**A Novel Method for Handwritten Digit Recognition System**

USING AI

*A Project report submitted in partial fulfilment of 7th semester in degree of*

BACHELOR OF TECHNOLOGY IN INFORMATION TECHNOLOGY

Submitted by

Team ID: PNT2022TMID44093

AAYESHA Y (723719205001)

ABINAYA R (723719205002)

AKALYADEVI B (723719205005)

JENISHA J (723719205019)



DEPARTMENT OF INFORMATION TECHNOLOGY

VSB COLLEGE OF ENGINEERING TECHNICAL CAMPUS, COIMBATORE

ANNA UNIVERSITY: CHENNAI 600025

**NOV-2022**

VSB COLLEGE OF ENGINEERING TECHNICAL CAMPUS, COIMBATORE (Affiliated College of Anna University, Chennai)

****

**BONAFIDE CERTIFICATE**

Certified that this project report ”**A NOVEL METHOD** **FOR HANDWRITTEN RECOGNITION SYSTEM**” is the Bonafide record work done by MS.**AAYESHA (723719205001),MS.ABINAYA(723719205002),MS.AKALYA DEVI(723719205005),MS.JENISHA(723719205019),**for **IBM- NALAIYATHIRAN** in VII semester of B.TECH., degree course in **Information Technology** branch during the academic year of 2022 - 2023.

**STAFF INCHARGE EVALUATOR**

**MS.D.S.Jayakumari., Mr.Murugadass.,**

**ABSTRACT**

The Handwritten digits are not always of the same size, width, orientation and justified to margins as they differ from writing of person to person.The similarity between digits such as 1 and 7, 5 and 6, 3 and 8, 2 and 7 etc. So, classifying between these numbers is also a major problem for computers.The uniqueness and variety in the handwriting of different individuals also influence the formation and appearance of the digits.Digit recognition plays an important role in the modern world.It can solve more complex problems and makes humans job easier.This type of system can be widely used in the world to recognize zip code or postal code for mail sorting In Banking Sector too where more handwritten numbers are involved like account number, figure ofcash and checks.Postal department and courier services can easily find the digits written.Old people who will have eye sight issues with handwritten digits.Baking sector and Postal sector by providing the services. Handwriting recognition is one of the compelling research works going on because every individual in this world has their own style of writing. It is the capability of the computer to identify and understand handwritten digits or characters automatically. Because of the progress in the field of science and technology, everything is being digitalized to reduce human effort. Hence, there comes a need for handwritten digit recognition in many real-time applications. MNIST data set is widely used for this recognition process and it has 70000 handwritten digits. We use Artificial neural networks to train these images and build a deep learning model. Web application is created where the user can upload an image of a handwritten digit. this image is analyzed by the model and the detected result is returned on to UI.

**TABLE OF CONTENTS**

**1.INTRODUCTION**

1.1 PROJECT OVERVIEW

1.2 PURPOSE

**2.LITERATURE SURVERY**

2.1 EXISTING PROBLEM

2.2 REFERENCES

2.3 PROBLEM STATEMENT DEFINITION

**3.IDEATION AND PROPOSED SOLUTION**

3.1 EMPATHY MAP CANVAS

3.2 IDEATION AND BRAINSTROMING

3.3 PROPOSED SOLUTION

3.4 PROBLEM SOLUTION FIT

**4.REQUIREMENTS ANALYSIS**

4.1 FUNCTIONAL REQUIREMENTS

4.2 NON-FUNCTIONAL REQUIREMENTS

**5.PROJECT DESIGN**

5.1 DATAFLOW DIAGRAM

5.2 SOLUTION AND TECHNICAL ARCHITECTURE

5.3 USER STORIES

**6.PROJECT PLANNING AND SCHEDULING**

6.1 SPRINT PLANNING AND ESTIMATION

6.2 SPRINT DELIVERY SCHEDULE

6.3 REPORTS FROM JIRA

**7.CODING AND SOLUTIONING**

7.1 FEATURE 1

7.2 FEATURE 2

7.3 DATABASE SCHEMA

**8.TESTING**

8.1 TESTCASES

8.2 USER ACCEPTANCE TESTING

**9.RESULTS**

9.1 PERFORMANCE METRICS

**10.ADVANTAGES AND DISADVANTAGES**

**11.CONCLUSION**

**12.FUTURE SCOPE**

**13.APPENDIX**

**SOURCE CODE**

**GITHUB & PROJECT DEMO LINK**

**CHAPTER 1**

**INTRODUCTION**

Category: Artificial Intelligence

Skills Required:Python,CNN,IBM Cloud,

IBM Watson Studio,IBM Cloudant DB,Deep Learning,Python-Flask

**a. PROJECT OVERVIEW**

Machine learning and deep learning play an important role in computer applied science and artificial intelligence. With the use of deep learning , machine learning can be reduced in recognition, predictions and many more areas.

Handwritten Digit Recognition is the ability of computer systems to recognize handwritten digits from various sources, such as images, documents, among other examples. This project aims to let users take advantage of machine learning to reduce manual tasks in recognizing digits.

**b. PURPOSE**

Digit recognition systems are capable of recognizing the digits from different sources like emails,bank quench, papers,images, etc. and in different real-world scenarios for online handwriting recognition on computer tablets or system, recognize number plates of vehicles, processing bank quench amounts, numeric entries in forms filled up by hand (tax forms) and so on.

**CHAPTER 2**

**LITERATURE SURVEY**

**TITLE :** Handwritten Character Recognition

**AUTHOR :** Ayush Purohit

# DESCRIPTION :

Handwriting recognition has gained a lot of attention in the fieldof pattern recognition and machine learning due to its application in various fields. Optical Character Recognition (OCR) and Handwritten Character Recognition (HCR) has specific domain to apply. Various techniques have been proposed to for character recognition in handwriting recognition system. Even though, sufficient studies and papers describes the techniques for converting textual content from a paper document into machine readable form. In coming days, character recognition system might serve as a key factor to create a paperless environment by digitizing and processing existing paper documents.

**PUBLISHED IN :** 2016

**TITLE :** Automatic Handwritten Digit Recognition

**AUTHOR :** Akkireddy Challa

# DESCRIPTION :

The main purpose of this thesis is to build an automatic handwritten digit recognition method for the recognition of connected handwritten digit strings. To accomplish the recognition task, first, the digits were segmented into individual digits. Then, a digit recognition module is employed to classify each segmented digit completing the handwritten digit string recognition task. In this study, different machine learning methods, which are SVM, ANN and CNN architectures are used to achieve high performance on the digit string recognition problem. In these methods, images of digit strings are trained with the SVM, ANN and CNN model with HOG feature vectors and Deep learning methods structure by sliding a fixed size window through the images labeling each sub-image as a part of a digit or not. After the completion of the segmentation, to achieve the complete recognition of handwritten digits.

**PUBLISHED IN :** 2019

**TITLE :** Handwritten Numeral Recognition

**AUTHOR :** Stuti Asthana

# DESCRIPTION :

An extensive literature review on Neural Network based numeric recognition by describing the survey of some research articles have been involved for better analysis in order to enhance the system efficiency. Handwritten Numeric Recognition is very interesting area of Pattern Recognition and it deals with Offline Handwriting Recognition. Handwriting Recognition has kept on continuing as a method for correspondence, gathering, recording and transmitting data in everyday life since the hundreds of years even with the appearance of the new advancements. Machine recognition has numerous functional applications, perusing manually written postal envelopes, sum written in bank checks, bill handling, government records, business frames, signature confirmation, disconnected from the net archive acknowledgment and so on. This Paper portrays the bestin class study of the work accomplished for the Numeric recognition.

**PUBLISHED IN :** 2017

**TITLE :** Neural Network Based Handwritten Digit Recognition

**AUTHOR :** Ankit Sharma

# DESCRIPTION :

Recognition of handwritten character is a difficult task in the field of image processing, artificial intelligence since the handwriting varies from person to person. In proposed paper, we are training the neural network to recognize the off-line strategies for the isolated handwritten character (0 to 9). This work improves the character recognition and pre- processing of the Character is done by image rendering, character extraction and training and testing steps. The proposed method is based on the use of linear regression algorithm to classify the characters and is used to train the given dataset. After training a network performance curve is generated along with the individual required characters. In given system, numerical character is represented by binary numbers that are used as input then they are fed to an ANN. Neural network followed by the linear regression, algorithm which compromises Training.

**PUBLISHED IN :** 2016

**TITLE :** Handwritten Optical Character Recognition

**AUTHOR :** Jamshed Memon

# DESCRIPTION :

Given the ubiquity of handwritten documents in human transactions, Optical Character Recognition (OCR) of documents have invaluable practical worth. Optical character recognition is a science that enables to translate various types of documents or images into analyzable, editable and searchable data. During last decade, researchers have used artificial intelligence/machine learning tools to automatically analyze handwritten and printed documents in order to convert them into electronic format. The objective of this review paper is to summarize research that has been conducted on character recognition of handwritten documents and to provide research directions. In this Systematic Literature Review (SLR) we collected, synthesized and analyzed research articles on the topic of handwritten OCR (and closely related topics) which were published between year 2000 to 2019. We followed widely used electronic databases by following pre-defined review protocol. Articles were searched using keywords, forward reference searching and backward reference searching in order to search all the articles related to the topic. After carefully following study selection process 176 articles were selected for this SLR. This review article serves the purpose of presenting state of the art results and techniques on OCR and also provide research directions by highlighting research gaps.

