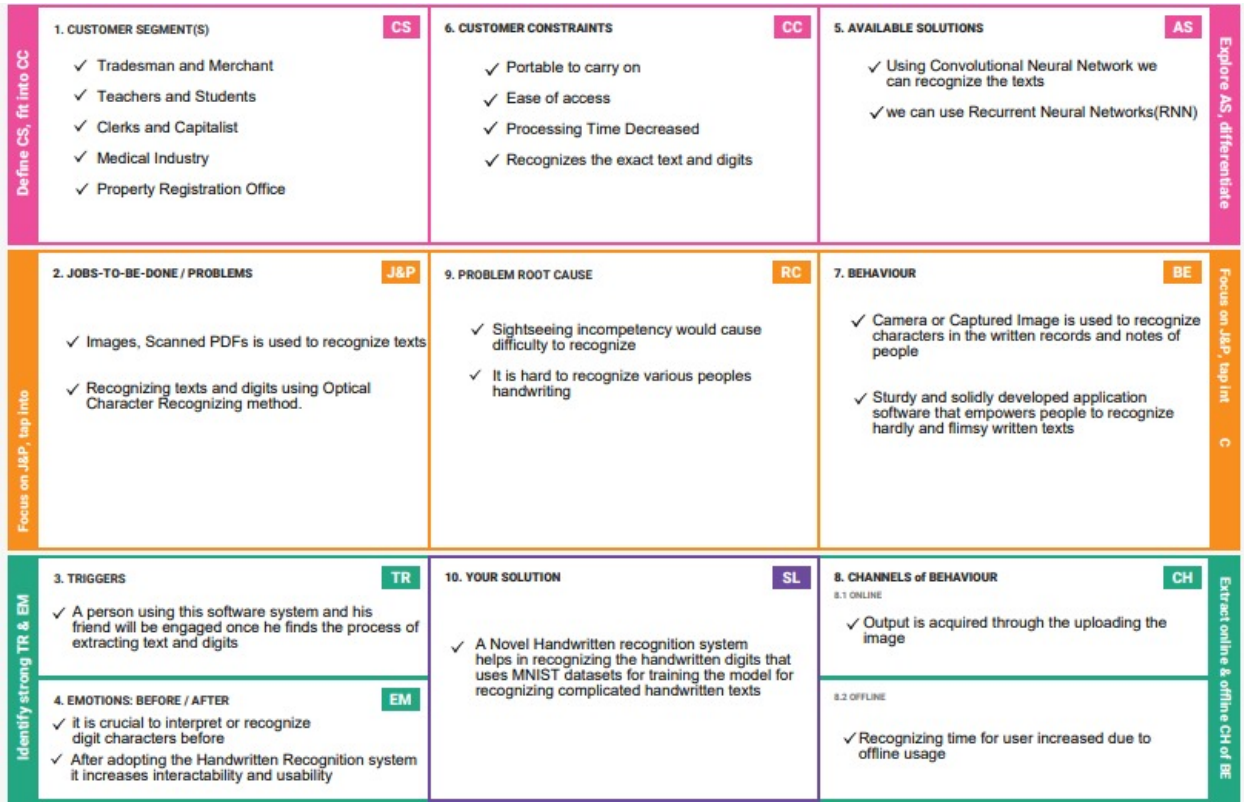
everyone within a team to participate in the creative thinking process that leads to problem solving. Prioritizing volume over value, out-of-the-box ideas are welcome and built upon, and all participants are encouraged to collaborate, helping each other develop a rich amount of creative solutions. Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.



3.3 Proposed solution

S.NO	PARAMETER	DESCRIPTION
1.	Problem Statement (Problem to be solved)	The shape and will be varying for each user. So recognizing the text will not easier.
2.	Idea/Solution description	This Hand written digit recognition method will solve the problem and also recognize the digits present in the image.
3.	Novelty/Uniqueness	This system is mainly used for digit recognition but the other existing system is also to be recognized for alphabets and also expressions.
4.	Social Impact/Customer Satisfaction	• Postal department and courier services can easily find the digits written. • Old people who will have eye sight issues with handwritten digits can use this system to recognize the handwritten digits correctly.
5.	Business Model	It provides services to Banking sector and Postal sector by converting the system to a business model.

