A PROJECT REPORT ON

“Open AI Handwritten Notes Assistant”

Submitted for fulfillment of award of the degree

BACHELOR OF TECHNOLOGY

(Computer Science & Engineering)

Akshay Sanghani (MITU20BTCS0024)
Archana Bidave (MITU20BTCSD010)
Swaroop Patil (MITU20BTCS0307)
Pragati Satpute (MITU20BTCSD023)

Under the guidance of

Prof . Abhishek Das

Department of Computer Science and Engineering

MIT School of Engineering

MIT Art, Design and Technology University, Pune

MAEER’s Rajbaug Campus, Loni-Kalbhor, Pune 412201

May, 2024

MIT SCHOOL OF ENGINEERING

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

MAEER's Rajbaug Campus, Loni-Kalbhor, Pune - 412201

CERTIFICATE

This is to certify that the project report entitled
"Open AI Handwritten Notes Assistant"

Submitted by
Akshay Sanghani (MITU20BTCS0024)
Archana Bidave (MITU20BTCSD010)
Swaroop Patil (MITU20BTCS0307)
Pragati Satpute (MITU20BTCSD023)

is a bonafide work carried out by students under the supervision of Prof. Abhishek Das and it is submitted towards the fulfillment of the requirement of MIT-ADT University, Pune for the award of the degree of Bachelor of Technology (Computer Science & Engineering)

Prof. Abhishek Das
Guide

Dr. Ganesh Pathak
Head of Department

Prof . Suresh Kapare
Project Coordinator

Prof. Dr. Rajneeshkaur Sachdeo
Director

Seal/Stamp of the College
Place : Date :

DECLARATION

We, the team members

Akshay Sanghani (MITU20BTCS0024)
Archana Bidave (MITU20BTCSD010)
Swaroop Patil (MITU20BTCS0307)
Pragati Satpute (MITU20BTCSD023)

Hereby declare that the project work incorporated in the present project entitled "**Open AI Handwritten Notes Assistant**" is original work. This work (in part or in full) has not been submitted to any University for the award or a Degree or a Diploma. We have properly acknowledged the material collected from secondary sources wherever required. We solely own the responsibility for the originality of the entire content.

Date: 17/05/2024

Name & Signature of the Team Members

Akshay Sanghani
Archana Bidave
Swaroop Patil
Pragati Satpute

Name & Signature of the Guide

Prof. Abhishek Das

Seal/Stamp of the College
Place: Loni Kalbhor
Date : 17/05/2024

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

MIT SCHOOL OF ENGINEERING,

RAJBAUG, LONI KALBHOR,

PUNE – 412201

EXAMINER'S APPROVAL CERTIFICATE

The project report entitled **"Open AI Handwritten Notes Assistant"** submitted By **AKSHAY SANGHANI , ARCHANA BIDAVE , PRAGATI SATPUTE, SWAROOP PATIL** in partial fulfilment for the award of the degree of **"Bachelor of Technology(Computer Science & Engineering) "** during the academic year 2023-24, of **MIT- ADT University, MIT School of Engineering, Pune,** is hereby approved.

Examiners:

1.

2.

ACKNOWLEDGEMENT

I express my profound thanks to my Guide **Prof. Abhishek Das** for her expert guidance, encouragement and inspiration during this project work.

I would like to thank **Prof. Suresh Kapare** Project Coordinator, Department Computer Science & Engineering for extending all support during the execution of the project work.

I sincerely thank to **Dr. Ganesh Pathak**, Head, Department of Computer Science & Engineering, MIT School of Engineering, MIT-ADT Univer- sity,Pune, for providing necessary facilities in completing the project.

I also thank all the faculty members in the Department for their support and advice.

Name and Enrollment no.
Akshay Sanghani (MITU20BTCS0024)
Archana Bidave (MITU20BTCSD010)
Swaroop Patil (MITU20BTCS0307)
Pragati Satpute (MITU20BTCSD023)

CONTENTS

CONTENTS

MODULE – 2

MODULE - 1 :

“Handwritten Character Recognition Using CNN”

I . Introduction :

Handwriting is the most typical and systematic way of recording facts and information. The handwriting of an individual is idiosyncratic and unique to individual people. The capability of software or a device to recognize and analyze human handwriting in any language is called a handwritten character recognition(HCR) system. Recognition can be performed from both online and offline handwriting. In recent years, applications of handwriting recognition are thriving, widely used in reading postal addresses, language translation, bank forms and check amounts, digital libraries, keyword spotting, and traffic sign detection. Image acquisition, preprocessing, segmentation, featureextraction, and classification are the typical processes of an HCR system, as shown in Figure 1. The initial step is to receive an image form of handwritten characters, which is recognized as image acquisition hat will proceed as an input to preprocessing. In preprocessing, distortions of the scanned images are removed and converted into binary images. Afterward, in the segmentationstep, each character is divided into sub images. Then, it willextract every characteristic of the features from each image of the character. This stage is especially important for the last step of the HCR system, which is called classification. Based on classification accuracy and different approaches to recognize the images, there are many classification methods, i.e., convolutional neural networks (CNNs), support vector machines (SVMs), recurrent neural networks (RNNs), deep belief networks, deep Boltzmann machines, and K-nearest neighbour.

The initial step is to receive an image form of handwritten characters, which is recognized as image acquisition that will proceed as an input to preprocessing. In preprocessing, distortions of the scanned images are removed and converted into binary images. Afterward, in the segmentation step, each character is divided into sub-images. Then, it will extract every characteristic of the features from each image of the character. This stage is especially important for the last step of the HCR system, which is called classification.

II. Problem Statement :

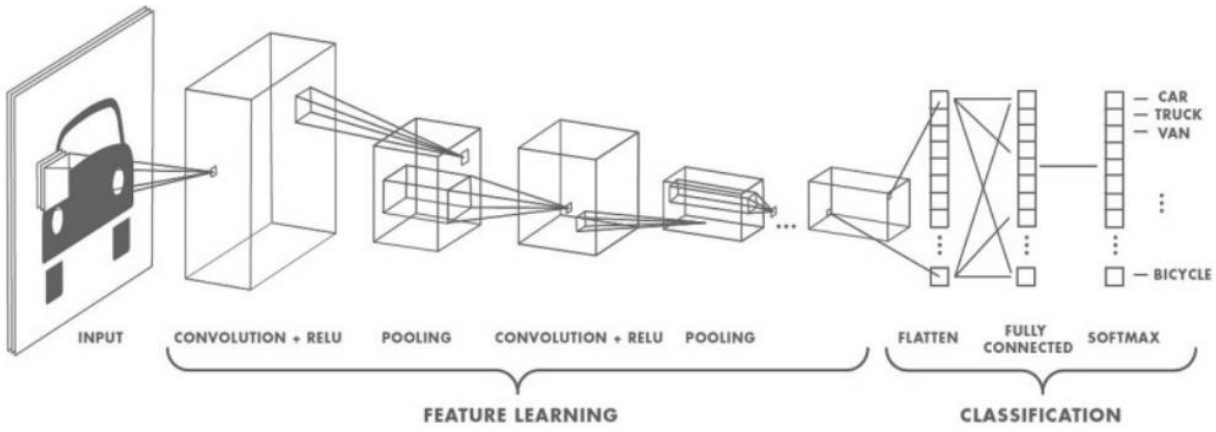
Developing a neural network-based system for handwritten text recognition involves designing a robust framework capable of accurately converting images of handwritten text into machine-readable text. The objective is to create a versatile system that can effectively handle diverse handwriting styles, languages, and varying levels of complexity.

Ideation for a Handwritten Character Recognition (HCR) system using Convolutional Neural Networks (CNNs) involves brainstorming and outlining the key components, objectives, and potential features of such a system. Here's an ideation process for an HCR system:

1. Objective Definition
2. Data Collection
3. Preprocessing
4. Model Architecture
5. Training Data
6. Data Labeling
7. Model Training
8. Validation and Testing
9. Fine-tuning
10. Real-time Recognition
11. Multilingual Support
12. Personalization
13. Feedback and Continuous Improvement
14. Scalability and Deployment

III. Abstract :

Neural networks have made big strides in image classification. Convolutional neural networks (CNN) work successfully to run neural networks on direct images. Handwritten character recognition (HCR) is now a very powerful tool to detect traffic signals, translate language, and extract information from documents, etc. Although handwritten character recognition technology is in use in the industry, present accuracy is not outstanding, which compromises both performance and usability. Thus, the character recognition technologies in use are still not very reliable and need further improvement to be extensively deployed for serious and reliable tasks. On this account, characters of the English alphabet and digit recognition are performed by proposing a custom-tailored CNN model with two different datasets of handwritten images, i.e., Kaggle and MNIST, respectively, which are lightweight but achieve higher accuracies than state-of-the-art models. The best two models from the total of twelve designed are proposed by altering hyper-parameters to observe which models provide the best accuracy for which dataset. In addition, the classification reports (CRs) of these two proposed models are extensively investigated considering the performance matrices, such as precision, recall, specificity, and F1 score, which are obtained from the developed confusion matrix (CM). To simulate a practical scenario, the dataset is kept unbalanced and three more averages for the F measurement (micro, macro, and weighted) are calculated, which facilitates better understanding of the performances of the models. The highest accuracy of 99.642% is achieved for digit recognition, with the model using 'RMSprop', at a learning rate of 0.001, whereas the highest detection accuracy for alphabet recognition is 99.563%, which is obtained with the proposed model using 'ADAM' optimizer at a learning rate of 0.00001. The macro F1 and weighted F1 scores for the best two models are 0.998, 0.997:0.992, and 0.996, respectively, for digit and alphabet recognition.



IV. Key Components and Steps :

a) Data Collection and Preprocessing:

Collect a diverse dataset of handwritten text samples that cover different styles, languages, and handwriting qualities. Preprocess the dataset by resizing images, normalizing pixel values, and augmenting data to increase its variability.

b) Feature Extraction:

For neural network-based handwriting recognition, the initial step involves extracting relevant features from the input images. CNNs are often used to automatically learn and extract these features.

c) Model Architecture:

Design a suitable neural network architecture. A common approach is to use a combination of CNNs and RNNs. The CNN layers can capture spatial features and patterns from the image.. The RNN layers, such as Long Short-Term Memory (LSTM) or Gated Recurrent Unit (GRU), can process sequences of features extracted by CNNs and learn the context of the handwriting.

d) Training:

Split the dataset into training, validation, and test sets. Train the neural network using the training data. Use appropriate loss functions such as CTC (Connectionist Temporal Classification) loss for sequence data. Experiment with various hyperparameters like learning rate, batch size, and network architecture to optimize performance.

e) Validation and Testing:

Evaluate the model's performance on the validation set to prevent overfitting and fine-tune hyperparameters. Test the final model on the test set to measure its real-world performance.

f) Post-Processing:

Apply post-processing techniques to improve the accuracy of recognized text, like language modelling , dictionary lookup, and spell-checking.

g) Deployment:

Once the model achieves satisfactory accuracy, deploy it as a user-friendly application or service . Users should be able to upload images of handwritten text, and the system should provide accurate text transcriptions. languages, using a standard API which is also known as the Arduino Programming Language, inspired by the Processing language and used with a modified version of the Processing IDE.

V. Challenges :

- Variability in Handwriting Styles:
  Handwriting can vary significantly from person to person, making it challenging to develop a modelthat handles all styles effectively.

- Noisy and Poor-Quality Images:
  Handwritten text can be unclear, distorted, or have smudges, making accurate recognition difficult.