**PUBLISHED IN :** 2020

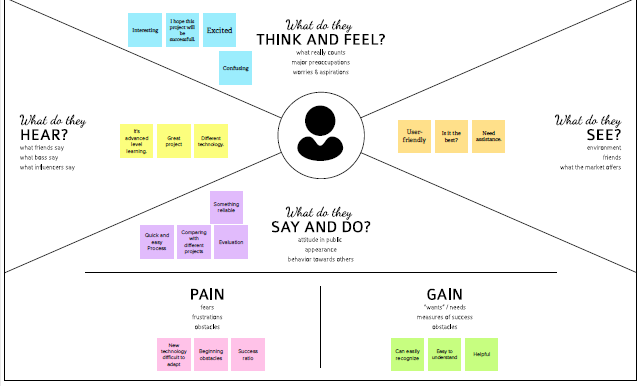
**PROBLEM STATEMENT DEFINITION**

For years,the traffic department has been combating traffic law violators. These offenders endanger not only their own lives, but also the lives of other individuals. Punishing these offenders is critical to ensuring that others do not become like them. Identification of these offenders is next to impossible because for the average individual to write down the license plate of a reckless driver. Therefore, the goal of this project is to help the traffic department identify these offenders and reduce traffic violations as a result.

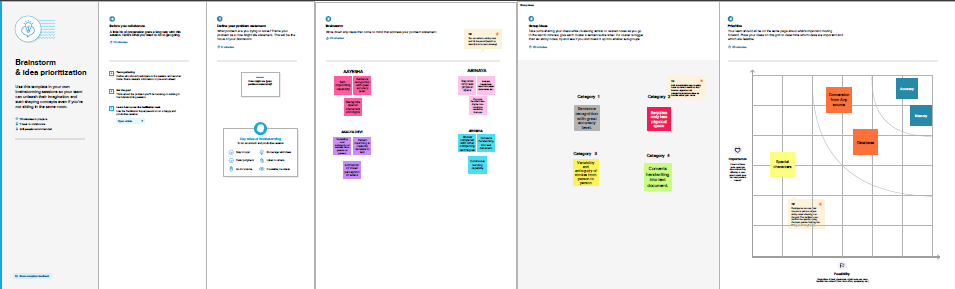
**CHAPTER-3**

**IDEATION AND PROPOSEDSOLUTION**

**3.1 EMPATHY MAP CANVAS**

****

**IDEATION & BRAINSTORMING**

****

**PROPOSED SOLUTION**

**A NOVEL METHOD FOR HANDWRITTEN DIGIT RECOGNITION SYSTEM**

**PROBLEM STATEMENT:**

The handwritten digit recognition is the capability of computer applications to recognize the human handwritten digits.

It is a hard task for the machine because handwritten digits are not perfect and can be made with many different shapes and sizes. The handwritten digit recognition system is a way to tackle this problem which uses the image of a digit and recognizes the digit present in the image. Convolutional Neural Network model created

using **PyTorch library** over the **MNIST dataset** to **recognize handwritten digits**.

**IDEA/SOLUTION DESCRIPTION:**

* MNIST database contains 60,000 training images of handwritten digits from zero to nine and 10,000 images for testing.
* We will create our CNN model. It works better for data that are represented as grid structures; this is the reason why CNN works well for image classification problems.

**SOLUTION ARCHITECTURE**

Input data

**v**

Classification And Recognition

Segmentation and Scaling

Feature Extraction

Image preprocessing

**NOVELTY/UNIQUENESS:**

* Handwritten digit recognition using MNIST dataset is a major project made with the help of neural network. It basically detects the scanned images of handwritten digits.
* We have taken this a step further where are handwritten digit recognition system not only detects the scanned images of handwritten digits but also allows writing digits on the screen with the help of an Integrated GUI for recognition.

**SOCIAL IMPACT/CUSTOMER SATISFACTION**:

Digital Recognition is nothing other than recognizing or identifying digits in any document. The framework of digital recognition is simply the operation of the machine to prepare or interpret digits. Handwritten Digit Recognition is the power of computers to translate handwritten digits from a variety of sources such as text messages, bank checks, papers, photos, etc. method

With the use of in-depth learning methods, human efforts can be reduced in perception, learning, perception and in too many regions. Using in-depth learning, the computer learns to perform distinctive functions in images or content anywhere

accuracy, in addition to the performance of the human level. The digital recognition model uses large data sets to detect digits from different sources.

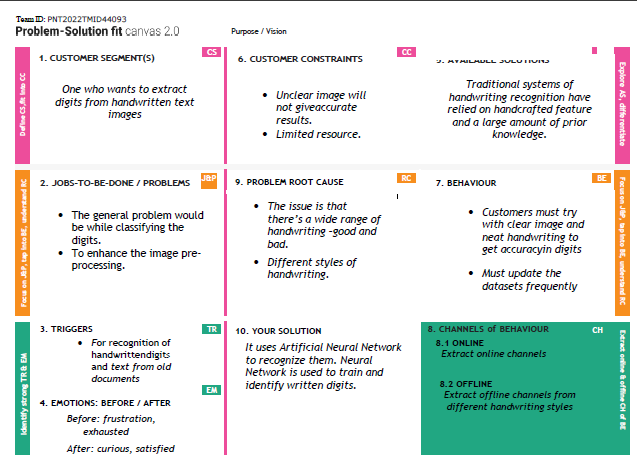
**BUSINESS MODEL (FINANCIAL BENEFIT):**

* Handwritten digit recognition refers to a model’s (machine’s) capacity to detect any handwritten digits from various sources, such as photographs, papers, and touch displays, and classify them into ten specified categories 0-9.
* Several ways and algorithms are used to recognize handwritten digits, such as Deep Learning/CNN, SVM (Support Vector Machine), Gaussian Naive Bayes, KNN (K-Nearest Neighbour), Decision Trees, Random Forests, etc.
* We used the CNN (Convolutional Neural network) algorithm to recognize handwritten digits in this project.

**SCALABILITY OF SOLUTION:**

* The variations of accuracies for handwritten digit were observed for 15 epochs by varying the hidden layers using CNN model and MNIST digit dataset.
* The maximum accuracy in the performance was found 99.64% and the total lowest test loss is 0.0239 approximately.

**PROBLEM SOLUTION FIT**

****

**CHAPTER – 4**

**REQUIREMENT ANALYSIS**

**FUNCTIONAL REQUIREMENTS**

|  |  |  |
| --- | --- | --- |
| **FR.NO** | **FUNCTIONAL REQUIREMENTS** | **SUB REQUIREMENTS** |
| FR-1 | Model Creation | Get access the MNIST dataset |
| Analyze the dataset |
| Deﬁne a CNN model |
| Train and Test the Model |
| FR-2 | Application Development | Create a website to let the user recognize handwritten digits. |
| Create a home page to upload images |
| Create a result page to display the results |
| Host the website to let the users use it from anywhere |
| FR-3 | Input Image Upload | Let users upload images of various formats. |
| Let users upload images of various size |
| Prevent users from uploading unsupported image formats |
| Pre-Process the image to use it on the model |
| Create a database to store all the input images |

|  |  |  |
| --- | --- | --- |
| FR-4 | Display Results | Display the result from the model |
| Display input image |
| Display accuracy the result |
| Display other possible predictions with their respective accuracy |

**NON-FUNCTIONAL REQUIREMENTS**

|  |  |  |
| --- | --- | --- |
| **NFR** | **NON-FUNCTIONAL REQUIREMENTS** | **DESCRIPTION** |
| NFR-1 | Usability | The application must be usable in all devices |
| NFR-2 | Security | The application must protect user uploaded image |
| NFR-3 | Reliability | The application must give an accurate result as much as possible |
| NFR-4 | Performance | The application must be fast and quick to load up |
| NFR-5 | Availability | The application must be available to use all the time |
| NFR-6 | Scalability | The application must scale along with the user base |

CHAPTER 6

PROJECT PLANNING AND SCHEDULING

SPRINT PLANNING AND ESTIMATION

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Sprint** | **Functional**  **Requirement (Epic)** | **User Story Number** | **User Story / Task** | **Story Points** | **Priority** | **Team Members** |
| Sprint-1 | Registration | USN-1 | As a user, I can register for the application by entering my email, password, and confirming my password. | 2 | High | Aayesha.Y |
| Sprint-1 | Login | USN-2 | As a user, I can log into the application by entering email & password | 1 | High | Akalya Devi.B |
| Sprint-2 | Upload Image of  digital document | USN-3 | As a user, I can able to input the images of  digital documents to the application | 2 | Medium | Jenisha.J |
| Sprint-2 | Prediction | USN-4 | As a user, I can predict the word | 1 | Medium | Abinaya.R |
| Sprint-3 | Upload Image of Handwritten document | USN-5 | As a user, I can able to input the images of the handwritten documents or images to the application | 2 | High | Aayesha.Y |
| Sprint-3 | Recognize text | USN-6 | As a user, I can able to choose the font of the text to be displayed | 1 | Medium | Akalya Devi.B |
| Sprint-4 | Recognize digit | USN-7 | As a user I can able to get the recognized  digit as output from the images of digital documents or images | 1 | Medium | Jenisha.J |
| Sprint-4 | Recognize digit | USN-8 | As a user I can able to get the recognized  digit as output from the images of handwritten documents or images | 2 | High | Aayesha.Y |