3.4 Proposed solution Fit



4.REQUIREMENT ANALYSIS

4.1 Functional Requirements

i) Image Data: Handwritten digit recognition refers to a computer's capacity to identify human handwritten digits from a variety of sources, such as photographs, documents, touch screens, etc., and categorise them into ten established classifications (0-9). In the realm of deep learning, this has been the subject of countless studies.

ii) Website: Web hosting makes the code, graphics, and other items that make up a website accessible online. A server hosts every website you've ever visited. The type of hosting determines how much space is allotted to a website on a server. Shared, dedicated, VPS, and reseller hosting are the four basic varieties.

iii) Digit Classifier Model: To train a convolutional network to predict the digit from an image, use the MNIST database of handwritten digits. get the training and validation data first.

iv) Cloud: The cloud offers a range of IT services, including virtual storage, networking, servers, databases, and applications. In plain English, cloud computing is described as a virtual platform that enables unlimited storage and access to your data over the internet.

v)Modified National Institute of Standards and Technology dataset: The abbreviation MNIST stands for the MNIST dataset. It is a collection of 60,000 tiny square grayscale photographs, each measuring 28 by 28, comprising handwritten single digits between 0 and 9.

4.2 Non Functional Requirements:

i) Usability: One of the very significant problems in pattern recognition applications is the recognition of handwritten characters. Applications for digit recognition include filling out forms, processing bank checks, and sorting mail.

ii) Security: The system generates a thorough description of the instantiation parameters, which might reveal information like the writing style, in addition to a categorization of the digit. The generative models are capable of segmentation driven by recognition. The procedure uses a relatively.

iii) Reliability: The samples are used by the neural network to automatically deduce rules for reading handwritten digits. Furthermore, the network may learn more about handwriting and hence enhance its accuracy by increasing the quantity of training instances. Numerous techniques and algorithms, such as Deep Learning/CNN, SVM,

Gaussian Naive Bayes, KNN, Decision Trees, Random Forests, etc., can be used to recognise handwritten numbers.

iv) Accuracy: With typed text in high -quality photos, optical character recognition (OCR) technology offers accuracy rates of greater than 99%. However, variances in spacing, abnormalities in handwriting, and the variety of human writing styles result in less precise character identification.

5. PROJECT DESIGN

5.1 Data Flow Diagrams

Figure 1:

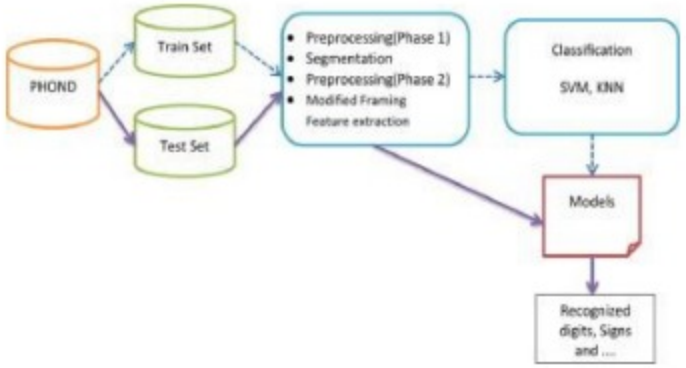


Figure 2:

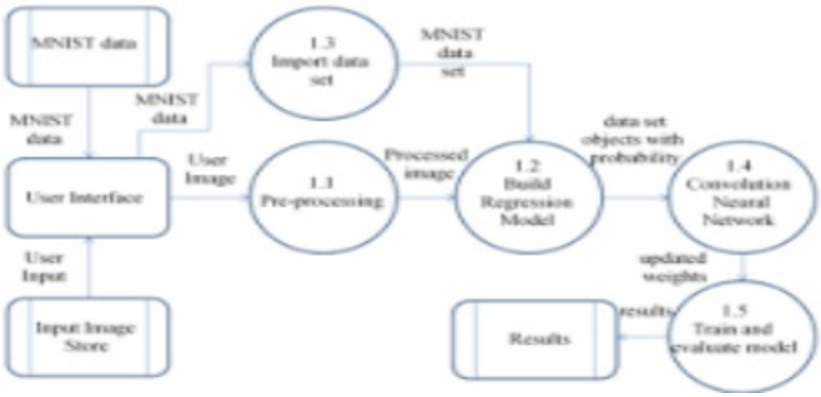
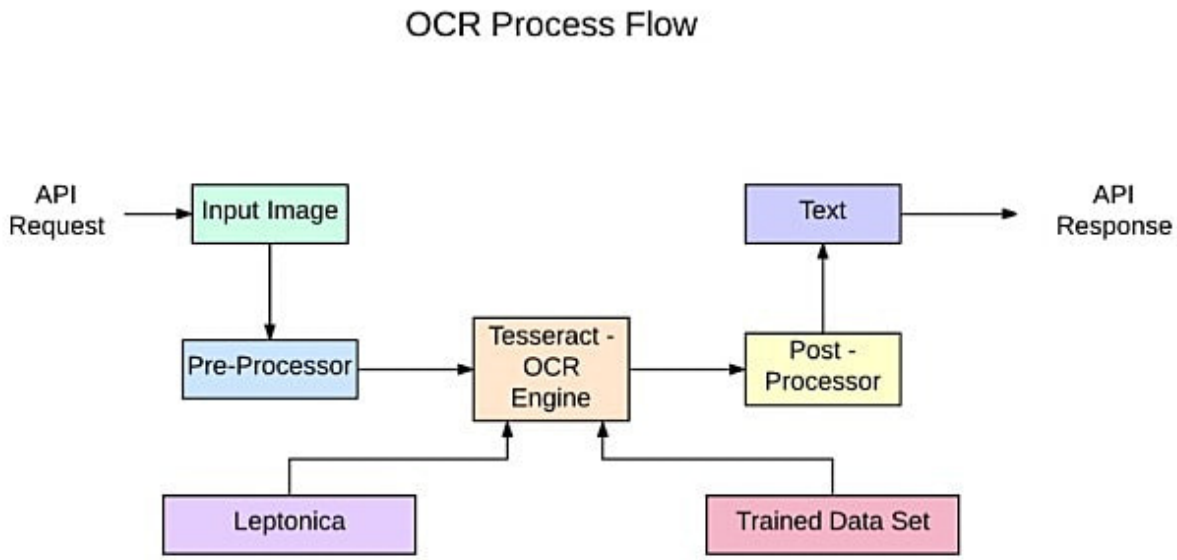


Figure 3:

5.2



Solution and Technical Architecture

Implementation:

The convolutional and max-pool layer: The feature is extracted from the input image and got max pooled to reduce the dimensionality of the image without any changes in the extracted feature. The dense layer: The flattened output from the max-pool layer is fed to a feed-forward neural network and backpropagation applied to every iteration of training. The output layer: The nodes in this stratum are referred to as output units. It gives us access to the neural network's final prediction, which may be used to make final predictions.

Methodology:

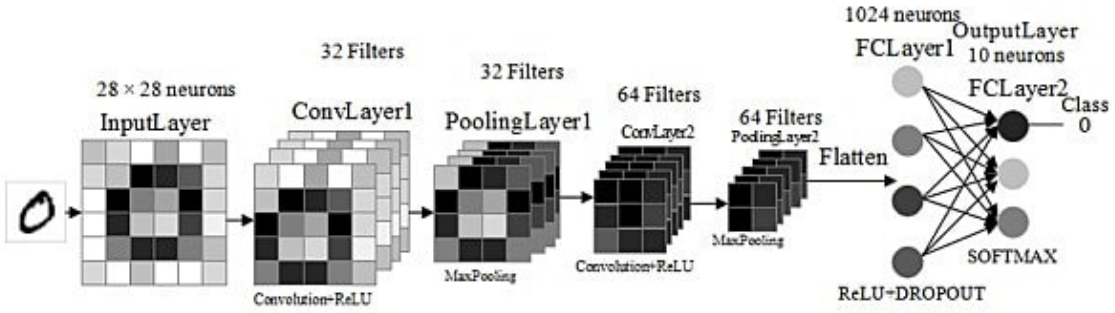
A neural network with one hidden layer and 100 activation units has been put into practice (excluding bias units). The features (X) and labels (Y) were retrieved after the

data was loaded from a mat file. To prevent overflow during computation, features are then scaled into a range of [0,1] by dividing by 255. 10,000 testing cases and 60,000 training examples make up the data. With the training data, feedforward is used to calculate the hypothesis, and backpropagation is then used to lower the error between the layers. To combat overfitting, the regularization parameter lambda is set to 0.1. To identify the model that fits the situation the optimizer runs for 70 times.