- Multilingual Support:
  Developing a model that can recognize multiple languages adds complexity to the problem.

- Real-time Processing:
  Achieving real-time or near-real-time processing is crucial for user satisfaction.

- Scaling:
  As the user base grows, the system should be able to handle a large volume of requests efficiently.

VI. Phases :

1. Image source :

   This phase comes in offline hand-written character recognition. Image source can be from any digitized tool. A scanner or a camera captures the image and is sent to next phase.

2. Pre-processing :

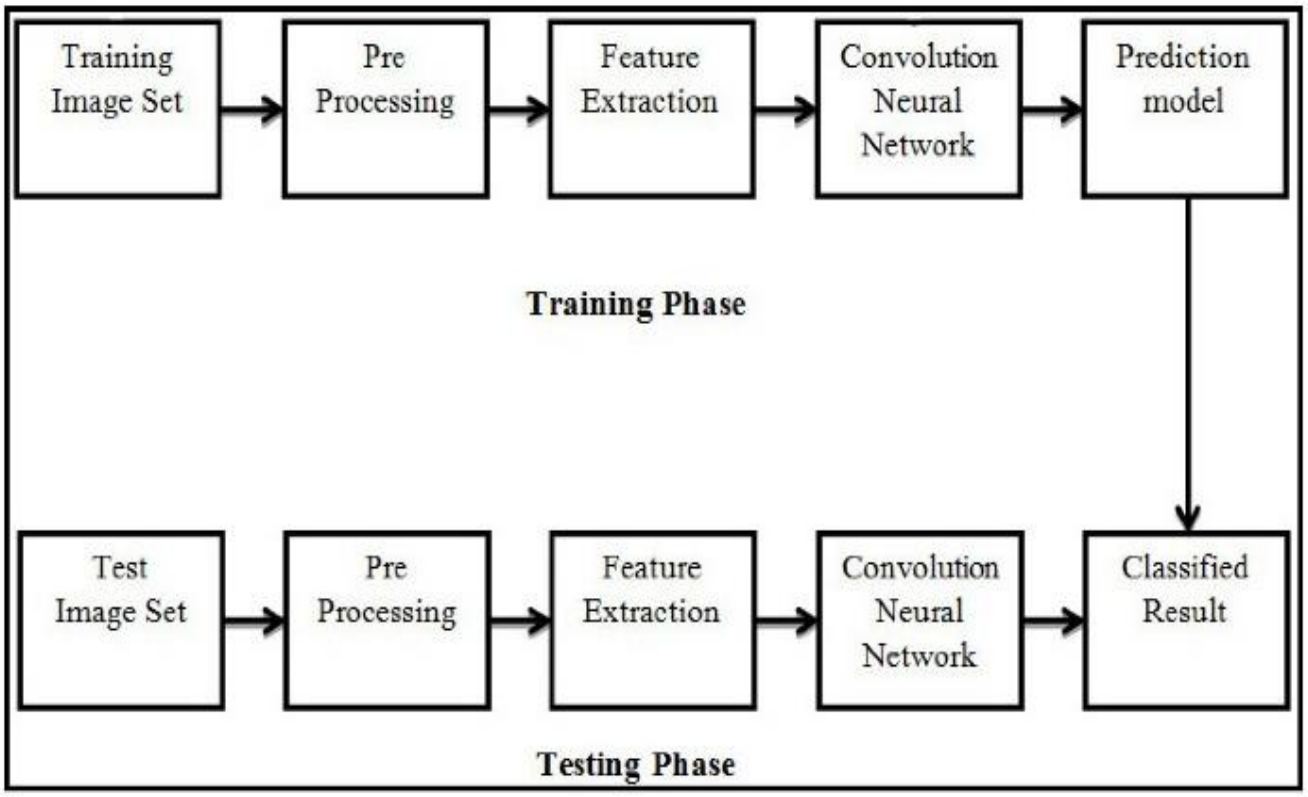
   Pre-processing is sequence of operation, that improves the quality of image and hence increases the accuracy of image. For the hand-written character recognition process, the following pre processing techniques are followed

   - Noise Removal
   - Binarization
   - Morphological operation

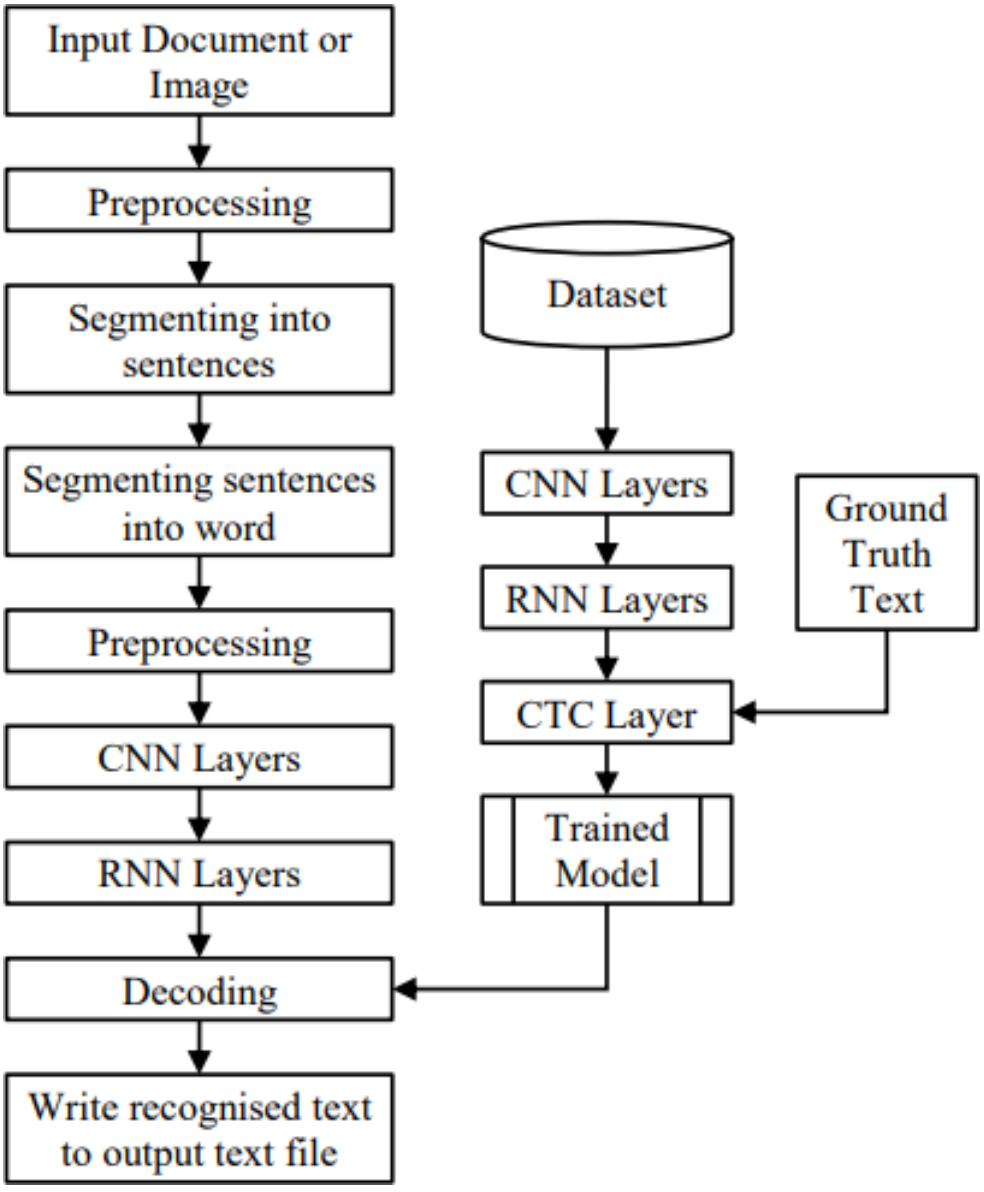
3. Segmentation :
   Segmentation is a mechanism that extracts individual characters in the image. There exists two types of segmentation. They are Implicit segmentation and Explicit segmentation. In implicit segmentation the words are recognized without segmentation process. But in explicit segmentation, words are predicted by extracting individual character.

4. Feature-extraction :
   This is the important phase in the recognition process, and the algorithm of recognition starts from here. Each character contains its own features. It contains group of rules where each rule explains feature of a character. Extraction of such features is done in this phase.

5. Classification :
   By this time, the training would have completed and the testing of input data starts. The testing data would pass all the above process and the varying probabilities are assigned to the matching rules. The rule with highest probability is selected and the corresponding class-label is made recognized character.



VII. Block Diagram :



VIII. Literature Survey :

Advent of deep learning encourages us to use convolutional neural network in machine learning problem. Convolutional neural network (CNN) is a multi layer perceptron model proposed by D.H.Hubel and T.N.Weisel that is inspired by biological neuron and animal vision. The basic design of CNN is inspired from the scientist Yann le Cunn who conducted certain experiments regarding mammal's perception and vision. Since the day of propose of CNN different researchers have created different version of the CNN and developed different type of applications like character recognition, object detection, face recognition and sound detection. Some of the works are presented here:

Bishwajit purkaystha, Tapos Datta,Md Saiful Islam Et al , this paper proposes a convolutional neural network model for recognizing handwritten Bengali characters. They proposed a convolutional neural network using two convolution layers and then using three densely connected layers, on the final dense layer Softmax Function is used. The first layer scans for 5*5 receptive field throughout the image. After scanning it is then passed through ReLu activation function. The ouput from this layer is then passed to max pooling layer having size 2*2. This output then again passed to second convolutional layer having 64 kernels of size 3*3. As in first layer same function ReLu and pooling of size 2*2 is applied. The output through this layer is actually used for classification process. They achieved 98.66% accuracy on numerals, accuracy of 94.99% on vowels, accuracy of 91.23% on alphabets and 89.93% accuracy on all Bengali characters. Some of the error in their prediction was due to misleading of the dataset.

Samad Roohi, Behnam Alizadehashrafi Et al, the study behind this paper is to investigate the performance of deep convolutional neural network on the Persian handwritten character dataset. To determine the outperformance of conventional method in PHCR problem, in this paper two type of CNN have been implemented. First network is single convolutional neural network which is based on the simple structure of CNN for example (LeNet-5). To extend it into ensemble CNN the bagging paradigm applied on CNN with a variety of network parameters. This paper concluded that the performance is 97.1% when ensemble CNN model was implemented and 96.3% accuracy when single CNN was implemented.

Mahesh jangid and sumit Srivastava Et al, this paper proposes the implementation of devanagari character using deep neural convolutional network (DCNN). They used the database provided by ISI (information sharing index) Kolkata, ISIDCHAR database and V2DMDCHAR database. They built six different architectures of DCNN to determine the performance of the network and also uses six different adaptive gradient method they use Adagrad optimizer, RMSProp optimizer, Adaptive Momentum estimation optimizer (Adam) etc. they concluded that AdaDelta, AdaMax and RMSProp optimizer outperformed the accuracy of Adam optimizer and got 97.30% accuracy on ISIDCHAR dataset by using DCNN. Similarly on V2DMDCHAR they achieved 97.65% accuracy by using layer wise DCNN and 96.45% accuracy by using DCNN and on the combination of both dataset they gained accuracy of 98.00% using layered wise DCNN and got 96.53% by using DCNN.