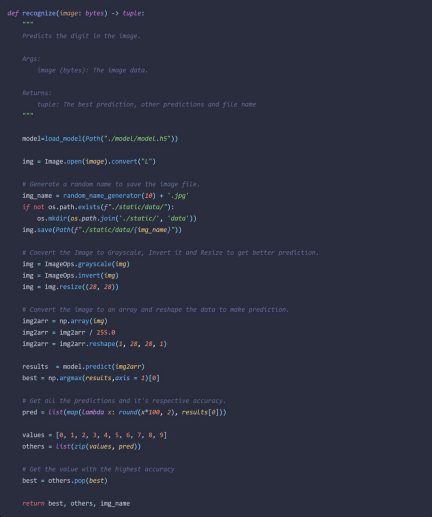
**PROJECT TRACKER, VELOCITY & BURNDOWN CHART:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **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 | 2 | 6 Days | 24 Oct 2022 | 29 Oct 2022 | 2 | 29 Oct 2022 |
| Sprint-2 | 2 | 6 Days | 31 Oct 2022 | 05 Nov 2022 | 2 | 05 Nov 2022 |
| Sprint-3 | 2 | 6 Days | 07 Nov 2022 | 12 Nov 2022 | 2 | 12 Nov 2022 |
| Sprint-4 | 2 | 6 Days | 14 Nov 2022 | 19 Nov 2022 | 2 | 19 Nov 2022 |

**CHAPTER 7**

**CODING & SOLUTION**

****

****

**CHAPTER 8**

**TESTING**

**8.2 USER ACCEPTANCE TESTING**

**8.2.1 DEFECT ANALYSIS**

| Resolution | Severity 1 | Severity 2 | Severity 3 | Severity 4 | Total |
| --- | --- | --- | --- | --- | --- |
| By Design | 1 | 0 | 1 | 0 | 2 |
| Duplicate | 0 | 0 | 0 | 0 | 0 |
| External | 0 | 0 | 3 | 0 | 3 |
| Fixed | 3 | 2 | 0 | 2 | 7 |
| Not Reproduced | 0 | 1 | 0 | 0 | 1 |
| Skipped | 0 | 0 | 0 | 0 | 0 |
| Won’t Fix | 1 | 0 | 1 | 1 | 3 |
| Total | 5 | 3 | 5 | 3 | 16 |

**8.2.2 TEST CASE ANALYSIS**

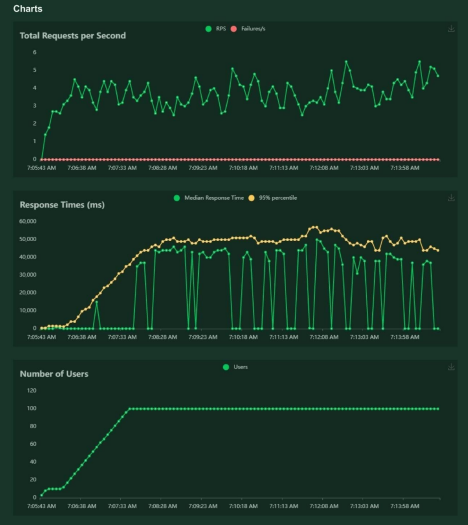
| Section | Total  Cases | Not  Tested | Fail | Pass |
| --- | --- | --- | --- | --- |
| Client Application | 10 | 0 | 3 | 7 |
| Security | 2 | 0 | 1 | 1 |
| Performance | 3 | 0 | 1 | 2 |
| Exception Reporting | 2 | 0 | 0 | 2 |

**CHAPTER 9**

**RESULTS**

9.1 PERFORMANCE METRICS



****

**CHAPTER 10**

**ADVANTAGES & DISADVANTAGES**

**ADVANTAGES**

● Decreases manual work

● More accurate prediction

● Ability to handle bulk of data

● Portable and Scalable

**DISADVANTAGES**

● Difficulty in handling complex data

● Data must be in digital format

● Need of high performance server for faster predictions

**CHAPTER 11**

**CONCLUSION**

This project demonstrated a web application that uses machine learning to recognize handwritten numbers. Flask, HTML, CSS, JavaScript, and a few other technologies were used to create this project. The model predicts the handwritten digit using a CNN network. During testing, the model achieved a 99.61% recognition rate. The proposed project isscalable and can easily handle a huge number of users.Since it is a web application, it is compatible with any device that can run a browser.This project is useful in real-world scenarios such as recognizing numberplates of vehicles, processing bank quench amounts, numeric entries in forms filledup by hand (tax forms) and so on. There is so much room for improvement, which can be implemented in subsequent versions.

**CHAPTER 12**

**FUTURE SCOPE**

This project is far from complete and there is a substantial amount of room foimprovement.

Some of the improvements that can be made to this project are as follows:

● Add support to detect from digits multiple images and save the results ● Add support to detect multiple digits

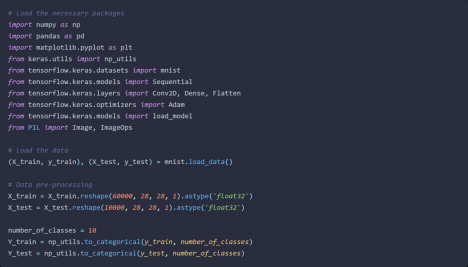
● Improve model to detect digits from complex images

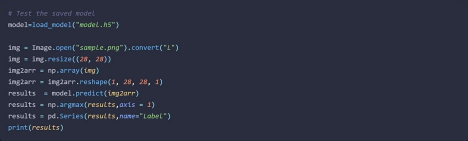
● Add support to different languages to help users from all over the world

This project has endless potential and can always be enhanced to become better.Implementing this concept in the real world will benefit several industries and reduce the workload on many workers,enhancing overall work efficiency.

**APPENDIX**

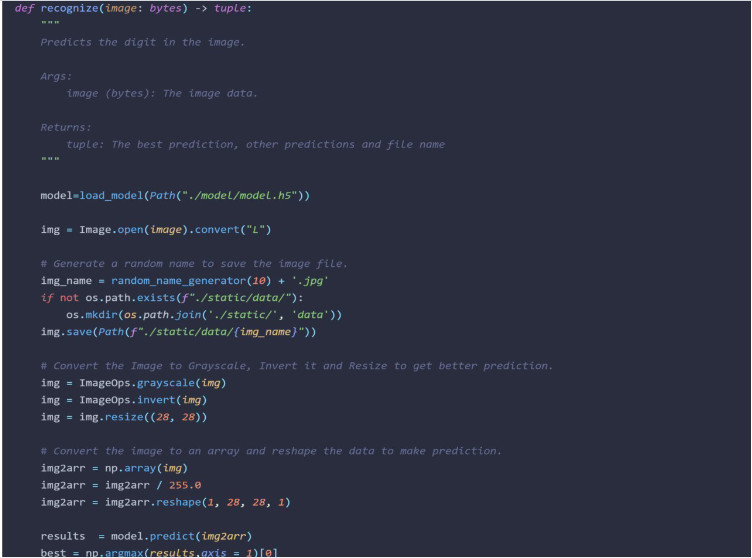
**SOURCE CODE**

****

**FLASK APP**

****

****

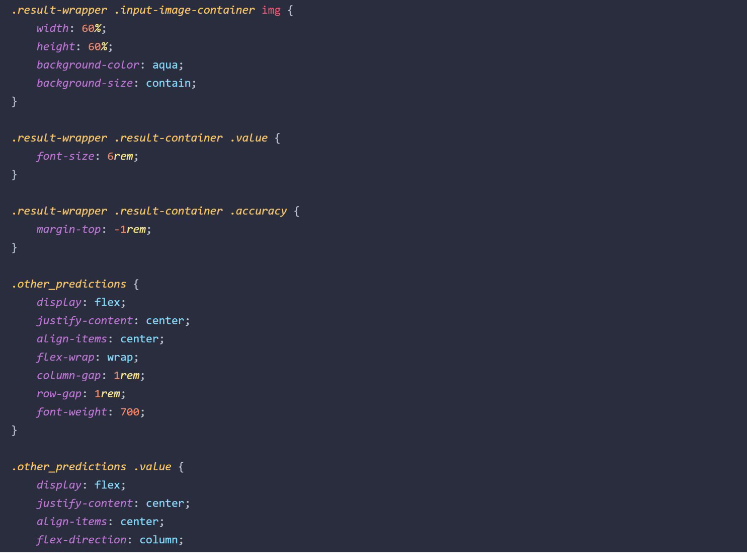
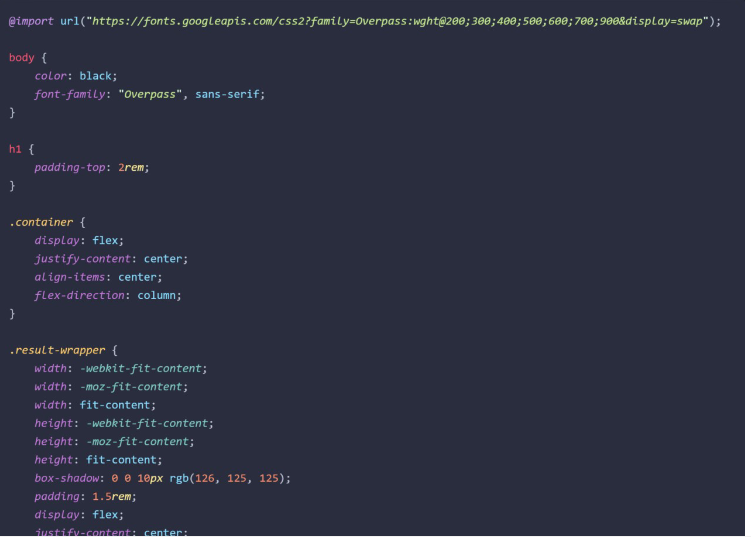
****