Process:

After receiving an input, neural networks change it using a number of hidden layers. Each group of neurons in a hidden layer is completely linked to every other neuron in the layer above it. One layer of neurons has perfect independence from one another. The "output layer" is the final layer to be fully connected.

Convolutional Neural Network Architecture:



Convolutional Layer:

The foundational component of a CNN is the convolutional layer. The parameters of the layer are a set of learnable filters (or kernels) that cover the entire depth of the input volume but have a narrow receptive field. As a result, the network picks up filters that turn on when it detects a certain kind of feature at a particular spatial location in the input.

Feature Extraction:

All neurons in a feature share the same weights. In this way all neurons detect the same feature at different positions in the input image. Reduce the number of free parameters.

Tensorflow:

An open-source machine learning library for both research and production is called TensorFlow. TensorFlow provides developers of all skill levels with APIs for desktop, mobile, web, and cloud applications.

Pytorch:

PyTorch is a machine learning framework based on the Torch library, used for applications such as computer vision and natural language processing, originally developed by Meta AI and now part of the Linux Foundation umbrella. Although the Python interface is more polished and the primary focus of development, PyTorch also has a C++ interface.

5.3 User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
1111		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-1
		USN-3	As a user, I can register for the application through Facebook	I can register & access the dashboard with Facebook Login	Low	Sprint-2
		USN-4	As a user, I can register for the application through Gmail	I can register the application with Gmail	Medium	Sprint-2
	Login	USN-5	As a user, I can log into the application by entering email & password	I can login to the application	High	Sprint-1
	Home	USN-6	As a user, I can view the application's home page where I can read the instructions to use this application	I can read instructions also and the home page is user-friendly.	Low	Sprint-1
	Upload Image	USN-7	As a user, I can able to input the images of digital documents to the application	As a user, I can able to input the images of digital documents to the application	High	Sprint-3
	Predict	USN-8	As a user I can able to get the recognised digit as output from the images of digital documents or images	I can access the recognized digits from digital document or images	High	Sprint-3
		USN-9	As a user, I will train and test the input to get the maximum accuracy of output.	I can able to train and test the application until it gets maximum accuracy of the result.	Medium	Sprint-4
Customer (Web user)	Accessibility	USN-10	As a user, I can use the web application virtually anywhere.	I can use the application in any device with a browser	Medium	Sprint-4

6. PROJECT PLANNING AND SCHEDULING

6.1 Sprint planning and Estimation

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	7	6 Days	24 Oct 2022	29 Oct 2022	7	29 Oct 2022
Sprint-2	7	6 Days	31 Oct 2022	05 Nov 2022	7	05 Nov 2022
Sprint-3	7	6 Days	07 Nov 2022	12 Nov 2022	7	12 Nov 2022
Sprint-4	7	6 Days	14 Nov 2022	19 Nov 2022	7	19 Nov 2022

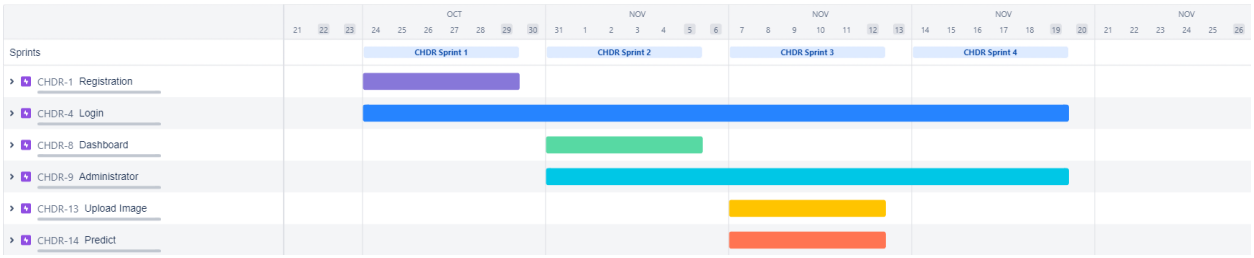
Velocity:

Imagine we have a 10-day sprint duration, and the velocity of the team is 20 (points per sprint). Let's calculate the team's average velocity (AV) per iteration unit

(story points $AV = \frac{sprint\ duration}{velocity} = \frac{20}{10} = 2$

6.3 Reports from JIRA

Figure 1:



7. CODING AND SOLUTIONING

Feature1: Authentication

i) Registeration:

For a security purpose user needs to register before recognizing the digits.

ii) Login:

User needs to login before getting the recognizing of digits. By this it gives a secured way of authentication.

iii) Logout:

After completion of the digits recognition and got desired outputand user can logout from the website.