Jia Xiadong, Gong wedong, Yuan Jie , CNN is used by many researchers for character recognition task but they concluded two main problem which they face while using CNN.
(i)The first problem in CNN is the manual training of data is very time consuming and it is labor intensive.
(ii)Second problem is the design and parameter problem all depend on experiences.
(iii) To address these problem stated above they use density based clustering algorithm which is used as remedy for the first problem that means it is effective in data labeling. In this paper the handwritten Yi character recognition is divided into two parts. In the first part they describe how to construct the Yi characters database which uses improved density based clustering algorithm and in the second part they explain CNN architecture. In the construction of dataset they scan the Yi character documents then they binarize and normalize them and finally an image slice is used as the input in the clustering algorithm.
The CNN is formed by using four convolutional layers, four max pooling layers and one fully connected layer. The size of input image is 52*52 and the size of kernel or filter is 5*5 and 20 kernels are used therefore the feature map is 20 and the output size of feature map is 48*48 then max pooling layer was applied which reduces the output size into 20*20. In the second convolution layer they used 60 kernel of size 5*5 the feature obtained are 20*20 in size and 60 in number after pooling 0f size 2*2 the size of the output feature map is 10*10. In the third layer they use 120 kernel of size 3*3 after pooling the size 4*4. In the last layer kernel size is 2*2 they got 2*2 feature map and after that they applied logistic regression layer whose output is 100 because of 100 clusters in the database. Finally they achieved 99.65% accuracy on test set. They also concluded that if we pay more attention

towards the combination of CNN with different techniques then there are chances of higher accuracy.

Ashok kumar pant, Prashnna kumar Gyawali Shailesh Acharya they propose system for recognizing devnagari characters. They propose their own dataset of devnagari characters. In the dataset, 92 thousand images of 46 different classes of characters are segmented from handwritten documents. Along with the dataset they also propose the CNN architecture which was based on simple CNN. They built two models, model A and model B. In model A, the image in the dataset is rescaled from 36*36 to 28*28 which was convolve with 5*5 filter and a 2*2 size pooling layer was used which gives a output of 14*14 feature map. In the second convolutional layer this 14*14 feature map got convolved with 5*5 filter after that 2*2 max pooling layer was applied on it and the final output was of size 5*5 feature map after that Softmax Function was applied. The model B was derived from LeNet family. It has shallow architecture that means it consist of fewer number of parameters. They used mini batch size 200 and 50 epochs with learning rate 0.005 for model A and for model B it is 0.001 then stochastic gradient descent optimizer with momentum 0.9 is applied. The higher test accuracy obtained for model A is 0.98471 and for model B the higher test accuracy is 0.982681 addition of dropout show better result in model B.

Mathew Y.W. Teow presented a work on how to bridge the gap on understanding the mathematical structure and the computational implementation of a CNN using minimal model they proposed a minimal CNN which consist mainly two computational networks. They are feature learning network and the classification network. The objective of fully connected network is to perform unsupervised image feature learning. The learned image feature will be synthesized by feature learning with high level representation. Finally this high level image will be input to the classification to perform image recognition. They used MNIST dataset for handwritten digit recognition which consist of 70,000 different handwritten digits images where 60,000 digits are used for training and 10,000 images used for testing. Each digit image is of 28*28 in size and in grey scale. 20 trainable convolution kernels with kernel size 9*9 and used with input image, convolution perform feature extraction and generate 20 convolved feature map. This convolved feature rectified by ReLu activation function finally pooling layer perform mean pool to rectify feature map. Another proposed model is the extended version of the minimal model where they used 2 pooling layer; 2 convolution layer and 2 ReLu layer they called this model as extended minimal model. And then compare this with LeNet 5. The performance of the minimal

model is 97.3% and extended minimal is 98.50% with epoch 10.

Mujjadded AL Rabbani Alif, Sabbir Ahmed, Muhammad Abdul Hasan et al, in this paper they proposed a modified ResNet-18 architecture for Bangla handwritten characters. In order to measure the performance they used two recently introduced large dataset called Bangla lekha-Isolated dataset and CMARTER db dataset. The image size of this dataset vary from 110*110 to 220*220 pixels. Handwriting of this dataset is collected from 4 to 27 years age group. In the first experiment, to measure the performance three optimizers were used namely RMS prop, Adam and SGD on 110*110 inputs then using Adam they achieve 0.4% and 0.1% performance. In Second experiment they investigate the performance of proposed method with different dropout rates. Then investigate the performance of International Journal of Management, Technology And Engineering Volume 8, Issue VII, JULY/2018 ISSN NO : 2249-7455 Page No:262 bangla character recognition using several states of art CNN models. They use VGC Net-16, VGC Net19, ResNet-18, ResNet-34 and achieve 91.0%, 92.11%, 94.52%, 94.59%.

IX. References :

1. Priya, A.; Mishra, S.; Raj, S.; Mandal, S.; Datta, S. Online and offline character recognition: A survey. In Proceedings of the International Conference on Communication and Signal Processing, (ICCSP), Melmaruvathur, Tamilnadu, India, 6–8 April 2016; pp. 967–970.
2. Gunawan, T.S.; Noor, A.F.R.M.; Kartiwi, M. Development of english handwritten recognition using deep neural network. Indones. J. Electr. Eng. Comput. Sci. 2018, 10, 562–568. [CrossRef]
3. Vinh, T.Q.; Duy, L.H.; Nhan, N.T. Vietnamese handwritten character recognition using convolutional neural network. IAES Int. J. Artif. Intell. 2020, 9, 276–283. [CrossRef].
4. Dalal, N.; Triggs, B. Histograms of oriented gradients for human detection. In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), San Diego, CA, USA, 20–25 June 2005; Volume 1, pp. 886–893.
5. Lowe, D.G. Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vis. 2004, 60, 91–110. [CrossRef]
6. Xiao, J.; Zhu, X.; Huang, C.; Yang, X.; Wen, F.; Zhong, M. A New Approach for Stock Price Analysis and Prediction Based on SSA and SVM. Int. J. Inf. Technol. Decis. Mak. 2019, 18, 35–63. [CrossRef]
7. Wang, D.; Huang, L.; Tang, L. Dissipativity and synchronization of generalized BAM neural networks with multivariate discontinuous activations. IEEE Trans. Neural Netw. Learn. Syst. 2018, 29, 3815–3827. [CrossRef]
8. Kuang, F.; Zhang, S.; Jin, Z.; Xu, W. A novel SVM by combining kernel principal component analysis and improved chaotic particle swarm optimization for intrusion detection. Soft Comput. 2015, 19, 1187–1199. [CrossRef]
9. Choudhary, A.; Ahlawat, S.; Rishi, R. A binarization feature extraction approach to OCR: MLP vs. RBF. In Proceedings of the International Conference on Distributed Computing and Technology (ICDCIT), Bhubaneswar, India, 6–9 February 2014; Springer: Cham, Switzerland, 2014; pp. 341–346.
10. Fukushima, K. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. Biol. Cybern. 1980, 36, 193–202. [CrossRef]
11. Ahlawat, S.; Choudhary, A.; Nayyar, A.; Singh, S.; Yoon, B. Improved handwritten digit recognition using convolutional neural networks (Cnn). Sensors 2020, 20, 3344. [CrossRef]

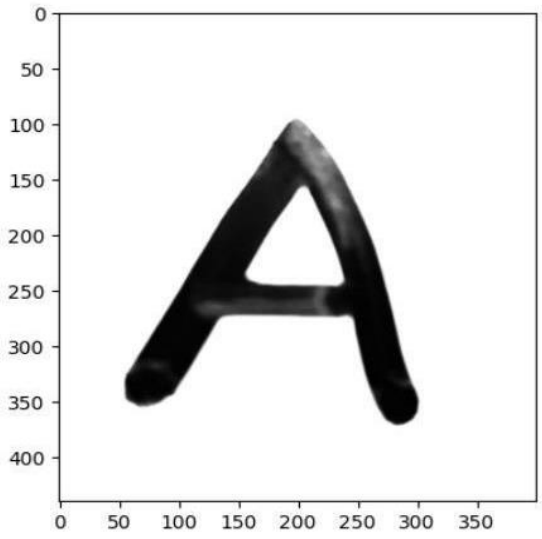
12. Jarrett, K.; Kavukcuoglu, K.; Ranzato, M.; LeCun, Y. What is the best multi-stage architecture for object recognition? In Proceedings of the IEEE 12th International Conference on Computer Vision (ICCV), Kyoto, Japan, 29 September–2 October 2009; pp. 2146–2153.
13. Cire¸san, D.C.; Meier, U.; Masci, J.; Gambardella, L.M.; Schmidhuber, J. High-Performance Neural Networks for Visual Object Classification. arXiv 2011, arXiv:1102.0183v1.
14. Ciresan, D.; Meier, U.; Schmidhuber, J. Multi-column deep neural networks for image classification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, 16–21 June 2012; IEEE: Piscataway, NJ, USA, 2012; pp. 3642–3649.
15. Niu, X.X.; Suen, C.Y. A novel hybrid CNN-SVM classifier for recognizing handwritten digits. Pattern Recognit. 2012, 45, 1318– 1325. [CrossRef]
16. Bishwajit Purkaystha, Tapos Datta, Md Saiful Islam, "Bengali Handwritten Character Recognition Using Deep Convolutional Neural Network", 2017 20th International Conference of Computer and Information technology(ICCIT), 22-24 December, 2017
17. Samad Roohi, Behnam Alizadehashrafi, "Persian Handwritten Character Recognition Using Convolutional Neural Network," 10th Iranian Confernce on Machine Vision and Image Processing, Nov, 22-23, 2017, Isfahan Univ. of Technology, Isfahan, Iran.
18. Mahesh Jangid and Sumit Srivastava, "Handwritten Devnagari Character Recognition Using Layer Wise Training of Deep Convolutional Neural Networks And Adaptive Gradient Methods", 2018 journal of imaging.
19. Jia xiaodong, gong wednog, yuan jie, "Handwritten Yi Character Recognition with Density Based Clustering Algorithm and Convolutional Neural Network", 2017 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference Embedded and Ubiquitous Computing (EUC).
20. Ashok Kumar Pant, Prashnna Kumar Gyawali, Shailesh Acharya, "Deep Learning Base Scale Handwritten Devanagri Character Recognition",
21. Matthew Y.W. Teow "Understanding Convolutional Neural Network Using A Minimal Model for handwritten Digit Recognition", 2017 IEEE 2nd International conferences on automatic and intelligent system, kota kinabalu, sabah, Malaysia.
22. Mujjadded AL Rabbani Alif, Sabbir Ahmed, Muhammad Abdul Hasan, "Isolated Bangla Handwritten Character Recognition with Convolutional Neural Network"
23. Karishma Verma, Manjeet Singh, "Hindi handwritten Character recognition using
24. Convolutional neural network", International journal of computer sciences and Engineering, vol.6,Issue.6, pp.909-914, 2018.