|  |  |
| --- | --- |
|  | **HTML HOMEPAGE**  <html> |
|  |  |
|  | <head> |
|  | <title>Handwritten digit recognition</title> |
|  |  |
|  | <meta name="viewport" content="width=device-width"> |
|  |  |
|  | <link href="https://fonts.googleapis.com/css2?family=Prompt:wght@600&display=swap" rel="stylesheet"> |
|  | <link href="https://fonts.googleapis.com/css2?family=Varela+Round&display=swap" rel="stylesheet"> |
|  | <link href="https://fonts.googleapis.com/css2?family=Source+Code+Pro:wght@500&display=swap" rel="stylesheet"> |
|  | <link href="https://fonts.googleapis.com/css?family=Calistoga|Josefin+Sans:400,700|Pacifico&display=swap" rel="stylesheet"> |
|  |  |
|  | <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css" integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T" crossorigin="anonymous"> |
|  | <link rel="stylesheet" type= "text/css" href= "style.css"> |
|  |  |
|  | <script src="https://kit.fontawesome.com/b3aed9cb07.js" crossorigin="anonymous"></script> |
|  |  |
|  | <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js" integrity="sha384-q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo" crossorigin="anonymous"></script> |
|  | <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js" integrity="sha384-UO2eT0CpHqdSJQ6hJty5KVphtPhzWj9WO1clHTMGa3JDZwrnQq4sF86dIHNDz0W1" crossorigin="anonymous"></script> |
|  | <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js" integrity="sha384-JjSmVgyd0p3pXB1rRibZUAYoIIy6OrQ6VrjIEaFf/nJGzIxFDsf4x0xIM+B07jRM" crossorigin="anonymous"></script> |
|  | <script src="https://cdn.jsdelivr.net/npm/@tensorflow/tfjs@latest"></script> |
|  | <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.6.1/dist/css/bootstrap.min.css"> |
|  | <script src="https://cdn.jsdelivr.net/npm/jquery@3.6.0/dist/jquery.slim.min.js"></script> |
|  | <script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.1/dist/umd/popper.min.js"></script> |
|  | <script src="https://cdn.jsdelivr.net/npm/bootstrap@4.6.1/dist/js/bootstrap.bundle.min.js"></script> |
|  |  |
|  | </head> |
|  | <style> |
|  | body{ |
|  | background-image: url('backimg.jpg'); |
|  | background-repeat: no-repeat; |
|  | background-size: cover; |
|  | } |
|  | </style> |
|  |  |
|  | <script> |
|  | function preview() { |
|  | frame.src=URL.createObjectURL(event.target.files[0]); |
|  | } |
|  |  |
|  | $(document).ready(function() { |
|  | $('#clear\_button').on('click', function() { |
|  | $('#image').val(''); |
|  | $('#frame').attr('src',""); |
|  | }); |
|  | }); |
|  |  |
|  | </script> |
|  | <body> |
|  | <h1>HandWritten Digit Recognition System</h1> |
|  | <div class="container p-3 my-3 bg-black text-white"> |
|  | <p>Handwritten Digit Recognition is a technology that is much needed in this world as of Today.This Digit Recognition System is used to recognize the digits from different sources like email, posts, cheque etc. Before proper implementation of this technology we have relied on writing text with our own hands which can result in error.It's difficult to store and access physical data with efficiency.The project presents in representing the recognization of handwritten digits (0 - 9) from the famous MNIST dataset. Here we will be using AlexNet which is an architecture of Convolutional Neural Network.</p> |
|  | </div> |
|  | <section id="content"> |
|  |  |
|  | <div class="leftside"> |
|  | <form action="/predict" method="POST" enctype="multipart/form-data"> |
|  | <label>Select a image:</label> |
|  | <input id="image" type="file" name="image" accept="image/png, image/jpeg" onchange="preview()"><br><br> |
|  | <img id="frame" width="100px" height="100px"/> |
|  | <div class="buttons\_div"> |
|  | <button type="submit" class="btn btn-light" onclick="myfunc()">Predict</button> |
|  | <script> |
|  | function myfunc() |
|  | { |
|  | window.location.href="C:\IBM project\predict.html"; |
|  | } |
|  | </script> |
|  | <button type="button" class="btn btn-light">&nbsp Clear &nbsp</button> |
|  | </div> |
|  | </form> |
|  | </div> |
|  | </section> |
|  |  |
|  | </body> |
|  |  |
|  | </html>  **PREDICT.HTML**   |  | | --- | | <!DOCTYPE html> | |  | <html lang="en"> | |  | <head> | |  | <meta charset="UTF-8"> | |  | <title>Prediction</title> | |  | </head> | |  |  | |  | <style> | |  | body{ | |  | background-image: url('backimg.jpg'); | |  | background-repeat: no-repeat; | |  | background-size: cover; | |  | } | |  |  | |  | #rectangle{ | |  | width:400px; | |  | height:150px; | |  | background-color: #000000; | |  | border-radius: 15px; | |  | position:absolute; | |  | box-shadow: 0px 0px 10px 5px grey; | |  | top:25%; | |  | left:50%; | |  | transform:translate(-50%,-50%); | |  | } | |  |  | |  | #head{ | |  | text-align: center; | |  | font-size: 30px; | |  | margin: 0 auto; | |  | padding: 3% 5%; | |  | font-family: Arial, Helvetica, sans-serif; | |  | color: white; | |  | } | |  |  | |  | #num{ | |  | font-size: 50px; | |  | } | |  |  | |  | </style> | |  |  | |  | <body> | |  |  | |  | <div id="rectangle"> | |  | <h1 id="head">Predicted Number : <br><center id="num">{{num}}</center></h1> | |  | </div> | |  |  | |  | </body> | |  | </html> | |

****

HOME PAGE (JS)