Feature 2: Recognition

i) User can give input as images.

ii) By this digits can be easily recognized and it can be in JPEG or PNG format.

iii) Data accuracy is fully protected.

8.TESTING

8.1 Test Cases

#import the library for loading the model

```
from keras.models import load_model
import matpotlib.pyplot as plt

#Load the Model
model = load_model("MNIST.h5")
from PIL import Image
import numpy as np
import matplotlib.pyplot as plt

#Preview the Sample data
img = Image.open(r"D:\Images\image_1.jpg")
plt.imshow(img)

#Test the model with User's Input
for index in range(1,5):
    img = Image.open(r"D:\Images\image_" +str(index)+'.jpg').convert("L")
    img = img.resize((28,28))
    img2arr = np.array(img)
    img2arr = img2arr.reshape(1,28,28,1)
    y_pred = model.predict(img2arr)
    print(y_pred)
    print(f"The image_{index} predicted as value {np.argmax(y_pred)}")
```

8.2 USER ACCEPTANCE

Purpose of Document:

The purpose of this document is to briefly explain the test coverage and open issues of the [A Novel Method for Handwritten Digit Recognition System] project at the time of the release to User Acceptance Testing (UAT).

Defect Analysis

This report shows the number of resolved or closed bugs at each severity level, and how they were resolved.

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	17	3	4	3	27
Duplicate	1	3	2	1	7
External	2	5	3	1	11
Fixed	9	4	7	28	48
Not Reproduced	0	0	1	0	1
Skipped	1	0	1	1	3
Won't Fix	0	0	0	1	1
Totals	29	15	18	35	98

Test Case Analysis

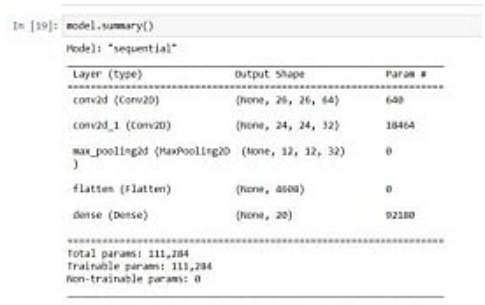
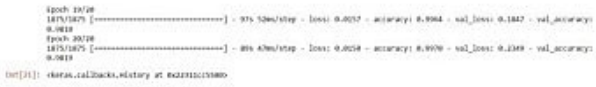
This report shows the number of test cases that have passed, failed, and untested.

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	7	0	0	7
Client Application	33	0	0	33
Security	10	0	0	10
Outsource Shipping	6	0	0	6
Exception Reporting	17	0	0	17
Final Report Output	8	0	0	8
Version Control	5	0	0	5

9. RESULTS

9.1 Performance Metrices

#code to perform
metrics = model.evaluate(X_test, y_test, verbose=0)
print("Metrics(Test loss & Test Accuracy): ")
print(metrics)

S.No	Parameter	Values	Screenshot
1.	Model Summary	Layers: Conv2d Conv2d_1 max_pooling2d (MaxPooling2D) flatten (Flatten) dense (Dense)	
2.	Accuracy	Training Accuracy – 0.9970 Validation Accuracy 0.9819	

10. ADVANTAGES and DISADVANTAGES

ADVANTAGES

i) The text should be extracted from the writtend file.
ii) User can recognized the hardly written text.
iii) User can get access to the Alphabets, Numbers, and symbols.

DISADVANTAGES

i) User needs a camera for scanning the file.
ii) Internet Connection is mandatory.