X. Conclusion :

In modern days, applications of handwritten character recognition (HRC) systems are flourishing. In this paper, to address HCR systems with multiclass classification, a CNN-based model is proposed that achieved exceptionally good results with this multiclass classification. The CNN models were trained with the MNIST digit dataset, which is shaped with 60,000 training and 10,000 testing images. They were also trained with the substantially larger Kaggle alphabet dataset, which comprises over 297,000 training images and a test set which is shaped on testing over 74,490 images. For the Kaggle dataset, the overall accuracies using the 'ADAM' optimizer were 99.516%, 99.511%, and 99.563% for learning rate (LR) 0.001, LR 0.0001, and LR 0.00001, respectively. Meanwhile, the same model using 'RMSprop' achieved accuracies of 99.292%, 99.108%, and 99.191%, respectively, by LR 0.001, LR 0.0001, and LR 0.00001. For the MNIST dataset, the overall accuracies using 'RMSprop' were 99.642%, 99.452%, and 98.142% for LR 0.001, LR 0.0001, and LR 0.00001, respectively. Meanwhile, the same model using the 'ADAM' optimizer achieved accuracies of 99.571%, 99.309%, and 98.142% with LR 0.001, LR 0.0001, and LR 0.00001, respectively. It can be easily understood that, for alphabet recognition, accuracy decreases with the increase in learning rate (LR); contrarily, overall accuracy is proportionately related to LR for digit recognition. In addition, precision, recall, specificity, measured from confusion matrices. Of all the discussed twelve models, the model using the 'ADAM' optimizer with LR 0.00001 obtained a recall value of 99.56%, and the model with LR 0.001 with the 'RMSprop' optimizer obtained the recall value of 99.64%; therefore, these two models excel other models for the Kaggle and MNIST datasets, respectively.

XI. Result and Output :

```
<matplotlib.image.AxesImage at 0x799b74a5b8e0>
```

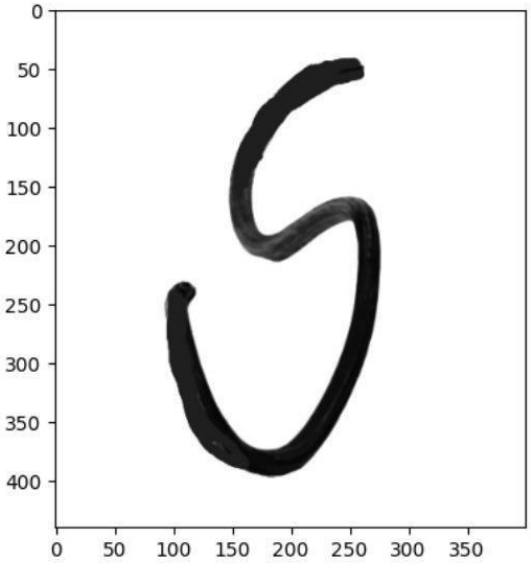



```
print(prediction)
```

```
A
```

```
plt.show
plt.imshow(image)
```

```
<matplotlib.image.AxesImage at 0x799b7499e890>
```




```
print(prediction)
```

```
S
```

CODE :

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

data = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/A_Z Handwritten Data.csv').astype('float32')

data.head(10)

X = data.drop('0',axis = 1)
y = data['0']

from sklearn.model_selection import train_test_split
from sklearn.utils import shuffle
x_train, x_test, y_train, y_test = train_test_split(X, y, test_size = 0.2)

x_train = np.reshape(x_train.values, (x_train.shape[0], 28,28))
x_test = np.reshape(x_test.values, (x_test.shape[0], 28,28))
print("Shape of Training data: ", x_train.shape)
print("Shape of Testing data: ", x_test.shape)

shuffle_data = shuffle(x_train)

import cv2

fig, axes = plt.subplots(3, 3, figsize=(10, 10))
axes = axes.flatten()

for i in range(9):
    _, shu = cv2.threshold(shuffle_data[i], 30, 200, cv2.THRESH_BINARY)
    axes[i].imshow(np.reshape(shu, (28, 28)), cmap="Greys")
```

```
plt.show()

x_train = x_train.reshape(x_train.shape[0],x_train.shape[1],x_train.shape[2],1)
x_test = x_test.reshape(x_test.shape[0], x_test.shape[1], x_test.shape[2],1)
print("New shape of training data: ", x_train.shape)
print("New shape of testing data: ", x_test.shape)

import tensorflow
from tensorflow.keras.utils import to_categorical

y_training = to_categorical(y_train, num_classes = 26, dtype='int')
y_testing = to_categorical(y_test, num_classes = 26, dtype='int')

from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Flatten, Conv2D, MaxPool2D, Dropout
from tensorflow.keras.optimizers import SGD, Adam
from tensorflow.keras.callbacks import ReduceLROnPlateau, EarlyStopping

model = Sequential()

model.add(Conv2D(64 , (3, 3), activation='relu', input_shape=(28,28,1)))
model.add(MaxPool2D(2, 2))

model.add(Conv2D(64, (3, 3), activation='relu'))
model.add(MaxPool2D(2, 2))

model.add(Conv2D(64, (3, 3), activation='relu'))
model.add(MaxPool2D(2,2))

model.add(Flatten())

model.add(Dense(128,activation ="relu"))

model.add(Dense(256,activation ="relu"))
```

```
model.add(Dense(128,activation ="relu"))
model.add(Dense(64,activation ="relu"))

model.add(Dense(32,activation ="relu"))

model.add(Dense(26,activation ="softmax"))


model.summary()

model.compile(optimizer = Adam(learning_rate=0.001), loss='categorical_crossentropy', metrics=['accuracy'])

history = model.fit(x_train, y_training, epochs=5,  validation_data = (x_test,y_testing))

model.save(r'handwritten_character_recog_model.h5')

words = {0:'A',1:'B',2:'C',3:'D',4:'E',5:'F',6:'G',7:'H',8:'I',9:'J',10:'K',11:'L',12:'M',13:'N',14:'O',15:'P',16:'Q',17:'R',18:'S',19:'T',20:'U',21:'V',22:'W',23:'X', 24:'Y',25:'Z'}

fig, axes = plt.subplots(3,3, figsize=(8,9))
axes = axes.flatten()

for i,ax in enumerate(axes):
    image = np.reshape(x_test[i], (28,28))
    ax.imshow(image, cmap="Greys")
    pred = words[np.argmax(y_testing[i])]
    ax.set_title("Prediction: "+pred)
    ax.grid()

image = cv2.imread('/content/A_Image.jpeg')
image_copy = image.copy()

image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
```

```
image = cv2.resize(image, (400,440))


image_copy = cv2.GaussianBlur(image_copy, (7,7), 0)
gray_image = cv2.cvtColor(image_copy, cv2.COLOR_BGR2GRAY)
_, img_thresh = cv2.threshold(gray_image, 100, 255, cv2.THRESH_BINARY_INV)

final_image = cv2.resize(img_thresh, (28,28))
final_image =np.reshape(final_image, (1,28,28,1))



prediction = words[np.argmax(model.predict(final_image))]
cv2.putText(image, "Prediction: " + prediction, (20,410),
cv2.FONT_HERSHEY_DUPLEX, 1.3, color = (0,255))

plt.show()