****

****

**APP.PY PAGE:**

|  |
| --- |
| importnumpyasnp |
|  | import os | |
|  | from PIL import Image | |
|  | from flask import Flask, request, render\_template, url\_for | |
|  | from werkzeug.utils import secure\_filename, redirect | |
|  | #from gevent.pywsgi import WSGIServer | |
|  | from keras.models import load\_model | |
|  | from keras.preprocessing import image | |
|  | from flask import send\_from\_directory | |
|  |  | |
|  | UPLOAD\_FOLDER = 'C:\IBM project\data' | |
|  |  | |
|  |  | |
|  | app = Flask(\_\_name\_\_) | |
|  | app.config['UPLOAD\_FOLDER'] = UPLOAD\_FOLDER | |
|  |  | |
|  | model = load\_model("./models/mnistCNN.h5") | |
|  |  | |
|  |  | |
|  | @app.route('/') | |
|  | def index(): | |
|  | return render\_template('home.html') | |
|  |  | |
|  |  | |
|  | @app.route('/predict', methods=['GET', 'POST']) | |
|  | def upload(): | |
|  | if request.method == "POST": | |
|  | f = request.files["image"] | |
|  | filepath = secure\_filename(f.filename) | |
|  | f.save(os.path.join(app.config['UPLOAD\_FOLDER'], filepath)) | |
|  |  | |
|  | upload\_img = os.path.join(UPLOAD\_FOLDER, filepath) | |
|  | img = Image.open(upload\_img).convert("L") # convert image to monochrome | |
|  | img = img.resize((28, 28)) # resizing of input image | |
|  |  | |
|  | im2arr = np.array(img) # converting to image | |
|  | im2arr = im2arr.reshape(1, 28, 28, 1) # reshaping according to our requirement | |
|  |  | |
|  | pred = model.predict(im2arr) | |
|  |  | |
|  | num = np.argmax(pred, axis=1) # printing our Labels | |
|  |  | |
|  | return render\_template('predict.html', num=str(num[0])) | |
|  |  | |
|  |  | |
|  | if \_\_name\_\_ == '\_\_main\_\_': | |
|  | app.run(debug=True, threaded=False)import numpy as np | |
|  | import os | |
|  | from PIL import Image | |
|  | from flask import Flask, request, render\_template, url\_for | |
|  | from werkzeug.utils import secure\_filename, redirect | |
|  | #from gevent.pywsgi import WSGIServer | |
|  | from keras.models import load\_model | |
|  | from keras.preprocessing import image | |
|  | from flask import send\_from\_directory | |
|  |  | |
|  | UPLOAD\_FOLDER = 'C:\IBM project\data' | |
|  |  | |
|  |  | |
|  | app = Flask(\_\_name\_\_) | |
|  | app.config['UPLOAD\_FOLDER'] = UPLOAD\_FOLDER | |
|  |  | |
|  | model = load\_model("./models/mnistCNN.h5") | |
|  |  | |
|  |  | |
|  | @app.route('/') | |
|  | def index(): | |
|  | return render\_template('home.html') | |
|  |  | |
|  |  | |
|  | @app.route('/predict', methods=['GET', 'POST']) | |
|  | def upload(): | |
|  | if request.method == "POST": | |
|  | f = request.files["image"] | |
|  | filepath = secure\_filename(f.filename) | |
|  | f.save(os.path.join(app.config['UPLOAD\_FOLDER'], filepath)) | |
|  |  | |
|  | upload\_img = os.path.join(UPLOAD\_FOLDER, filepath) | |
|  | img = Image.open(upload\_img).convert("L") # convert image to monochrome | |
|  | img = img.resize((28, 28)) # resizing of input image | |
|  |  | |
|  | im2arr = np.array(img) # converting to image | |
|  | im2arr = im2arr.reshape(1, 28, 28, 1) # reshaping according to our requirement | |
|  |  | |
|  | pred = model.predict(im2arr) | |
|  |  | |
|  | num = np.argmax(pred, axis=1) # printing our Labels | |
|  |  | |
|  | return render\_template('predict.html', num=str(num[0])) | |
|  |  | |
|  |  | |
|  | if \_\_name\_\_ == '\_\_main\_\_': | |
|  | app.run(debug=True, threaded=False)  **STYLE.CSS** | |
| #clear\_button{ | |
|  | | margin-left: 20px; |
|  | | font-weight: bold; |
|  | | color: rgb(14, 137, 195); |
|  | | } |
|  | |  |
|  | | #confidence{ |
|  | | font-family: cursive; |
|  | | margin-top: 7.5%; |
|  | | } |
|  | |  |
|  | | #content{ |
|  | | margin: 0 auto; |
|  | | padding: 2% 15%; |
|  | | padding-bottom: 0; |
|  | | } |
|  | |  |
|  | | .welcome{ |
|  | | text-align: center; |
|  | | position: relative; |
|  | | color: rgb(0, 32, 112); |
|  | | background-color: rgb(106, 173, 199); |
|  | | padding-top: 1%; |
|  | | padding-bottom: 1%; |
|  | | font-weight: bold; |
|  | | font-family: cursive; |
|  | | } |
|  | |  |
|  | |  |
|  | | #predict\_button{ |
|  | | margin-right: 20px; |
|  | | color: rgb(53, 75, 150); |
|  | | font-weight: bold; |
|  | | } |
|  | |  |
|  | | #prediction\_heading{ |
|  | | font-family: cursive; |
|  | | margin-top: 7.5%; |
|  | | } |
|  | |  |
|  | | #result{ |
|  | | font-size: 5rem; |
|  | | } |
|  | |  |
|  | | #title{ |
|  | | padding: 1.5% 15%; |
|  | | margin: 0 auto; |
|  | | text-align: center; |
|  | | } |
|  | |  |
|  | | .btn { |
|  | | font-size: 15px; |
|  | | padding: 10px; |
|  | | /\* -webkit-appearance: none; \*/ |
|  | | background: #eee; |
|  | | border: 1px solid #888; |
|  | | margin-top: 20px; |
|  | | margin-bottom: 20px; |
|  | | } |
|  | |  |
|  | | .buttons\_div{ |
|  | | margin-bottom: 30px; |
|  | | margin-right: 80px; |
|  | | } |
|  | |  |
|  | | .heading{ |
|  | | font-family:cursive; |
|  | | font-weight: 700; |
|  | | font-size: 2rem; |
|  | | display: inline; |
|  | | } |
|  | |  |
|  | | .leftside{ |
|  | | text-align: center; |
|  | | margin: 0 auto; |
|  | | margin-top: 2%; |
|  | |  |
|  | | } |
|  | |  |
|  | | #frame{ |
|  | | margin-right: 10%; |
|  | | } |
|  | |  |
|  | | .predicted\_answer{ |
|  | | text-align: center; |
|  | | margin: 0 auto; |
|  | | padding: 3% 5%; |
|  | | padding-top: 0; |
|  | |  |
|  | | } |
|  | |  |
|  | | h1{ |
|  | | text-align: center; |
|  | | color: aliceblue; |
|  | | padding: 100px 50px 65px 100px; |
|  | | } |
|  | |  |
|  | | @media (min-width: 720px) { |
|  | | .leftside{ |
|  | | padding-left: 10%; |
|  | | } |
|  | | } |

**FINAL CODE:**

**Import required libraries**\*

**from** keras.datasets **import** mnist

**import** matplotlib.pyplot **as** plt

**from** keras.utils **import** np\_utils

**from** tensorflow.keras.models **import** Sequential

**from** tensorflow.keras.layers **import** Conv2D,Dense,Flatten

**from** tensorflow.keras.optimizers **import** Adam

**Loading the dataset**

(X\_train,y\_train),(X\_test,y\_test) **=**mnist**.**load\_data()

print(X\_train**.**shape)

print(X\_test**.**shape)

print(y\_test**.**shape)

print(y\_train**.**shape)

Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/mnist.npz

11490434/11490434 [==============================] - 0s 0us/step

(60000, 28, 28)

(10000, 28, 28)

(10000,)

(60000,)

**Fetch the data from the dataset**

print("The label value is ",y\_test[9]) *#Value in y\_test*

plt**.**imshow(X\_test[9])

The label value is 9

print("The label value is ",y\_test[10]) *#Value in y\_test*

plt**.**imshow(X\_test[10])

The label value is 0

print("The label value is ",y\_test[23]) *#Value in y\_test*

plt**.**imshow(X\_test[23])

The label value is 5

**Applying one hot encoding**

X\_train**.**shape

(60000, 28, 28)

X\_test**.**shape

(10000, 28, 28)

X\_train1 **=** X\_train**.**reshape(60000, 28, 28, 1)**.**astype('float32')

X\_test1 **=** X\_test**.**reshape(10000, 28, 28, 1)**.**astype('float32')

number\_of\_classes**=** 12

y\_train1 **=** np\_utils**.**to\_categorical(y\_train,number\_of\_classes)

y\_test1 **=** np\_utils**.**to\_categorical(y\_test,number\_of\_classes)

**Encoding the value**

print("After encoding the value",y\_test[9] ,"become", y\_test1[9])

After encoding the value 9 become [0. 0. 0. 0. 0. 0. 0. 0. 0. 1. 0. 0.]

print("After encoding the value",y\_test[10] ,"become", y\_test1[10])

After encoding the value 0 become [1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]

print("After encoding the value",y\_test[23] ,"become", y\_test1[23])

After encoding the value 5 become [0. 0. 0. 0. 0. 1. 0. 0. 0. 0. 0. 0.]

**Add CNN layers**

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(Flatten())

model**.**add(Dense(number\_of\_classes, activation**=**"softmax"))

**Compile the model**

model**.**compile(loss**=**'categorical\_crossentropy', optimizer**=**"Adam", metrics**=**["accuracy"])

**Train the model**

model**.**fit(X\_train1, y\_train1, batch\_size**=**32, epochs**=**10, validation\_data**=**(X\_test1,y\_test1))

Epoch 1/10

1875/1875 [==============================] - 194s 103ms/step - loss: 0.2549 - accuracy: 0.9504 - val\_loss: 0.0902 - val\_accuracy: 0.9730

Epoch 2/10

1875/1875 [==============================] - 194s 104ms/step - loss: 0.0726 - accuracy: 0.9782 - val\_loss: 0.0820 - val\_accuracy: 0.9742

Epoch 3/10

1875/1875 [==============================] - 194s 103ms/step - loss: 0.0496 - accuracy: 0.9848 - val\_loss: 0.0817 - val\_accuracy: 0.9759

Epoch 4/10

1875/1875 [==============================] - 195s 104ms/step - loss: 0.0383 - accuracy: 0.9880 - val\_loss: 0.0899 - val\_accuracy: 0.9785

Epoch 5/10

1875/1875 [==============================] - 195s 104ms/step - loss: 0.0313 - accuracy: 0.9907 - val\_loss: 0.1074 - val\_accuracy: 0.9761

Epoch 6/10

1875/1875 [==============================] - 194s 104ms/step - loss: 0.0244 - accuracy: 0.9928 - val\_loss: 0.1156 - val\_accuracy: 0.9773

Epoch 7/10

1875/1875 [==============================] - 193s 103ms/step - loss: 0.0218 - accuracy: 0.9936 - val\_loss: 0.1221 - val\_accuracy: 0.9771

Epoch 8/10

1875/1875 [==============================] - 192s 102ms/step - loss: 0.0196 - accuracy: 0.9946 - val\_loss: 0.1727 - val\_accuracy: 0.9778

Epoch 9/10

1875/1875 [==============================] - 192s 103ms/step - loss: 0.0171 - accuracy: 0.9953 - val\_loss: 0.1468 - val\_accuracy: 0.9785

Epoch 10/10

1875/1875 [==============================] - 193s 103ms/step - loss: 0.0144 - accuracy: 0.9962 - val\_loss: 0.1704 - val\_accuracy: 0.9777

**Observing the metrics**

metrics **=** model**.**evaluate(X\_test1, y\_test1, verbose**=**0)

print("Checking the Metrics (Test Loss & Test Accuracy): ")

print(metrics)

Checking the Metrics (Test Loss & Test Accuracy):

[11.306961059570312, 0.12229999899864197]

**Test the model**

prediction **=** model**.**predict(X\_test1[:4])

print(prediction)

1/1 [==============================] - 0s 112ms/step

[[5.0968147e-06 3.2904151e-08 2.4547335e-08 3.8771137e-09 9.9999297e-01

2.1400561e-12 9.0379384e-09 1.9089430e-06 2.7502803e-10 2.1564152e-10

1.3407317e-11 2.5973085e-08]

[1.0000000e+00 2.0193573e-12 1.2437545e-10 3.0768805e-12 1.9168457e-14

6.3709477e-10 1.7837687e-10 2.4965596e-14 3.3803925e-13 1.5835364e-13

1.1105061e-16 9.6047545e-12]

[2.0305255e-05 1.8551295e-04 2.5913024e-03 1.0359057e-05 1.3580263e-04

2.6764979e-05 1.2820570e-02 6.5554171e-03 8.3878607e-02 8.9356083e-01

3.7450151e-05 1.7707223e-04]

[7.9626665e-07 2.8583373e-09 1.5453403e-09 3.7636035e-04 8.5368520e-06

9.6965458e-08 9.9961424e-01 1.3939068e-12 9.4559267e-09 9.6343879e-14

7.8885800e-18 1.2240818e-09]]

**import** numpy **as** np

print(np**.**argmax(prediction, axis**=**1))

print(y\_test1[:4])

[4 0 9 6]

[[0. 0. 0. 0. 0. 0. 0. 1. 0. 0. 0. 0.]