11. CONCLUSION

An implementation of Handwritten Digit Recognition using Deep Learning has been implemented in this paper. Additionally, some of the most widely used Machine Learning algorithms i.e. CNN using Tensorflowhave been trained and tested on the same data to draw a comparison as to why we require deep learning methods in critical applications like Handwritten Digit Recognition. In this paper, I have shown that that using Deep Learning techniques, a very high amount of accuracy can be achieved. Using the Convolutional Neural Network with Keras and Theano as backend, I am able to get an accuracy of 95.72%. Every tool has its own complexity and accuracy. Although, we see that the complexity of the code and the process is bit more as compared to normal Machine Learning algorithms but looking at the accuracy achieved, it can be said that it is worth it. Also, the current implementation is done only using the CPU Thus we settled on classifying a given handwritten digit image as the required digit using three different algorithms and consequently testing its accuracy. In future we are planning to further explore the topic to recognize people's handwriting.

12. FUTURE SCOPE

It will need to take a close look at the system and should look for improvements for the future. From the net-file, the system was able to produce an image-file. The image-file produced showed the recognized number This part will also need more improvements. Apart from the above problems and parts that need improvements, the overall recognition system was successful.

13. APPENDIX
Source Code

app/urls.py:

```
from django.urls import path
from .views import home_view, extract_view, result_view

app_name = "app"

urlpatterns = [
    path('', home_view, name="home-view"),
    path('extract/', extract_view, name="extract-view"),
    path('result/<str:value>/<str:accuracy>/', result_view, name="result-view"),
]
```

app/views.py:

```
from django.shortcuts import render
from django.http import HttpResponse, HttpResponseRedirect
from django.urls import reverse
from .forms import PhotoForm
from .recognizer import recognize
from django.contrib.auth.decorators import login_required

# Create your views here.

def home_view(request, *args, **kwargs):
    context = {}
    return render(request, "app/home.html", context)

@login_required
def extract_view(request, *args, **kwargs):
    my_form = PhotoForm()
    context = {
        "form": my_form,
    }
    if request.method =="GET":
        pred = ()

    if request.method == "POST":
        img = request.FILES.get('photo')
        pred, others, img_name = recognize(img)
        return HttpResponseRedirect(reverse("app:result-view", kwargs={"value": str(pred[0]),
```

```
"accuracy": str(pred[1])}))
        #

    return render(request, "app/extract.html", context)

@login_required
def result_view(request, *args, **kwargs):
    print(request.path)
    var = request.path
    l = var.split("/")
    print(l)
    value = l[2]
    accuracy = l[3]
    return render(request, "app/result.html", { "val": value, "acc": accuracy})
```

app/recognizer.py:

```
import os
import random
import string
from pathlib import Path
import numpy as np
from keras.models import load_model
from PIL import Image, ImageOps


def random_name_generator(n):
    return ''.join(random.choices(string.ascii_uppercase + string.digits, k=n))

def recognize(image):
    model=load_model(Path("./model/model.h5"))

    img = Image.open(image).convert("L")
    img_name = random_name_generator(10) + '.jpg'

    if not os.path.exists(f"./static/data/"):
        os.mkdir(os.path.join('./static/', 'data'))
    img.save(Path(f"./static/data/{img_name}"))

    img = ImageOps.grayscale(img)
    img = ImageOps.invert(img)
    img = img.resize((28, 28))
```

```
    img2arr = np.array(img)
    img2arr = img2arr / 255.0
    img2arr = img2arr.reshape(1, 28, 28, 1)

    results  = model.predict(img2arr)
    best = np.argmax(results,axis = 1)[0]

    pred = list(map(lambda x: round(x*100, 2), results[0]))

    values = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
    others = list(zip(values, pred))

    best = others.pop(best)

    return best, others, img_name
```

accounts/urls.py:

```
from django.urls import path
from.views import register_view, login_view, logout_view

app_name = "accounts"

urlpatterns = [
   path('register/', register_view, name = "register"),
   path('login/', login_view, name = "login"),
   path('logout/', logout_view, name = "logout"),

]
```