plt.imshow(image)
```

MODULE - 2 :

“Open AI Handwritten Notes Assistant”

I . Introduction :

In today's age, inundated with a surplus of information, the skill of swiftly retrieving specific data is crucial for enhanced productivity and informed decision-making. People from diverse backgrounds—students, researchers, professionals, and entrepreneurs—often confront the daunting task of managing extensive repositories of accumulated notes. Whether these notes stem from lectures, research discoveries, meeting summaries, or personal reflections, the primary hurdle lies in swiftly locating pertinent information when required.

Responding to this pervasive challenge, the OpenAI Notes Assistant project has emerged with the goal of crafting an advanced software tool that streamlines the note retrieval process. Our initiative acknowledges the significance of offering users an intuitive, effective, and personalized solution for proficient note management.

In contemporary society, where data deluge is the norm, the ability to swiftly pinpoint specific information is a critical skill that drives productivity and supports well-informed decision-making. Individuals across diverse fields—ranging from students and researchers to professionals and entrepreneurs—regularly encounter the formidable task of managing extensive archives of accumulated notes. These notes may encompass a wide spectrum of content, from lecture summaries and research insights to meeting synopses and personal musings. The central challenge lies in promptly accessing relevant information at the moment it is needed.

In response to this prevalent challenge, the OpenAI Notes Assistant project has been conceived to develop a sophisticated software tool that simplifies the note retrieval process. Our project recognizes the importance of providing users with an intuitive, efficient, and personalized solution for managing their notes effectively.

Living in an era dominated by an overflow of information, the ability to efficiently retrieve specific pieces of data is essential for productivity and effective decision-making. Individuals from various walks of life—be they students, researchers, professionals, or entrepreneurs—regularly grapple with the task of managing extensive collections of notes accumulated over time. These notes may encompass a variety of sources, such as lecture notes, research findings, meeting minutes, or personal reflections. The primary challenge lies in swiftly accessing the precise information required at any given moment.

The OpenAI Notes Assistant project has emerged as a response to this pressing challenge,

aiming to create a sophisticated software tool that simplifies the process of note retrieval. Our project acknowledges the critical need to provide users with an intuitive, efficient, and personalized solution for managing their notes effectively.

In an age characterized by an overwhelming amount of information, the ability to quickly find specific pieces of data is essential for productivity and effective decision-making. Individuals across different fields—from students and researchers to professionals and entrepreneurs—often struggle with managing extensive collections of notes accumulated over time. Whether it's lecture notes, research findings, meeting minutes, or personal reflections, the challenge lies in efficiently accessing the right information when needed.

The OpenAI Notes Assistant project is a response to this challenge, aiming to develop a sophisticated software tool that simplifies the process of retrieving notes. Our project recognizes the importance of providing users with an intuitive, efficient, and personalized solution for managing their notes effectively.

Living in an era of information overload, the ability to efficiently retrieve specific pieces of information is crucial for productivity and decision-making. People across various sectors—students, researchers, professionals, and entrepreneurs—often deal with vast collections of notes accumulated over time. Whether it's lecture notes, research findings, meeting minutes, or personal thoughts, the challenge lies in quickly accessing the relevant information when needed.

The OpenAI Notes Assistant project aims to address this challenge by developing a sophisticated software tool that simplifies note retrieval. Our project acknowledges the importance of offering users an intuitive, efficient, and personalized solution for managing their notes effectively.

In today's world, where information overload is the norm, the skill of efficiently retrieving specific information is vital for productivity and effective decision-making. Individuals across different domains—students, researchers, professionals, and entrepreneurs—frequently contend with large volumes of accumulated notes. Whether these notes are from lectures, research projects, meetings, or personal reflections, the primary obstacle is swiftly accessing the right information at the right time.

Responding to this challenge, the OpenAI Notes Assistant project aims to develop an advanced software tool that streamlines the note retrieval process. Our project recognizes the importance of providing users with an intuitive, efficient, and personalized solution for managing their notes effectively.

II. Problem Statement :

In today's information-rich environment, individuals face significant challenges in efficiently managing and retrieving specific information from their accumulated notes. Students, researchers, professionals, and entrepreneurs encounter difficulties accessing relevant data stored across various formats such as lecture notes, research findings, meeting minutes, and personal reflections. The complexity lies in swiftly accessing pertinent information when required, impacting productivity and decision-making.

The OpenAI Notes Assistant project aims to address this critical challenge by developing a sophisticated software tool that streamlines note retrieval. The goal is to provide users with an intuitive, efficient, and personalized solution for managing their notes effectively. This project recognizes the pressing need for a robust system to navigate and utilize vast collections of notes, empowering users to enhance their productivity and decision-making abilities in information-intensive environments.

Key challenges to be addressed:

- Information Overload: Users are overwhelmed by extensive collections of notes spanning different subjects and contexts.
- Time-Consuming Retrieval: Locating specific information quickly from accumulated notes is inefficient and time-consuming.
- Productivity Impact: Difficulty in accessing relevant information hampers productivity and effective decision-making.
- Need for Personalization: Users require tailored solutions to manage notes according to individual preferences and workflows.

Objectives of the OpenAI Notes Assistant project:

- Develop a sophisticated software tool to simplify and accelerate the process of retrieving specific pieces of information from extensive note collections.

- Design an intuitive user interface that enables seamless navigation and efficient management of notes across various domains.

- Implement personalized features that cater to individual user needs and preferences for enhanced usability.

- Enhance productivity and support informed decision-making by facilitating quick and accurate access to pertinent information stored in notes.

By addressing these challenges and objectives, the OpenAI Notes Assistant project aims to revolutionize note management, offering users a powerful and user-friendly solution to navigate the complexities of information overload in today's digital age.

III. Abstract :

The "OpenAI Notes Assistant" represents a sophisticated system engineered to optimize productivity and organization across personal and professional contexts. This innovative tool harnesses cutting-edge artificial intelligence and natural language processing technologies to streamline note retrieval and management, revolutionizing the traditional methods of manual searching and categorization.

At its core, the OpenAI Notes Assistant features a powerful search engine adept at interpreting intricate queries and swiftly retrieving pertinent notes. Users benefit from the convenience of inputting queries in natural language, fostering seamless interaction and alleviating cognitive burden. This intuitive interface empowers users to effortlessly classify notes based on topics, projects, or deadlines, thereby facilitating rapid access and easy reference to stored information.

The assistant's search capabilities transcend basic keyword matching, employing advanced algorithms to understand context and infer user intent. This intelligent approach enables users to pose complex questions or request specific information, confident that the assistant will deliver precise and relevant results promptly. By leveraging artificial intelligence, the OpenAI Notes Assistant adapts to user preferences and usage patterns over time, refining its recommendations and enhancing user satisfaction.

Moreover, the system's organizational features allow users to efficiently manage their notes, ensuring structured repositories that facilitate efficient retrieval. Users can categorize notes according to customized tags or labels, enabling personalized organization tailored to individual workflows. Additionally, the assistant supports collaboration by providing seamless sharing and access controls, enhancing teamwork and knowledge sharing within professional environments.

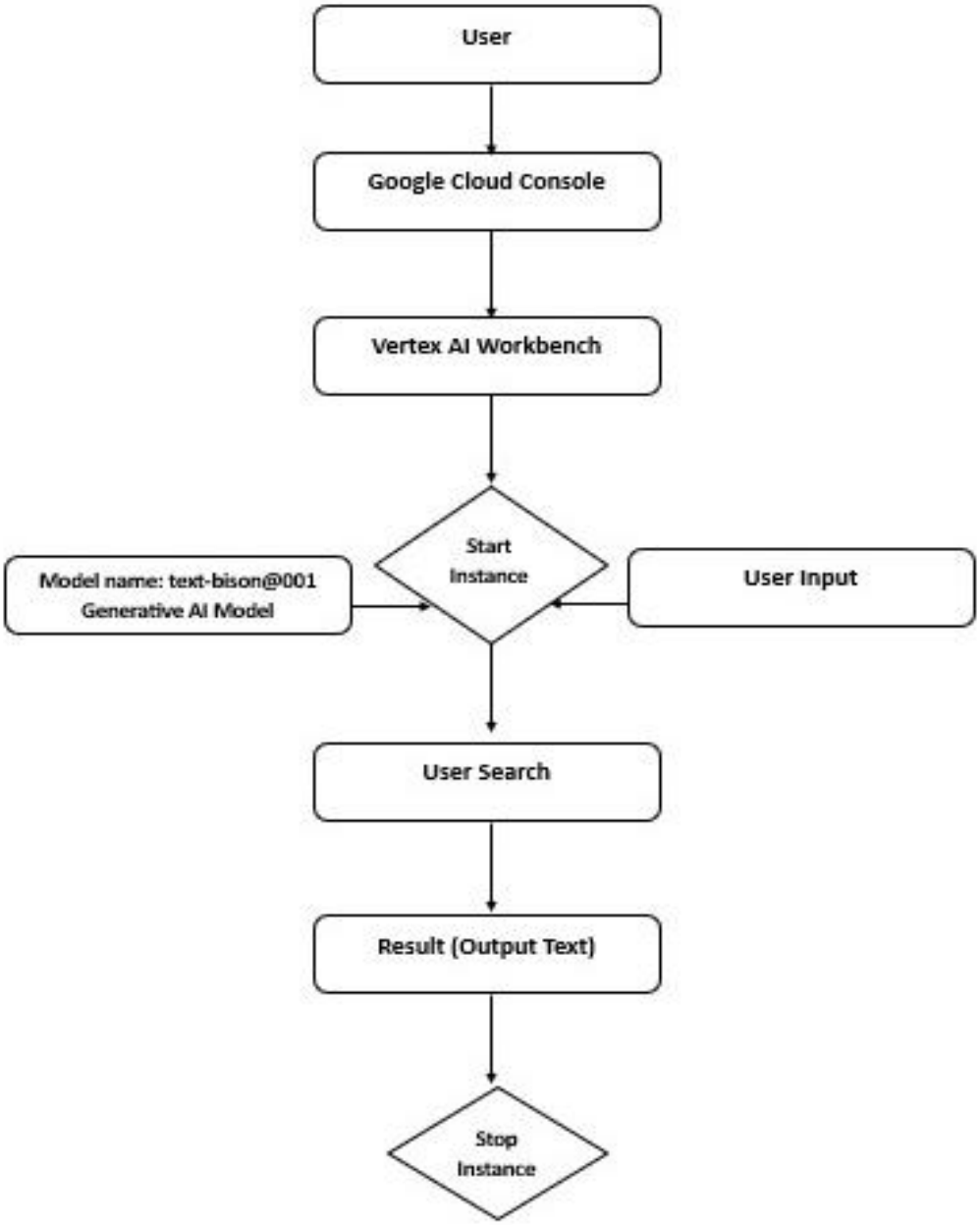
The OpenAI Notes Assistant represents a significant advancement in information management technology, offering a holistic solution for individuals seeking to optimize productivity and streamline their workflow. Its integration of AI-driven capabilities, intuitive interface, and robust organizational tools make it a valuable asset in both personal

and professional settings. As information continues to proliferate, tools like the OpenAI Notes Assistant play a pivotal role in empowering users to navigate the complexities of data management with efficiency and precision.

Index Terms

Artificial Intelligence (AI) ,Natural Language Processing (NLP) ,Information Retrieval , Note Management ,Productivity Enhancement ,Search Engine ,Query Interpretation, Contextual Understanding ,User Interaction ,Cognitive Load Reduction ,Query Language Intelligent Note Classification ,Topic-based Note Organization ,Project Management, Deadline Tracking ,Advanced Algorithms ,User Intent Inference ,Workflow Optimization, Structured Repositories ,Efficient Retrieval ,Information Overload Management

IV . Flow Diagram :



V . Challenges :

1. Information Overload:
   In today's information-rich world, individuals accumulate vast amounts of notes across different formats (e.g., lecture notes, research findings, meeting minutes). Managing this volume of information becomes challenging, leading to inefficiencies in accessing specific and relevant data.

2. Complexity of Note Retrieval:
   Retrieving pertinent information from extensive collections of notes can be complex and time-consuming. Users struggle to quickly locate specific content when needed, impacting their productivity and decision-making abilities.

3. Diverse User Needs:
   Students, researchers, professionals, and entrepreneurs have varied requirements for note management. The project must cater to a diverse user base with different preferences and workflows, necessitating a flexible and adaptable software solution.

4. Intuitive Interface:
   Designing an interface that is intuitive and user-friendly is crucial for user adoption and satisfaction. The software tool needs to simplify the note management process without adding cognitive burden to the users.

5. Personalization:
   Delivering a personalized experience that aligns with individual user preferences and habits is a challenge. The project aims to leverage advanced technologies to tailor note organization and retrieval based on user behavior.

6. Efficiency and Speed:
   The solution must be efficient in retrieving notes promptly. Users require a system that can handle complex queries and deliver accurate results in real-time, enhancing their efficiency and saving time.

7. Robust Search Capabilities:
   Developing a robust search engine capable of understanding complex queries and context is essential. The software needs to interpret natural language queries and retrieve relevant information accurately.

8. Data Security and Privacy:
   Ensuring the security and privacy of users' notes and data is paramount. The project must implement robust measures to protect sensitive information stored within the system.

9. Integration and Compatibility:
   The software tool should seamlessly integrate with existing platforms and applications commonly used by individuals across various domains. Compatibility with different devices and operating systems is essential for widespread adoption.

10. Continuous Improvement:
    The project needs to adopt an iterative approach to development, incorporating user feedback and evolving technological advancements. Continuous improvement is crucial for enhancing the effectiveness and relevance of the OpenAI Notes Assistant over time.

VI . Model :

Model_name="text-bison@001"

The PaLM 2 for Text (text-bison, text-unicorn) foundation models represent a cutting-edge advancement optimized specifically for a range of natural language processing tasks, including sentiment analysis, entity extraction, and content generation. These models excel in generating various types of textual content, such as document summaries, answers to questions, and labels for content classification. They are designed to provide comprehensive solutions for tasks that can be executed with a single API response, streamlining processes that do not necessitate ongoing dialogue or interaction.

The versatility of the PaLM 2 for Text models makes them particularly suitable for scenarios where immediate and concise outputs are required. For instance, they excel in providing quick insights through document summarization, extracting relevant entities from text, or generating responses to specific questions. This efficiency enhances productivity by reducing the time and effort needed to derive actionable information from textual data.

Furthermore, the PaLM 2 for Text models are adept at generating labels that aid in organizing and categorizing content based on its nature or topic. This capability is valuable for automated content management systems that require efficient tagging and classification of large volumes of text data.