[0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0.]

[0. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]

[1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]]

**Save the model**

model**.**save("digit.h5")

**from** tensorflow.keras.models **import** load\_model

model**=**load\_model("digit.h5")

model**.**summary()

Model: "sequential"

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Layer (type) Output Shape Param #

=================================================================

conv2d (Conv2D) (None, 26, 26, 64) 640

conv2d\_1 (Conv2D) (None, 24, 24, 32) 18464

flatten (Flatten) (None, 18432) 0

dense (Dense) (None, 12) 221196

=================================================================

Total params: 240,300

Trainable params: 240,300

Non-trainable params: 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*# Saving in tar*

**!**tar -zcvf digit\_recognition.tgz digit.h5

digit.h5

**!**pip install watson-machine-learning-client

Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/

Collecting watson-machine-learning-client

Downloading watson\_machine\_learning\_client-1.0.391-py3-none-any.whl (538 kB)

|████████████████████████████████| 538 kB 4.4 MB/s

Requirement already satisfied: tabulate in /usr/local/lib/python3.7/dist-packages (from watson-machine-learning-client) (0.8.10)

Requirement already satisfied: tqdm in /usr/local/lib/python3.7/dist-packages (from watson-machine-learning-client) (4.64.1)

Collecting lomond

Downloading lomond-0.3.3-py2.py3-none-any.whl (35 kB)

Requirement already satisfied: requests in /usr/local/lib/python3.7/dist-packages (from watson-machine-learning-client) (2.23.0)

Requirement already satisfied: urllib3 in /usr/local/lib/python3.7/dist-packages (from watson-machine-learning-client) (1.24.3)

Collecting ibm-cos-sdk

Downloading ibm-cos-sdk-2.12.0.tar.gz (55 kB)

|████████████████████████████████| 55 kB 3.5 MB/s

Requirement already satisfied: pandas in /usr/local/lib/python3.7/dist-packages (from watson-machine-learning-client) (1.3.5)

Requirement already satisfied: certifi in /usr/local/lib/python3.7/dist-packages (from watson-machine-learning-client) (2022.9.24)

Collecting boto3

Downloading boto3-1.26.9-py3-none-any.whl (132 kB)

|████████████████████████████████| 132 kB 53.2 MB/s

Collecting s3transfer<0.7.0,>=0.6.0

Downloading s3transfer-0.6.0-py3-none-any.whl (79 kB)

|████████████████████████████████| 79 kB 6.4 MB/s

Collecting jmespath<2.0.0,>=0.7.1

Downloading jmespath-1.0.1-py3-none-any.whl (20 kB)

Collecting botocore<1.30.0,>=1.29.9

Downloading botocore-1.29.9-py3-none-any.whl (9.9 MB)

|████████████████████████████████| 9.9 MB 45.8 MB/s

Requirement already satisfied: python-dateutil<3.0.0,>=2.1 in /usr/local/lib/python3.7/dist-packages (from botocore<1.30.0,>=1.29.9->boto3->watson-machine-learning-client) (2.8.2)

Collecting urllib3

Downloading urllib3-1.26.12-py2.py3-none-any.whl (140 kB)

|████████████████████████████████| 140 kB 45.8 MB/s

Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.7/dist-packages (from python-dateutil<3.0.0,>=2.1->botocore<1.30.0,>=1.29.9->boto3->watson-machine-learning-client) (1.15.0)

Collecting ibm-cos-sdk-core==2.12.0

Downloading ibm-cos-sdk-core-2.12.0.tar.gz (956 kB)

|████████████████████████████████| 956 kB 55.8 MB/s

Collecting ibm-cos-sdk-s3transfer==2.12.0

Downloading ibm-cos-sdk-s3transfer-2.12.0.tar.gz (135 kB)

|████████████████████████████████| 135 kB 53.0 MB/s

Collecting jmespath<2.0.0,>=0.7.1

Downloading jmespath-0.10.0-py2.py3-none-any.whl (24 kB)

Collecting requests

Downloading requests-2.28.1-py3-none-any.whl (62 kB)

|████████████████████████████████| 62 kB 1.2 MB/s

Requirement already satisfied: charset-normalizer<3,>=2 in /usr/local/lib/python3.7/dist-packages (from requests->watson-machine-learning-client) (2.1.1)

Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.7/dist-packages (from requests->watson-machine-learning-client) (2.10)

Requirement already satisfied: numpy>=1.17.3 in /usr/local/lib/python3.7/dist-packages (from pandas->watson-machine-learning-client) (1.21.6)

Requirement already satisfied: pytz>=2017.3 in /usr/local/lib/python3.7/dist-packages (from pandas->watson-machine-learning-client) (2022.6)

Building wheels for collected packages: ibm-cos-sdk, ibm-cos-sdk-core, ibm-cos-sdk-s3transfer

Building wheel for ibm-cos-sdk (setup.py) ... done

Created wheel for ibm-cos-sdk: filename=ibm\_cos\_sdk-2.12.0-py3-none-any.whl size=73931 sha256=828cd7ebe3989eb3f0f89d8aa8b2672fdfedbacff67110754e1186bc114462b3

Stored in directory: /root/.cache/pip/wheels/ec/94/29/2b57327cf00664b6614304f7958abd29d77ea0e5bbece2ea57

Building wheel for ibm-cos-sdk-core (setup.py) ... done

Created wheel for ibm-cos-sdk-core: filename=ibm\_cos\_sdk\_core-2.12.0-py3-none-any.whl size=562962 sha256=e3e83fbd43e20a5e9f729519f4f078ad1ddd5e749e91026173e51feee7d799e8

Stored in directory: /root/.cache/pip/wheels/64/56/fb/5cd6f4f40406c828a5289b95b2752a4d142a9afb359244ed8d

Building wheel for ibm-cos-sdk-s3transfer (setup.py) ... done

Created wheel for ibm-cos-sdk-s3transfer: filename=ibm\_cos\_sdk\_s3transfer-2.12.0-py3-none-any.whl size=89778 sha256=45c0dc69fa9821741f923f81f07af8872d68c74bf9cbc2ee0dd7a6237a07a3d2

Stored in directory: /root/.cache/pip/wheels/57/79/6a/ffe3370ed7ebc00604f9f76766e1e0348dcdcad2b2e32df9e1

Successfully built ibm-cos-sdk ibm-cos-sdk-core ibm-cos-sdk-s3transfer

Installing collected packages: urllib3, requests, jmespath, ibm-cos-sdk-core, botocore, s3transfer, ibm-cos-sdk-s3transfer, lomond, ibm-cos-sdk, boto3, watson-machine-learning-client

Attempting uninstall: urllib3

Found existing installation: urllib3 1.24.3

Uninstalling urllib3-1.24.3:

Successfully uninstalled urllib3-1.24.3

Attempting uninstall: requests

Found existing installation: requests 2.23.0

Uninstalling requests-2.23.0:

Successfully uninstalled requests-2.23.0

Successfully installed boto3-1.26.9 botocore-1.29.9 ibm-cos-sdk-2.12.0 ibm-cos-sdk-core-2.12.0 ibm-cos-sdk-s3transfer-2.12.0 jmespath-0.10.0 lomond-0.3.3 requests-2.28.1 s3transfer-0.6.0 urllib3-1.26.12 watson-machine-learning-client-1.0.391

In [ ]:

**!**pip install ibm\_watson\_machine\_learning

Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/

Collecting ibm\_watson\_machine\_learning

Downloading ibm\_watson\_machine\_learning-1.0.257-py3-none-any.whl (1.8 MB)

|████████████████████████████████| 1.8 MB 4.3 MB/s

Requirement already satisfied: pandas<1.5.0,>=0.24.2 in /usr/local/lib/python3.7/dist-packages (from ibm\_watson\_machine\_learning) (1.3.5)

Requirement already satisfied: packaging in /usr/local/lib/python3.7/dist-packages (from ibm\_watson\_machine\_learning) (21.3)

Requirement already satisfied: requests in /usr/local/lib/python3.7/dist-packages (from ibm\_watson\_machine\_learning) (2.28.1)