accounts/views.py:

```
from django.http import HttpResponseRedirect
from django.shortcuts import redirect, render
from django.contrib.auth.forms import AuthenticationForm, UserCreationForm
from django.contrib.auth import authenticate, login, logout, get_user_model
from django.urls import reverse

from .forms import LoginForm, RegisterForm

User = get_user_model()

def register_view(request):
```

```
    form = RegisterForm(request.POST or None)
    if form.is_valid():
        username = form.cleaned_data.get("username")
        email = form.cleaned_data.get("email")
        password = form.cleaned_data.get("password1")
        try:
            user = User.objects.create_user(username, email, password)
        except:
            user = None
        if user != None:
            return HttpResponseRedirect(reverse('accounts:login'))
    context = {
        "form": form
    }
    return render(request, "accounts/registration.html", context)

def login_view(request):
    form = LoginForm(request.POST or None)
    if form.is_valid():
        username = form.cleaned_data.get("username")
        password = form.cleaned_data.get("password")
        user = authenticate(request, username=username, password=password)
        if user != None:
            login(request, user)
            return HttpResponseRedirect(reverse('app:home-view'))
    context = {
        "form": form
    }
    return render(request, "accounts/login.html", context)

def logout_view(request):
    logout(request)
    return HttpResponseRedirect(reverse('accounts:login'))
```

hdr/urls.py:

```
from django.contrib import admin
from django.conf import settings
from django.urls import path, include
from django.conf.urls.static import static
#helo
```

```
urlpatterns = [
    path('', include("app.urls")),
    path('admin/', admin.site.urls),
    path('accounts/', include("accounts.urls")),
]

urlpatterns += static(settings.STATIC_URL, document_root=settings.STATIC_ROOT)
urlpatterns += static(settings.MEDIA_URL, document_root=settings.MEDIA_ROOT)
```

hdr/settings.py:

```
from pathlib import Path
import os

LOGIN_URL = "/accounts/login"

BASE_DIR = Path(__file__).resolve().parent.parent



SECRET_KEY = 'django-insecure-ah3f3p)q&k@)3pt((f*n&z-)d=+pwn!87k$ga$*wq3(f0!qg*+'

DEBUG = True

ALLOWED_HOSTS = []

INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'app',
    'crispy_forms',
    'accounts',
]

MIDDLEWARE = [
    'django.middleware.security.SecurityMiddleware',
    'django.contrib.sessions.middleware.SessionMiddleware',
```

```
    'django.middleware.common.CommonMiddleware',
    'django.middleware.csrf.CsrfViewMiddleware',
    'django.contrib.auth.middleware.AuthenticationMiddleware',
    'django.contrib.messages.middleware.MessageMiddleware',
    'django.middleware.clickjacking.XFrameOptionsMiddleware',
]

ROOT_URLCONF = 'hdr.urls'

TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': [BASE_DIR /'templates'],
        'APP_DIRS': True,
        'OPTIONS': {
            'context_processors': [
                'django.template.context_processors.debug',
                'django.template.context_processors.request',
                'django.contrib.auth.context_processors.auth',
                'django.contrib.messages.context_processors.messages',
            ],
        },
    },
]

WSGI_APPLICATION = 'hdr.wsgi.application'

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.sqlite3',
        'NAME': BASE_DIR / 'db.sqlite3',
    }
}

AUTH_PASSWORD_VALIDATORS = [
    {
        'NAME': 'django.contrib.auth.password_validation.UserAttributeSimilarityValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.MinimumLengthValidator',
    },
```

```
    {
        'NAME': 'django.contrib.auth.password_validation.CommonPasswordValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.NumericPasswordValidator',
    },
]

LANGUAGE_CODE = 'en-us'

TIME_ZONE = 'UTC'

USE_I18N = True

USE_L10N = True

USE_TZ = True

STATIC_URL = 'static/'

STATIC_ROOT = "/var/www/example.com/static/"

STATICFILES_DIRS = [
    BASE_DIR / "static",
    '/var/www/static/',
]

MEDIA_URL = '/media/'

MEDIA_ROOT = os.path.join(BASE_DIR, 'media/')

DEFAULT_AUTO_FIELD = 'django.db.models.BigAutoField'
```

templates/accounts/login.html:

```
{% extends 'base.html' %} {% block content %} {% load crispy_forms_tags %}
<div class="container ml-auto col-md-3 mt-5">
    <div class="card shadow-lg p-3 mb-5 bg-body rounded p-3">
        <div class="text-center mx-5 my-2">
            <h3>Enter Credentials to Login</h3>
        </div>
        <form method="POST">
```

```
        {% csrf_token %}
        {{ form.username|as_crispy_field }}
        {{ form.password|as_crispy_field }}
        <div class="text-end">
          <button class="btn btn-success mt-3" type="submit">Signin</button>
        </div>
      </form>
      <div class="text-end">
        <p class="mt-2">
          Don't have an account?
          <a href="{% url 'accounts:register' %}">signup</a>
        </p>
      </div>
    </div>
  </div>
{% endblock content %}
```