It's important to note that the PaLM 2 for Text models are specifically tailored for tasks that do not involve ongoing conversational interactions. For scenarios requiring dynamic dialogue or continuous exchanges, such as chatbots or conversational interfaces, the recommendation is to leverage the Generative AI on Vertex AI API. This API is optimized for handling text tasks that necessitate back-and-forth interactions and nuanced responses in a conversational context.

In summary, the PaLM 2 for Text foundation models offer a powerful toolkit for leveraging natural language processing in various applications. From sentiment analysis

to content creation and entity extraction, these models provide efficient solutions for generating insightful outputs from textual data, particularly suited for tasks that can be addressed with a single API call. For conversational scenarios demanding continuous interaction and dynamic responses, alternative APIs like Generative AI on Vertex AI are recommended to ensure optimal performance and user experience.

USE CASE of MODEL :

- Sentiment Analysis:

  The text_bison model can analyze the sentiment of text, determining whether the expressed sentiment is positive, negative, or neutral. This use case is valuable for understanding customer feedback, social media sentiment, or sentiment trends in product reviews.

- Entity Extraction:

  By leveraging named entity recognition capabilities, the text_bison model can identify and extract entities such as people, organizations, locations, dates, and more from textual data. This use case is useful for information retrieval, content categorization, and knowledge extraction.

- Document Summarization:

  The text_bison model can generate concise summaries of lengthy documents, distilling key information and main points. This use case is beneficial for quickly understanding the content of large volumes of text, such as research papers, news articles, or legal documents.

- Question Answering:

  Using its ability to comprehend and generate text, the text_bison model can answer specific questions based on provided context or knowledge sources. This use case is applicable in chatbot applications, virtual assistants, and information retrieval systems.

- Content Classification:

  Leveraging its classification capabilities, the text_bison model can assign labels or categories to textual content based on its topic, theme, or relevance. This use case supports content organization, information retrieval, and automated tagging.

- Text Generation:

  The text_bison model can generate coherent and contextually relevant text based on input prompts. This use case is valuable for content creation tasks such as generating product descriptions, marketing copy, or creative writing.

- Language Translation:

  Through its language understanding and generation capabilities, the text_bison model can facilitate language translation tasks by converting text from one language to another. This use case supports multilingual communication and global content localization.

- Content Moderation:

  The text_bison model can assist in content moderation by identifying inappropriate or offensive language within text. This use case is important for maintaining community guidelines and ensuring a safe online environment.

WORKING MECHANISM

The detailed working process of the text_bison model within the OpenAI Notes Assistant project involves several steps, leveraging advanced natural language processing (NLP) techniques to analyze and process textual data effectively. Here's a breakdown of the working process of text_bison:

1. Input Processing:

- Text Input: The text_bison model accepts textual input, which can include various forms such as sentences, paragraphs, or documents.
- Preprocessing: The input text undergoes preprocessing steps, which may include tokenization (splitting text into individual words or tokens), normalization (converting text to a standard format), and possibly removing stopwords (commonly occurring words with little semantic value).

2. Feature Extraction:

- Keyword Extraction: The text_bison model extracts key features from the input text. This can involve identifying important keywords or phrases that capture the essence of the content.
- Entity Recognition: Using named entity recognition (NER), the model identifies entities such as persons, organizations, locations, dates, or other relevant information mentioned in the text.
- Context Understanding: The model analyzes the context of the input text to comprehend the relationships between words, phrases, and entities.

3. Task-Specific Processing:

- Sentiment Analysis: For sentiment analysis tasks, the text_bison model may determine the sentiment expressed in the text (positive, negative, neutral) using classification techniques.
- Document Summarization: When tasked with document summarization, the model generates a concise summary by selecting important sentences or phrases that capture the main points of the document.
- Question Answering: In question-answering scenarios, the model interprets questions and generates relevant answers based on the context provided in the input text.

4. Neural Network Processing:

- Transformer Architecture: The text_bison model likely employs a transformer-based architecture, such as BERT (Bidirectional Encoder Representations from Transformers) or similar models. These architectures leverage self-attention mechanisms to capture relationships between words and encode contextual information efficiently.
- Fine-Tuning: The model may have been fine-tuned on specific tasks or datasets to optimize performance for tasks like sentiment analysis, entity extraction, or summarization.

5. Output Generation:

- Result Interpretation: Based on the input and task, the text_bison model generates outputs such as extracted keywords, identified entities, sentiment scores, summarized text, or answers to questions.
- Formatted Output: The model organizes and formats the outputs to be presented in a user-friendly manner, depending on the specific application or interface of the Notes Assistant.

6. Post-Processing and Presentation:

- Post-Processing: The generated outputs may undergo additional post-processing steps, such as filtering redundant information or enhancing readability.
- Presentation: The final results from the text_bison model are presented to users through the Notes Assistant interface, enabling efficient note management, content analysis, or other functionalities.

7. Continuous Learning and Improvement:

- Feedback Loop: The text_bison model may incorporate a feedback loop to continuously improve its performance based on user interactions and real-world usage.
- Model Updates: Periodic updates and enhancements to the model ensure it remains effective and adaptable to evolving language patterns and user needs.

VII. Algorithm :

The concept of a "MapReduce chain" in the context of the OpenAI Notes Assistant project refers to a specific approach or methodology used to process and analyze large volumes of data efficiently. This concept draws inspiration from the MapReduce framework, a programming model commonly used for processing and generating insights from big data sets. Let's break down the components and steps involved in a MapReduce chain within the context of the Notes Assistant project:

1. Mapping Phase:

- Input: The mapping phase starts with a large dataset, which in the context of the Notes Assistant project, could represent a collection of notes stored in various formats (text, images, audio).

- Mapping Function: A mapping function is applied to each data element (or note) to extract relevant information or features.
  For example:
  - Extracting keywords or key phrases from text notes.
  - Analyzing image notes to identify objects or themes using computer vision techniques.
  - Processing audio notes to transcribe spoken content into text.

- Intermediate Key-Value Pairs: The mapping phase generates intermediate key-value pairs, where the key represents a specific attribute or characteristic (e.g., keyword, object label) and the value corresponds to the data associated with that attribute (e.g., note content).

2. Shuffle and Sort:

- Data Redistribution: The intermediate key-value pairs generated in the mapping phase are shuffled and distributed across the processing nodes based on the keys.

- Sorting: Within each node, the key-value pairs are sorted based on the keys. This step prepares the data for the subsequent reduction phase.

3. Reducing Phase:

- Input: The sorted key-value pairs from the shuffle and sort phase serve as input to the reducing phase.

- Reducing Function: A reducing function is applied to each unique key to aggregate or summarize the associated values.
  For example:
  - Combining all text notes containing a specific keyword into a single document or summary.
  - Analyzing and summarizing image or audio notes associated with specific identified objects or themes.

- Output: The reducing phase produces the final output, which could be consolidated summaries, categorized data, or aggregated insights derived from the processed notes.

Application in Notes Assistant Project:
In the context of the OpenAI Notes Assistant project, a MapReduce chain could be implemented for various tasks such as:

- Note Organization: Mapping phase to extract features (keywords, topics) from notes, followed by a reducing phase to categorize and organize notes based on extracted features.
- Content Analysis: Mapping phase to analyze content attributes (sentiment, entities), followed by a reducing phase to generate insights (summary, sentiment analysis) from notes.
- Search and Retrieval: Mapping phase to index notes with relevant metadata, followed by a reducing phase to retrieve and present search results based on user queries.

Benefits and Use Cases:

- Scalability: MapReduce chains enable efficient processing of large datasets, making it suitable for managing extensive collections of notes in the Notes Assistant project.

- Parallel Processing: The framework allows for parallel execution of mapping and reducing tasks across distributed computing resources, improving performance and throughput.
- Flexibility: MapReduce chains can be customized with different mapping and reducing functions to accommodate diverse use cases, enhancing the versatility of the Notes Assistant project in handling various note management and analysis tasks.

In summary, implementing a MapReduce chain within the OpenAI Notes Assistant project involves leveraging the principles of mapping, shuffling, sorting, and reducing to efficiently process and derive insights from large volumes of notes. This approach contributes to scalable, parallelizable, and customizable data processing workflows that enhance the functionality and performance of the Notes Assistant in managing and analyzing textual content effectively.

VIII . Literature Survey :

Ardito, C.G. (2023). Contra generative AI detection in higher education assessments. arXiv preprint arXiv:2312.05241

This paper presents a critical analysis of generative Artificial Intelligence (AI) detection tools in higher education assessments. The rapid advancement and widespread adoption of generative AI, particularly in education, necessitates a reevaluation of traditional academic integrity mechanisms. We explore the effectiveness, vulnerabilities, and ethical implications of AI detection tools in the context of preserving academic integrity. Our study synthesises insights from various case studies, newspaper articles, and student testimonies to scrutinise the practical and philosophical challenges associated with AI detection. We argue that the reliance on detection mechanisms is misaligned with the educational landscape, where AI plays an increasingly widespread role. This paper advocates for a strategic shift towards robust assessment methods and educational policies that embrace generative AI usage while ensuring academic integrity and authenticity in assessments.

Bozkurt, A. (2023). Unleashing the potential of generative AI, conversational agents and chatbots in educational praxis: A systematic review and bibliometric analysis of GenAI in education. Open Praxis, 15(4).

In the rapidly evolving landscape of education, the pivotal axis around which transformation revolves is human-AI interaction. In this sense, this paper adopts a data mining and analytic approach to understand what the related literature tells us regarding the trends and patterns of generative AI research in educational praxis. Accordingly, this systematic exploration spotlights the following research themes: Interaction and communication with generative AI-powered chatbots; impact of the LLMs and generative AI on teaching and learning, conversational educational agents and their opportunities, challenges, and implications; leveraging Generative AI for enhancing social and cognitive learning processes; promoting AI literacy for unleashing future opportunities; harnessing Generative AI to expand academic capabilities, and lastly, augmenting educational experiences through human-AI interaction. Beyond the identified research themes and patterns, this paper argues that emotional intelligence, AI

literacy, and prompt engineering are the trending research topics that require further exploration. Accordingly, it's in this praxis that emotional intelligence emerges as a pivotal attribute, as AI technologies often struggle to comprehend and respond to the nuanced emotional cues. Generative AI literacy then takes center stage, becoming an indispensable asset in an era permeated with AI technologies, equipping students with the tools to critically engage with AI systems, thereby ensuring they become active, discerning users of these powerful tools. Concurrently, prompt engineering, the art of crafting queries that yield precise and valuable responses from AI systems, empowers both educators and students to maximize the utility of AI-driven educational resources.

Bull, C., & Kharrufa, A. (2023). Generative AI Assistants in Software Development Education: A vision for integrating Generative AI into educational practice, not instinctively defending against it. IEEE Software, 1–9.

The software development industry is amid another disruptive paradigm change—adopting the use of generative AI (GAI) assistants for programming. Whilst AI is already used in various areas of software engineering [1], GAI technologies, such as GitHub Copilot and ChatGPT, have ignited peoples' imaginations (and fears [2]). It is unclear how the industry will adapt, but the move to integrate these technologies by large software companies, such as Microsoft (GitHub1 , Bing2) and Google (Bard3), is a clear indication of intent and direction. We performed exploratory interviews with industry professionals to understand current practice and challenges, which we incorporate into our vision of a future of software development education and make some pedagogical recommendations.

Chan, C. K. Y., & Lee, K. K. (2023). The AI generation gap: Are Gen Z students more interested in adopting generative AI such as ChatGPT in teaching and learning than their Gen X and Millennial Generation teachers?. arXiv preprint arXiv:2305.02878

This study aimed to explore the experiences, perceptions, knowledge, concerns, and intentions of Gen Z students with Gen X and Gen Y teachers regarding the use of generative AI (GenAI) in higher education. A sample of students and teachers were recruited to investigate the above using a survey consisting of both open and closed questions. The findings showed that Gen Z participants were generally optimistic about the potential benefits of GenAI, including enhanced productivity, efficiency, and personalized learning, and expressed intentions to use GenAI for various educational

purposes. Gen X and Gen Y teachers acknowledged the potential benefits of GenAI but expressed heightened concerns about overreliance, ethical and pedagogical implications, emphasizing the need for proper guidelines and policies to ensure responsible use of the technology. The study highlighted the importance of combining technology with traditional teaching methods to provide a more effective learning experience. Implications of the findings include the need to develop evidence-based guidelines and policies for GenAI integration, foster critical thinking and digital literacy skills among students, and promote responsible use of GenAI technologies in higher education.

Faycal, F. et al. (2023). Analyzing The Students' Views, Concerns, and Perceived Ethics about Chat GPT Usage. Computers and Education: Artificial Intelligence, 5. Artificial Intelligence has greatly revolutionized education in many aspects. Today, AI-enabled language models, such as ChatGPT, are gaining popularity due to their characteristics and benefits. However, users also consider them a threat to educational integrity and purposes. This research examined ChatGPT usage among students in the United Arab Emirates (UAE), their views, concerns, and perceived ethics. The data was gathered from 388 students from two universities in Al Ain city using Yamane's formula. Findings showed that students consider ChatGPT a revolutionary technology that helps students in many ways. The gathered data showed that the effect of ChatGPT Usage remained significant on students' views. The path analysis also supported the second hypothesis, proposing the significant effect of ChatGPT on Students' Concerns. Finally, the findings also indicated the validation of the final hypothesis, showing the significant effect of ChatGPT Usage on the Perceived Ethics among the students in the UAE. Therefore, this study concluded that using ChatGPT in education has useful and concerning effects on educational integrity. However, implementing practical guidelines can assist in making informed decisions and shaping policies within educational institutions. Recognizing the complexities and importance of ChatGPT usage, teachers and policymakers can keep a balance by leveraging Artificial Intelligence technology to improve education while upholding ethical practices that promote critical thinking, originality, and integrity among students.

Gimpel, H., Et al. (2023). Unlocking the power of generative AI models and systems such as GPT-4 and ChatGPT for higher education: A guide for students and lecturers (No. 02-2023). Hohenheim Discussion Papers in Business, Economics and Social Sciences.

Generative AI technologies, such as large language models, have the potential to revolutionize much of our higher education teaching and learning. ChatGPT is an impressive, easy-to-use, publicly accessible system demonstrating the power of large language models such as GPT-4. Other compa- rable generative models are available for text processing, images, audio, video, and other outputs - and we expect a massive further performance increase, integration in larger software systems, and diffusion in the coming years. This technological development triggers substantial uncertainty and change in university-level teaching and learning. Students ask questions like: How can ChatGPT or other artificial intelligence tools support me? Am I allowed to use ChatGPT for a seminar or final paper, or is that cheating? How exactly do I use ChatGPT best? Are there other ways to access models such as GPT-4? Given that such tools are here to stay, what skills should I acquire, and what is obsolete? Lecturers ask similar questions from a different perspective: What skills should I teach? How can I test students' competencies rather than their ability to prompt generative AI models? How can I use ChatGPT and other systems based on generative AI to increase my efficiency or even improve my students' learning experience and outcomes? Even if the current discussion revolves around ChatGPT and GPT-4, these are only the forerunners of what we can expect from future generative AI-based models and tools. So even if you think ChatGPT is not yet technically mature, it is worth looking into its impact on higher education. This is where this whitepaper comes in. It looks at ChatGPT as a contemporary example of a conversational user interface that leverages large language models. The whitepaper looks at ChatGPT from the perspective of students and lecturers. It focuses on everyday areas of higher education: teaching courses, learning for an exam, crafting seminar papers and theses, and assessing students' learning outcomes and performance. For this purpose, we consider the chances and concrete application possibilities, the limits and risks of ChatGPT, and the underlying large language models.

Sullivan, M., Kelly, A., & McLaughlan, P. (2023). ChatGPT in higher education: Considerations for academic integrity and student learning. Journal of Applied Learning & Teaching, 6(1), 1-10.

The release of ChatGPT has sparked significant academic integrity concerns in higher education. However, some commentators have pointed out that generative artificial intelligence (AI) tools such as ChatGPT can enhance student learning, and consequently,

academics should adapt their teaching and assessment practices to embrace the new reality of living, working, and studying in a world where AI is freely available. Despite this important debate, there has been very little academic literature published on ChatGPT and other generative AI tools. This article uses content analysis to examine news articles (N=100) about how ChatGPT is disrupting higher education, concentrating specifically on Australia, New Zealand, the United States, and the United Kingdom. It explores several key themes, including university responses, academic integrity concerns, the limitations and weaknesses of AI tool outputs, and opportunities for student learning. The data reveals mixed public discussion and university responses, with a focus mainly on academic integrity concerns and opportunities for innovative assessment design. There has also been a lack of public discussion about the potential for ChatGPT to enhance participation and success for students from disadvantaged backgrounds. Similarly, the student voice is poorly represented in media articles to date. This article considers these trends and the impact of AI tools on student learning at university.

IX .References :

[1] Lund, B. D., & Wang, T. (2023). Chatting about ChatGPT: how may AI and GPT impact academia and libraries? Library Hi Tech News. https://doi.org/10.1108/lhtn-01-2023-0009

[2] Wei, J.; Bosma, M.; Zhao, V.Y.; Guu, K.; Yu, A.W.; Lester, B.; Du, N.; Dai, A.M.; Le, Q.V. Finetuned language models are zero-shot learners. arXiv 2022, arXiv:2109.01652

[3] Zhang, Y.; Sun, S.; Galley, M.; Chen, Y.-C.; Brockett, C.; Gao, X.; Gao, J.; Liu, J.; Dolan, B. Dialogpt: Large-scale generative pre-training for conversational response generation. arXiv 2022, arXiv:1911.00536.

[4] Seminck, O. Conversational AI: Dialogue systems, conversational agents, and Chatbots by Michael McTear. Comput. Linguist. 2023, 49, 257–259.

[5] Brownlee, J. How to Develop a GPT-2 Text Generator in Python. Machine Learning Mastery. 2021.Availableonline: https://machinelearningmastery.com/how-to-develop-a-generative-model-for-text-generation-in-python/

[6] OpenAI. Embeddings. Available online: https://platform.openai.com/docs/guides/embeddings

[7] OpenAI Models. Available online: https://platform.openai.com/docs/models/overview

[8] Howard, J.; Ruder, S. Universal language model fine-tuning for text classification. arXiv 2018, arXiv:1801.06146.

[9] J. Pradeep, E. Srinivasan and S. Himavathi,"Neural Network Based Recognition System Integrating Feature Extraction and Classification for English Handwritten", International Journal of Engineering (IJE) Transactions B: Applications Vol. 25, No. 2, (May 2019) 99-106.

[10] Hruday M. "Implementation of Handwritten Character Recognition using Neural Network".

[11] P. Shivakumara, D. Tang, M. Asadzadehkaljahi, T. Lu, U. Pal and M. Hossein Anisi, "CNN-RNN based Method for License Plate Recognition", CAAI Transactions on Intelligence Technology, Vol. 3, No. 3, pp. 169-175, 2018.

[12] Younus, S. B. S., S. Shajun Nisha, and M. Mohamed Sathik. "Comparative Analysis of Activation Functions in Neural Network for Handwritten Digits." Studies in Indian Place Names 40.71 (2020):793-799.

[13] Jagan Mohan Reddy D, A Vishnuvardhan Reddy "Recognition of Handwritten Characters using Deep Convolutional Neural Network".

[14] https://www.v7labs.com/blog/handwriting-recognition-guide

[15] Sullivan, M., Kelly, A., & McLaughlan, P. (2023). ChatGPT in higher education: Considerations for academic integrity and student learning. Journal of Applied Learning & Teaching, 6(1), 1-10.
[16] Gimpel, H., Et al. (2023). Unlocking the power of generative AI models and systems such as GPT-4 and ChatGPT for higher education: A guide for students and lecturers (No. 02-2023). Hohenheim Discussion Papers in Business, Economics and Social Sciences.
[17] Faycal, F. et al. (2023). Analyzing The Students' Views, Concerns, and Perceived Ethics about Chat GPT Usage. Computers and Education: Artificial Intelligence, 5.
[18] Chan, C. K. Y., & Lee, K. K. (2023). The AI generation gap: Are Gen Z students more interested in adopting generative AI such as ChatGPT in teaching and learning than their Gen X and Millennial Generation teachers?. arXiv preprint arXiv:2305.02878
[19] Bull, C., & Kharrufa, A. (2023). Generative AI Assistants in Software Development Education: A vision for integrating Generative AI into educational practice, not instinctively defending against it. IEEE Software, 1–9.
[20] Bozkurt, A. (2023). Unleashing the potential of generative AI, conversational agents and chatbots in educational praxis: A systematic review and bibliometric analysis of GenAI in education. Open Praxis, 15(4).
[21] Ardito, C.G. (2023). Contra generative AI detection in higher education assessments. arXiv preprint arXiv:2312.05241

X . Conclusion :

The OpenAI Notes Retrieval Assistant project marks a substantial stride in transforming how individuals engage with their notes. It merges cutting-edge search algorithms, natural language processing (NLP) methods, and personalized suggestions to empower users with an intuitive, efficient, and tailored solution for managing and accessing their notes efficiently. This initiative is centered on ensuring seamless integration, top-tier performance, and robust security, all aimed at elevating productivity and decision-making in diverse domains.

The significance of the OpenAI Notes Retrieval Assistant lies in its multifaceted capabilities. By leveraging advanced search algorithms, the assistant can swiftly locate relevant information within vast note repositories. These algorithms, integrated with NLP techniques, enable the assistant to understand natural language queries and provide precise results, enhancing user experience and accessibility. Moreover, the incorporation of personalized recommendations ensures that users receive content tailored to their preferences and work patterns, streamlining their workflow and optimizing their note retrieval process.

One of the primary objectives of this project is to deliver an intuitive interface that simplifies the note management experience. The assistant's user-friendly design allows for effortless navigation and interaction, making it accessible to users across different proficiency levels. By offering a seamless integration with existing note-taking platforms and applications, the assistant ensures compatibility and ease of adoption, thereby minimizing disruptions to established workflows.

Performance optimization is another key aspect of the OpenAI Notes Retrieval Assistant. Through continuous refinement and adaptation, the assistant prioritizes speed and accuracy in retrieving information, ensuring that users can access their notes promptly and reliably. This commitment to performance excellence is complemented by a robust security framework, safeguarding user data and ensuring confidentiality throughout the note management process.

In practical terms, the assistant's impact extends to various professional and personal domains. In academic settings, students and researchers benefit from efficient access to lecture notes, research materials, and study resources, facilitating knowledge acquisition and academic success. Similarly, in professional environments, the assistant aids in decision-making processes by providing quick access to project notes, meeting minutes, and critical information, empowering professionals to make informed decisions and drive productivity.

Furthermore, the OpenAI Notes Retrieval Assistant has broader implications for enhancing information management practices across industries. By promoting efficient note organization and retrieval, the assistant contributes to a more streamlined workflow, fostering collaboration and knowledge sharing within teams and organizations. This, in turn, cultivates a culture of productivity and innovation, where information is readily accessible and actionable.

In conclusion, the OpenAI Notes Retrieval Assistant embodies a forward-thinking approach to note management and information retrieval. By harnessing the latest advancements in search algorithms, NLP, and personalized recommendations, the assistant promises to redefine how users interact with their notes, offering a comprehensive and tailored solution that enhances productivity and decision-making efficiency across diverse domains. Its emphasis on integration, performance, and security underscores a commitment to empowering users with a transformative tool for note management and information access.

XI . Result and Output :

CODE :

```
import urllib
import warnings
from pathlib import Path as p
from pprint import pprint

import pandas as pd
from langchain import PromptTemplate
from langchain.chains.question_answering import load_qa_chain
from langchain.document_loaders import PyPDFLoader
from langchain.embeddings import VertexAIEmbeddings
from langchain.llms import VertexAI
from langchain.text_splitter import CharacterTextSplitter
from langchain.vectorstores import Chroma

pdf_loader = PyPDFLoader('exp_notes.pdf')
pages = pdf_loader.load_and_split()

vertex_llm_text = VertexAI(model_name="text-bison@001", max_output_tokens = 1000)
vertex_embeddings = VertexAIEmbeddings(model_name="textembedding-gecko@001")

text_splitter = CharacterTextSplitter(chunk_size=10000, chunk_overlap=0)
context = "\n\n".join(str(p.page_content) for p in pages)
texts = text_splitter.split_text(context)

vector_index = Chroma.from_texts(texts, vertex_embeddings).as_retriever()

question = " what is S/MIME  ?"

docs = vector_index.get_relevant_documents(question)

z = docs[0].page_content

context = " ".join(str(e.page_content) for e in docs)
```

```
def str_rm_whitespaces(ip_str):

    ip_str = ip_str.replace("\n", " ")
    ip_str = ip_str.replace("\\n", "")
    ip_str = ip_str.replace("\\u", "")
    ip_str = ip_str.replace("(", "")
    ip_str = ip_str.replace(")", "")
    ip_str = ip_str.replace("{", "")
    ip_str = ip_str.replace("}", "")
    ip_str = ip_str.replace(":", "")
    ip_str = ip_str.replace("'", "")
    ip_str = ip_str.replace(".", "")
    ip_str = ip_str.replace(",", "")

    while "  " in ip_str:
        ip_str = ip_str.replace("  ", " ")

    return ip_str
context = str_rm_whitespaces(context)

print(len(context.split()))

question_prompt_template = """
                Answer the question as precise as possible using the provided context. \n\n
                Context: \n {context} \n
                Question: \n {question} \n
                Answer:
                """
question_prompt = PromptTemplate(
    template=question_prompt_template, input_variables=["context", "question"]
)

# summaries is required. a bit confusing.
combine_prompt_template = """Given the extracted content and the question, find answer
in a docs and summerize answer in 1000 words.
If the answer is not contained in the docs, say "answer not available in context. \n\n
Summaries: \n {summaries}?\n
Question: \n {question} \n
```

```
Answer:
"""
combine_prompt = PromptTemplate(
    template=combine_prompt_template, input_variables=["summaries", "question"]
)

map_reduce_chain = load_qa_chain(
    vertex_llm_text,
    chain_type="map_reduce",
    return_intermediate_steps=True,
    question_prompt=question_prompt,
    combine_prompt=combine_prompt,
)

map_reduce_embeddings_outputs = map_reduce_chain(
    {"input_documents": docs, "question": question}
)

result = str_rm_whitespaces(map_reduce_embeddings_outputs["output_text"])

result

map_reduce_embeddings_outputs["output_text"]
```

Output :

```
question = " what is Huffmann Coding ?"
```

```
map_reduce_embeddings_outputs["output_text"]
```

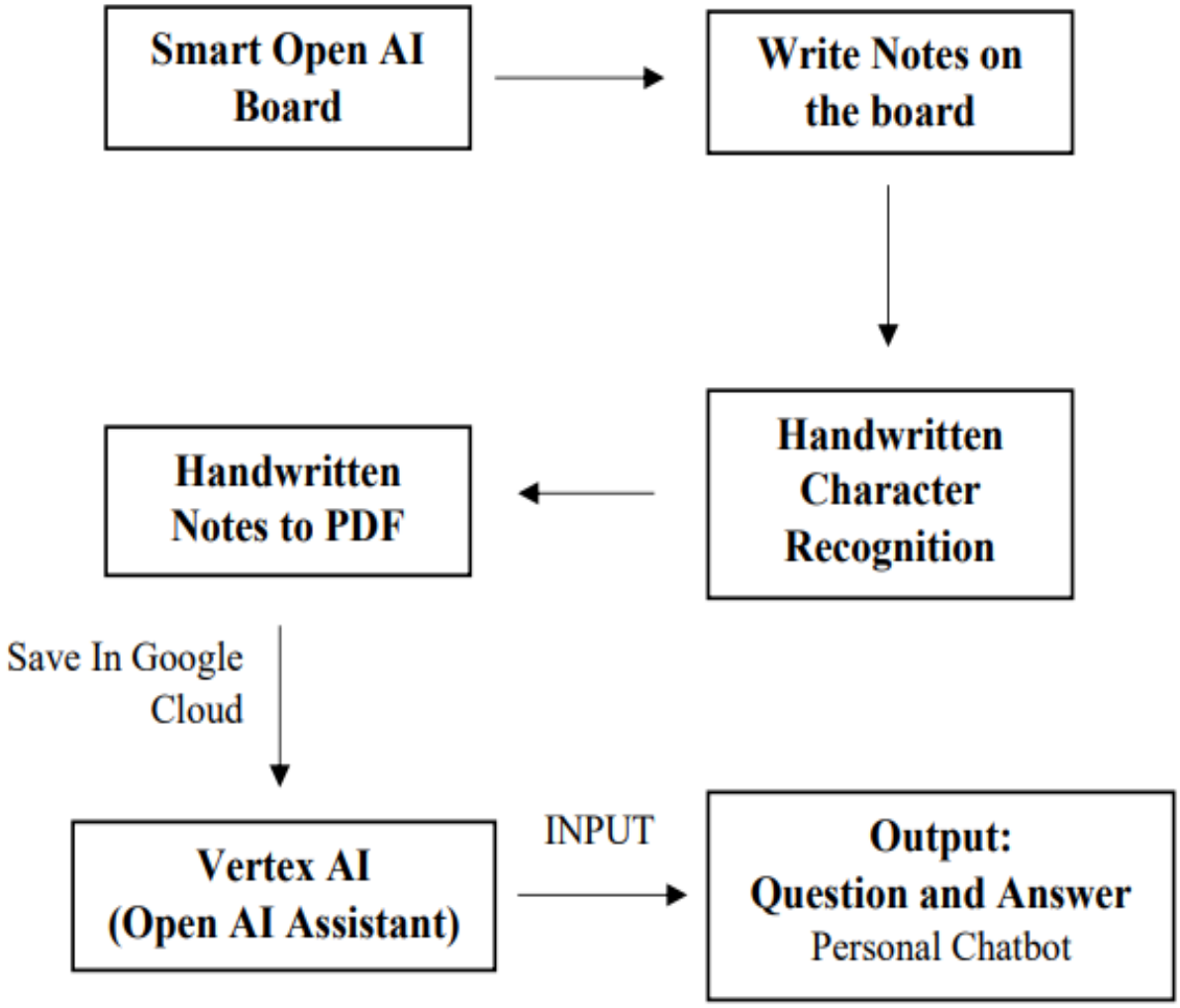
```
'Huffman coding is a variable-length prefix code that assigns shorter codes to more frequently occurring symbols and longer codes to less frequently occu
rring symbols. The codeword lengths are chosen such that the average codeword length is minimized. Huffman coding is an example of entropy coding, a clas
s of lossless data compression algorithms.\n\nHuffman coding is named after David A. Huffman, who published the algorithm in 1952. Huffman coding is a sp
ecial case of prefix codes, which are codes in which no codeword is a prefix of any other codeword. Prefix codes are used in many data compression algori
thms, including Huffman coding, arithmetic coding, and Lempel-Ziv coding.\n\nHuffman coding works by constructing a binary tree whose leaves are the symb
ols to be encoded. The internal nodes of the tree represent the codes for the symbols. The code for a symbol is the path from the root of the tree to the
leaf representing that symbol. The codeword lengths are chosen such that the average codeword length is minimized.\n\nHuffman coding can be used to compr
ess data of any type. However, it is most effective for data that has a high degree of statistical redundancy. For example, Huffman coding is often used
to compress text files, image files, and audio files.\n\nHuffman coding is a relatively simple and efficient algorithm. It is also a lossless algorithm,
which means that the original data can be perfectly reconstructed from the compressed data. However, Huffman coding is not always the best choice for dat
a compression. For some types of data, other algorithms, such as arithmetic coding or Lempel-Ziv coding, may be more effective.\n\nHuffman coding is a wi
dely used data compression algorithm. It is implemented in many popular compression software packages, including gzip, bzip2, and 7-zip.'
```

```
question = " what is S/MIME  ?"
```

```
map_reduce_embeddings_outputs["output_text"]
```

```
'S/MIME stands for "Secure/Multipurpose Internet Mail Extensions." It is a widely used security protocol that provides a way to secure and authenticate e
mail messages. S/MIME is primarily used for encrypting and digitally signing email messages to ensure the confidentiality, integrity, and authenticity of
the message content.\n\nS/MIME is based on the public key infrastructure (PKI) model. In this model, each user has a public key and a private key. The pu
blic key is used to encrypt messages, and the private key is used to decrypt messages. When a user sends an encrypted message to another user, the sender
\'s public key is used to encrypt the message. The recipient can then use the sender\'s private key to decrypt the message.\n\nS/MIME also supports digit
al signatures. A digital signature is a mathematical function that is used to verify the authenticity of a message. When a user digitally signs a messag
e, the user\'s private key is used to create a digital signature. The recipient can then use the user\'s public key to verify the digital signature. This
verification process ensures that the message was sent by the intended sender and that the message has not been tampered with.\n\nS/MIME is a powerful se
curity protocol that can be used to protect email messages from unauthorized access and tampering. However, it is important to note that S/MIME is not a
foolproof security measure. If a user\'s private key is compromised, then an attacker could use the private key to decrypt messages or forge digital sign
atures. Therefore, it is important to take steps to protect private keys, such as using strong passwords and storing private keys in a secure location.\n
\nS/MIME is supported by a variety of email clients, including Microsoft Outlook, Mozilla Thunderbird, and Apple Mail. To use S/MIME, you will need to in
stall an S/MIME certificate on your email client. S/MIME certificates can be obtained from a variety of certificate authorities.'
```

FUTURE SCOPE


Smart Open AI Board
Write Notes on the board
Handwritten Character Recognition
Handwritten Notes to PDF
Save In Google Cloud
Vertex AI (Open AI Assistant)
INPUT
Output: Question and Answer
Personal Chatbot

- Enhanced Personalization:
  Future iterations of the assistant can delve deeper into personalization, offering more tailored suggestions and insights based on individual preferences and behaviors. By leveraging machine learning techniques, the assistant can continually refine its understanding of user needs, providing increasingly relevant and timely note recommendations.

- Integration with IoT Devices:
  As the Internet of Things (IoT) ecosystem expands, the assistant can integrate with smart devices to offer seamless note-taking and retrieval experiences. Users could dictate notes through voice-controlled assistants or capture information directly from IoT-enabled devices, further enhancing accessibility and convenience.

- Collaborative Features:
  Collaboration is essential in many professional environments. The assistant could evolve to support real-time collaboration on notes, enabling multiple users to access, edit, and contribute to shared documents simultaneously. This feature would facilitate teamwork and streamline communication within organizations.

- Contextual Understanding:
  Improving the assistant's contextual understanding capabilities would enable it to interpret complex queries and provide more nuanced responses. By analyzing the context surrounding a note, such as related documents or previous interactions, the assistant can offer deeper insights and anticipate user needs more effectively.

- Advanced Security Measures:
  Given the importance of data security, future developments could focus on implementing advanced encryption techniques and privacy controls. This would ensure that user data remains protected throughout the note management process, fostering trust and compliance with stringent security standards.

- Cross-Platform Compatibility:
  To accommodate diverse user preferences and workflows, the assistant could extend its support to a wider range of platforms and applications. Seamless integration with popular note-taking tools, project management software, and communication platforms would enhance interoperability and simplify information sharing.

- Analytics and Insights:
  Leveraging data analytics, the assistant could provide valuable insights into note-taking behaviors and productivity trends. By analyzing patterns in note usage, the assistant can identify opportunities for optimization and offer actionable recommendations to improve efficiency.

- Mobile and Offline Capabilities:
  Supporting mobile devices and offline access would broaden the assistant's reach, allowing users to manage and retrieve notes anytime, anywhere. Offline synchronization capabilities ensure continuity of service even in environments with limited connectivity.

- Adaptation to Specialized Domains:
  Tailoring the assistant's capabilities to specific industries or domains (e.g., healthcare, legal, finance) would unlock new use cases and address unique challenges faced by professionals in these sectors. Customized features and integrations could cater to specialized workflows and compliance requirements.

- Ethical Considerations and Bias Mitigation:
  Continual efforts to address ethical considerations, such as bias mitigation in AI algorithms, are essential for fostering responsible usage of the assistant. Transparency in algorithmic decision-making and ongoing audits can help build trust and accountability among users.