Requirement already satisfied: certifi in /usr/local/lib/python3.7/dist-packages (from ibm\_watson\_machine\_learning) (2022.9.24)

Collecting ibm-cos-sdk==2.7.\*

Downloading ibm-cos-sdk-2.7.0.tar.gz (51 kB)

|████████████████████████████████| 51 kB 630 kB/s

Requirement already satisfied: urllib3 in /usr/local/lib/python3.7/dist-packages (from ibm\_watson\_machine\_learning) (1.26.12)

Requirement already satisfied: tabulate in /usr/local/lib/python3.7/dist-packages (from ibm\_watson\_machine\_learning) (0.8.10)

Requirement already satisfied: importlib-metadata in /usr/local/lib/python3.7/dist-packages (from ibm\_watson\_machine\_learning) (4.13.0)

Requirement already satisfied: lomond in /usr/local/lib/python3.7/dist-packages (from ibm\_watson\_machine\_learning) (0.3.3)

Collecting ibm-cos-sdk-core==2.7.0

Downloading ibm-cos-sdk-core-2.7.0.tar.gz (824 kB)

|████████████████████████████████| 824 kB 46.8 MB/s

Collecting ibm-cos-sdk-s3transfer==2.7.0

Downloading ibm-cos-sdk-s3transfer-2.7.0.tar.gz (133 kB)

|████████████████████████████████| 133 kB 39.6 MB/s

Requirement already satisfied: jmespath<1.0.0,>=0.7.1 in /usr/local/lib/python3.7/dist-packages (from ibm-cos-sdk==2.7.\*->ibm\_watson\_machine\_learning) (0.10.0)

Collecting docutils<0.16,>=0.10

Downloading docutils-0.15.2-py3-none-any.whl (547 kB)

|████████████████████████████████| 547 kB 54.5 MB/s

Requirement already satisfied: python-dateutil<3.0.0,>=2.1 in /usr/local/lib/python3.7/dist-packages (from ibm-cos-sdk-core==2.7.0->ibm-cos-sdk==2.7.\*->ibm\_watson\_machine\_learning) (2.8.2)

Requirement already satisfied: pytz>=2017.3 in /usr/local/lib/python3.7/dist-packages (from pandas<1.5.0,>=0.24.2->ibm\_watson\_machine\_learning) (2022.6)

Requirement already satisfied: numpy>=1.17.3 in /usr/local/lib/python3.7/dist-packages (from pandas<1.5.0,>=0.24.2->ibm\_watson\_machine\_learning) (1.21.6)

Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.7/dist-packages (from python-dateutil<3.0.0,>=2.1->ibm-cos-sdk-core==2.7.0->ibm-cos-sdk==2.7.\*->ibm\_watson\_machine\_learning) (1.15.0)

Requirement already satisfied: charset-normalizer<3,>=2 in /usr/local/lib/python3.7/dist-packages (from requests->ibm\_watson\_machine\_learning) (2.1.1)

Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.7/dist-packages (from requests->ibm\_watson\_machine\_learning) (2.10)

Requirement already satisfied: zipp>=0.5 in /usr/local/lib/python3.7/dist-packages (from importlib-metadata->ibm\_watson\_machine\_learning) (3.10.0)

Requirement already satisfied: typing-extensions>=3.6.4 in /usr/local/lib/python3.7/dist-packages (from importlib-metadata->ibm\_watson\_machine\_learning) (4.1.1)

Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in /usr/local/lib/python3.7/dist-packages (from packaging->ibm\_watson\_machine\_learning) (3.0.9)

Building wheels for collected packages: ibm-cos-sdk, ibm-cos-sdk-core, ibm-cos-sdk-s3transfer

Building wheel for ibm-cos-sdk (setup.py) ... done

Created wheel for ibm-cos-sdk: filename=ibm\_cos\_sdk-2.7.0-py2.py3-none-any.whl size=72563 sha256=659267c434e8e7c27acc7dda571c4454f1a639f6511dd150da1952a79c21e6cf

Stored in directory: /root/.cache/pip/wheels/47/22/bf/e1154ff0f5de93cc477acd0ca69abfbb8b799c5b28a66b44c2

Building wheel for ibm-cos-sdk-core (setup.py) ... done

Created wheel for ibm-cos-sdk-core: filename=ibm\_cos\_sdk\_core-2.7.0-py2.py3-none-any.whl size=501013 sha256=4df31bb57b8cc5edbe1054ca45f259583c0bedd53a63f1bdffa5b6207432b6e9

Stored in directory: /root/.cache/pip/wheels/6c/a2/e4/c16d02f809a3ea998e17cfd02c13369281f3d232aaf5902c19

Building wheel for ibm-cos-sdk-s3transfer (setup.py) ... done

Created wheel for ibm-cos-sdk-s3transfer: filename=ibm\_cos\_sdk\_s3transfer-2.7.0-py2.py3-none-any.whl size=88622 sha256=b0c77e9f333bbc5f59f67f5d8f87551684769077c751076e77c542812d38847e

Stored in directory: /root/.cache/pip/wheels/5f/b7/14/fbe02bc1ef1af890650c7e51743d1c83890852e598d164b9da

Successfully built ibm-cos-sdk ibm-cos-sdk-core ibm-cos-sdk-s3transfer

Installing collected packages: docutils, ibm-cos-sdk-core, ibm-cos-sdk-s3transfer, ibm-cos-sdk, ibm-watson-machine-learning

Attempting uninstall: docutils

Found existing installation: docutils 0.17.1

Uninstalling docutils-0.17.1:

Successfully uninstalled docutils-0.17.1

Attempting uninstall: ibm-cos-sdk-core

Found existing installation: ibm-cos-sdk-core 2.12.0

Uninstalling ibm-cos-sdk-core-2.12.0:

Successfully uninstalled ibm-cos-sdk-core-2.12.0

Attempting uninstall: ibm-cos-sdk-s3transfer

Found existing installation: ibm-cos-sdk-s3transfer 2.12.0

Uninstalling ibm-cos-sdk-s3transfer-2.12.0:

Successfully uninstalled ibm-cos-sdk-s3transfer-2.12.0

Attempting uninstall: ibm-cos-sdk

Found existing installation: ibm-cos-sdk 2.12.0

Uninstalling ibm-cos-sdk-2.12.0:

Successfully uninstalled ibm-cos-sdk-2.12.0

Successfully installed docutils-0.15.2 ibm-cos-sdk-2.7.0 ibm-cos-sdk-core-2.7.0 ibm-cos-sdk-s3transfer-2.7.0 ibm-watson-machine-learning-1.0.257

**Cloud deployment**

**from** ibm\_watson\_machine\_learning **import** APIClient

wml\_credentials **=** {

"url": "https://us-south.ml.cloud.ibm.com", *# example: "https://eu-gb.ml.cloud.ibm.com"*

"apikey":"Dt-EkyRgxXR--1mhO8JnCjRGR\_AvzoUpJQqbzFnWklU1"

}

client **=** APIClient(wml\_credentials)

client

Python 3.7 and 3.8 frameworks are deprecated and will be removed in a future release. Use Python 3.9 framework instead.

client**.**spaces**.**get\_details()

{'resources': [{'entity': {'compute': [{'crn': 'crn:v1:bluemix:public:pm-20:us-south:a/d74a81b5072a47ea932088f3c95b3d8d:ab0faf12-e097-475c-b555-79f9a13b440d::',

'guid': 'ab0faf12-e097-475c-b555-79f9a13b440d',

'name': 'Watson Machine Learning-lz',

'type': 'machine\_learning'}],

'description': '',

'name': 'digit\_deploy',

'scope': {'bss\_account\_id': 'd74a81b5072a47ea932088f3c95b3d8d'},

'stage': {'production': False},

'status': {'state': 'active'},

'storage': {'properties': {'bucket\_name': 'dede02b9-9740-4319-881c-f10ec6202dce',

'bucket\_region': 'us-south',

'credentials': {'admin': {'access\_key\_id': '9bfe67bd39f14cf5a8666e6188b02143',

'api\_key': '50PMGAm3eSnX\_G1VpNG6\_XJkwa-veWNCSyyru5ksZsWB',

'secret\_access\_key': 'b63dd4e1b1ecefdbdb32478174a66d411cd7a98519c8565b',

'service\_id': 'ServiceId-cf7956f9-5d6e-4fde-9bf9-c2d7d324d3d3'},

'editor': {'access\_key\_id': '9e76c7cc5b2c438396b834aaeda87df4',

'api\_key': 'EzZkGCey-46EuCVz3IztC8mnBFtuaD40Srufvm\_hFBUz',

'resource\_key\_crn': 'crn:v1:bluemix:public:cloud-object-storage:global:a/d74a81b5072a47ea932088f3c95b3d8d:b81cecb9-1689-4f8e-87d7-c70c72300b4e::',

'secret\_access\_key': '00cbee74cb48d75ca43d688108297703eea7ec26903a04cd',

'service\_id': 'ServiceId-725da56e-c4c0-4ecb-9d36-ea58872bbcf3'},

'viewer': {'access\_key\_id': '238ea99d20354b55b78c557fdb973972',

'api\_key': 'im-71co9LWBLEb295LCJlWx4AOejZgzJAxpq1SB9P5N9',

'resource\_key\_crn': 'crn:v1:bluemix:public:cloud-object-storage:global:a/d74a81b5072a47ea932088f3c95b3d8d:b81cecb9-1689-4f8e-87d7-c70c72300b4e::',

'secret\_access\_key': 'e3ca34240ce3757c166469ac364c6df4e20f464cbbad5d7a',

'service\_id': 'ServiceId-ca7069ff-0f0b-479e-af7f-4127e8cd1703'}},

'endpoint\_url': 'https://s3.us-south.cloud-object-storage.appdomain.cloud',

'guid': 'b81cecb9-1689-4f8e-87d7-c70c72300b4e',

'resource\_crn': 'crn:v1:bluemix:public:cloud-object-storage:global:a/d74a81b5072a47ea932088f3c95b3d8d:b81cecb9-1689-4f8e-87d7-c70c72300b4e::'},

'type': 'bmcos\_object\_storage'}},

'metadata': {'created\_at': '2022-11-13T07:31:19.376Z',

'creator\_id': 'IBMid-666002J5U4',

'id': '0d542d58-0e93-4b26-a2c6-156ce46c2f36',

'updated\_at': '2022-11-13T07:31:32.819Z',

'url': '/v2/spaces/0d542d58-0e93-4b26-a2c6-156ce46c2f36'}}]}

**def** guid\_space\_name(client,digit\_deploy):

space **=** client**.**spaces**.**get\_details()

**return**(next(item **for** item **in** space['resources'] **if** item['entity']['name']**==**digit\_deploy)['metadata']['id'])

space\_uid **=** guid\_space\_name(client,'digit\_deploy')

space\_uid

'0d542d58-0e93-4b26-a2c6-156ce46c2f36'

client**.**set**.**default\_space(space\_uid)

'SUCCESS'

client**.**software\_specifications**.**list()

----------------------------- ------------------------------------ ----

NAME ASSET\_ID TYPE

default\_py3.6 0062b8c9-8b7d-44a0-a9b9-46c416adcbd9 base

kernel-spark3.2-scala2.12 020d69ce-7ac1-5e68-ac1a-31189867356a base

pytorch-onnx\_1.3-py3.7-edt 069ea134-3346-5748-b513-49120e15d288 base

scikit-learn\_0.20-py3.6 09c5a1d0-9c1e-4473-a344-eb7b665ff687 base

spark-mllib\_3.0-scala\_2.12 09f4cff0-90a7-5899-b9ed-1ef348aebdee base

pytorch-onnx\_rt22.1-py3.9 0b848dd4-e681-5599-be41-b5f6fccc6471 base

ai-function\_0.1-py3.6 0cdb0f1e-5376-4f4d-92dd-da3b69aa9bda base

shiny-r3.6 0e6e79df-875e-4f24-8ae9-62dcc2148306 base

tensorflow\_2.4-py3.7-horovod 1092590a-307d-563d-9b62-4eb7d64b3f22 base

pytorch\_1.1-py3.6 10ac12d6-6b30-4ccd-8392-3e922c096a92 base

tensorflow\_1.15-py3.6-ddl 111e41b3-de2d-5422-a4d6-bf776828c4b7 base

autoai-kb\_rt22.2-py3.10 125b6d9a-5b1f-5e8d-972a-b251688ccf40 base

runtime-22.1-py3.9 12b83a17-24d8-5082-900f-0ab31fbfd3cb base

scikit-learn\_0.22-py3.6 154010fa-5b3b-4ac1-82af-4d5ee5abbc85 base

default\_r3.6 1b70aec3-ab34-4b87-8aa0-a4a3c8296a36 base

pytorch-onnx\_1.3-py3.6 1bc6029a-cc97-56da-b8e0-39c3880dbbe7 base

kernel-spark3.3-r3.6 1c9e5454-f216-59dd-a20e-474a5cdf5988 base

pytorch-onnx\_rt22.1-py3.9-edt 1d362186-7ad5-5b59-8b6c-9d0880bde37f base

tensorflow\_2.1-py3.6 1eb25b84-d6ed-5dde-b6a5-3fbdf1665666 base

spark-mllib\_3.2 20047f72-0a98-58c7-9ff5-a77b012eb8f5 base

tensorflow\_2.4-py3.8-horovod 217c16f6-178f-56bf-824a-b19f20564c49 base

runtime-22.1-py3.9-cuda 26215f05-08c3-5a41-a1b0-da66306ce658 base

do\_py3.8 295addb5-9ef9-547e-9bf4-92ae3563e720 base

autoai-ts\_3.8-py3.8 2aa0c932-798f-5ae9-abd6-15e0c2402fb5 base

tensorflow\_1.15-py3.6 2b73a275-7cbf-420b-a912-eae7f436e0bc base

kernel-spark3.3-py3.9 2b7961e2-e3b1-5a8c-a491-482c8368839a base

pytorch\_1.2-py3.6 2c8ef57d-2687-4b7d-acce-01f94976dac1 base

spark-mllib\_2.3 2e51f700-bca0-4b0d-88dc-5c6791338875 base

pytorch-onnx\_1.1-py3.6-edt 32983cea-3f32-4400-8965-dde874a8d67e base

spark-mllib\_3.0-py37 36507ebe-8770-55ba-ab2a-eafe787600e9 base

spark-mllib\_2.4 390d21f8-e58b-4fac-9c55-d7ceda621326 base

autoai-ts\_rt22.2-py3.10 396b2e83-0953-5b86-9a55-7ce1628a406f base

xgboost\_0.82-py3.6 39e31acd-5f30-41dc-ae44-60233c80306e base

pytorch-onnx\_1.2-py3.6-edt 40589d0e-7019-4e28-8daa-fb03b6f4fe12 base

pytorch-onnx\_rt22.2-py3.10 40e73f55-783a-5535-b3fa-0c8b94291431 base

default\_r36py38 41c247d3-45f8-5a71-b065-8580229facf0 base

autoai-ts\_rt22.1-py3.9 4269d26e-07ba-5d40-8f66-2d495b0c71f7 base

autoai-obm\_3.0 42b92e18-d9ab-567f-988a-4240ba1ed5f7 base

pmml-3.0\_4.3 493bcb95-16f1-5bc5-bee8-81b8af80e9c7 base

spark-mllib\_2.4-r\_3.6 49403dff-92e9-4c87-a3d7-a42d0021c095 base

xgboost\_0.90-py3.6 4ff8d6c2-1343-4c18-85e1-689c965304d3 base

pytorch-onnx\_1.1-py3.6 50f95b2a-bc16-43bb-bc94-b0bed208c60b base

autoai-ts\_3.9-py3.8 52c57136-80fa-572e-8728-a5e7cbb42cde base

spark-mllib\_2.4-scala\_2.11 55a70f99-7320-4be5-9fb9-9edb5a443af5 base

spark-mllib\_3.0 5c1b0ca2-4977-5c2e-9439-ffd44ea8ffe9 base

autoai-obm\_2.0 5c2e37fa-80b8-5e77-840f-d912469614ee base

spss-modeler\_18.1 5c3cad7e-507f-4b2a-a9a3-ab53a21dee8b base

cuda-py3.8 5d3232bf-c86b-5df4-a2cd-7bb870a1cd4e base

autoai-kb\_3.1-py3.7 632d4b22-10aa-5180-88f0-f52dfb6444d7 base

pytorch-onnx\_1.7-py3.8 634d3cdc-b562-5bf9-a2d4-ea90a478456b base

----------------------------- ------------------------------------ ----

Note: Only first 50 records were displayed. To display more use 'limit' parameter.

software\_space\_uid **=** client**.**software\_specifications**.**get\_uid\_by\_name('tensorflow\_rt22.1-py3.9')

software\_space\_uid

'acd9c798-6974-5d2f-a657-ce06e986df4d'

model\_details **=** client**.**repository**.**store\_model(model**=**'digit\_recognition.tgz',meta\_props**=**{

client**.**repository**.**ModelMetaNames**.**NAME:"DigitRecognition Model",

client**.**repository**.**ModelMetaNames**.**TYPE:"tensorflow\_2.7",

client**.**repository**.**ModelMetaNames**.**SOFTWARE\_SPEC\_UID:software\_space\_uid

})

model\_details

{'entity': {'hybrid\_pipeline\_software\_specs': [],

'software\_spec': {'id': 'acd9c798-6974-5d2f-a657-ce06e986df4d',

'name': 'tensorflow\_rt22.1-py3.9'},

'type': 'tensorflow\_2.7'},

'metadata': {'created\_at': '2022-11-15T06:32:10.093Z',

'id': '892f9dba-862a-4094-8701-f063b6fd66da',

'modified\_at': '2022-11-15T06:32:14.285Z',

'name': 'DigitRecognition Model',

'owner': 'IBMid-666002J5U4',

'resource\_key': '0961989d-65f0-4052-9429-70ed03c421fb',

'space\_id': '0d542d58-0e93-4b26-a2c6-156ce46c2f36'},

'system': {'warnings': []}}

model\_id **=** client**.**repository**.**get\_model\_id(model\_details)

model\_id

'892f9dba-862a-4094-8701-f063b6fd66da'

client**.**repository**.**download(model\_id,'Digit\_Recognition\_Model.tar.gb')

Successfully saved model content to file: 'Digit\_Recognition\_Model.tar.gb'

'/content/Digit\_Recognition\_Model.tar.gb'

\*\*-----------------THE END-----------------\*\*

|  |  |
| --- | --- |
| |  | | --- | | https://mail.google.com/mail/u/1/images/cleardot.gif | |