templates/accounts/registration.html:

```
{% extends 'base.html' %} {% block content %} {% load crispy_forms_tags %}

<div class="container ml-auto col-md-3 mt-5">
  <div class="card shadow-lg p-3 mb-5 bg-body rounded p-3">
    <div class="text-center mx-5 my-2">
      <h3>Enter Credentials to Signup</h3>
    </div>
    <form method="POST">
      {% csrf_token %}
      {{ form.username|as_crispy_field }}
      {{ form.email|as_crispy_field }}
      {{ form.password1|as_crispy_field }}
      {{ form.password2|as_crispy_field }}
      <div class="float-end mt-3">
        <button class="btn btn-success" type="submit">Signup</button>
      </div>
    </form>
  <div class="text-end mt-3">
    <p>
      Already have an account? <a href="{% url 'accounts:login' %}">Signin</a>
    </p>
  </div>
</div>
```

```
</div>
{% endblock content %}
```

templates/app/home.html:

```
{% extends 'base.html' %}

{% block content %}
<div class="container text-center" style=" display: flex; justify-content: center; align-items: center;
width: 1920px; height: 650px;">
   <h1 style="font-size: 4em;">A Novel Method for Handwritten Digit Recognition System</h1>
</div>

{% endblock content %}
```

templates/app/extract.html:

```
{% extends 'base.html' %}
{% load crispy_forms_tags %}
{% block content %}
{% comment %} <div class="container text-center" style="padding: 100px; padding-bottom: 0px
margin-top: 230px; margin-bottom: 0px justify-content: center; align-items: center; width: 800px;
height: 400px;"> {% endcomment %}
<div class="container col-md-3 text-center" style="align-items: center; justify-content: center; ">
   {% comment %} <h1 class="my-5" style="font-size: 3em;">Upload an image to Convert !!!</h1>
   <form method="POST" enctype="multipart/form-data"> {% csrf_token %}
      <div class="input-group">
         <input type="file" class="form-control" id="inputGroupFile04" aria-
describedby="inputGroupFileAddon04" aria-label="Upload">
         <button class="btn btn-outline-secondary" type="submit"
id="inputGroupFileAddon04">Button</button>
      </div>
   </form> {% endcomment %}

   <h1 class="my-5" style="font-size: 3em;">Upload an image to Convert !!!</h1>
   <form action="./" method="POST" enctype="multipart/form-data" class="row row-cols-lg-auto
align-items-center"> {% csrf_token %}
         {{ form.photo.label_tag }}
         {{ form.photo|as_crispy_field }}
         <button type="submit" class="btn btn-primary">Recognize</button>
   </form>
```

```
</div>
{% endblock content %}
```

templates/app/result.html:

```
{% extends 'base.html' %}
{% block content %}
<div class="my-5 container col-md-3 text-center" style="align-items: center; justify-content:
center; ">
   <h1>Value: {{ val }}</h1>
   <h1>Accuracy: {{ acc }}</h1>
   <a href="{% url 'app:extract-view' %}"><button class="btn btn-primary mt-
3">Back</button></a>
</div>
{% endblock content %}
```

model building code:

```
model = Sequential()
model.add(Conv2D(64,(3,3),
          input_shape=(28,28,1),
          activation="relu"))
model.add(Conv2D(32,(3,3),
          activation="relu"))
model.add(MaxPooling2D((2,2)))
model.add(Flatten())
model.add(Dense(number_of_classes,
                activation="softmax"))
model.compile(loss="categorical_crossentropy",
              optimizer="Adam",
              metrics=["accuracy"])
model.fit(X_train,y_train,
          epochs=20,
          validation_data=(X_test,y_test),
          batch_size=32)
metrics=model.evaluate(X_test,y_test,verbose=0)
print("Metrics(Test Loss & Test Accuracy):")
print(metrics)
prediction = model.predict(X_test[:4])
print(prediction)
model.save("model.h5")
```

GITHUB LINK: ***https://github.com/IBM-EPBL/IBM-Project-35209-1660282729***

DEMO LINK: