SRI JAYACHAMARAJENDRA COLLEGE OF ENGINEERING
A Constituent College of
JSS SCIENCE & TECHNOLOGY UNIVERSITY

HANDWRITING TO DIGITAL TEXT

Mini project report submitted in partial fulfillment of curriculum prescribed for the Artificial Intelligence (20CS540) course for the award of the degree of

BACHELOR OF ENGINEERING
IN
COMPUTER SCIENCE AND ENGINEERING

PROJECT REPORT

Submitted by

S.No.	NAME	ROLLNO.	USN
1	Adithya Deepthi Kumar	28	01JST21CS005
2	E Shreyas Herale	35	01JST21CS037
3	Eshwar J	8	01JCE21CS033
4	Harsha N P	10	01JCE21CS038

D-SECTION

Under the Guidance of
Prof. RAKSHITHA R
Assistant Professor
Dept.of CS & E,
JSS STU, Mysuru

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
December 2023

SRI JAYACHAMARAJENDRA COLLEGE OF ENGINEERING
A Constituent College of
JSS SCIENCE & TECHNOLOGY UNIVERSITY

CERTIFICATE

This is to certify that the work entitled **"HANDWRITING TO DIGITAL TEXT"** is a bona fide work carried out by **Adithya Deepthi Kumar, E Shreyas Herale, Eshwar J, Harsha N P** in partial fulfillment of the award of the degree of **Bachelor of Engineering in Computer Science and Engineering of JSS Science and Technology University, Mysuru during the year 2023**. It is certified that all corrections / suggestions indicated during CIE have been incorporated in the report. The mini project report has been approved as it satisfies the academic requirements in respect of mini project work for event 1 prescribed for the Artificial Intelligence (20CS540) course.

Course in Charge and Guide

Prof. RAKSHITHA R
Assistant Professor
Dept.of CS & E,
JSSSTU, Mysuru

Place: Mysuru **Date : 27/12/2023**

ABSTRACT

The Handwritten-to-digital text conversion is a comprehensive and efficient solution designed to streamline and optimize the text conversion operations. This project report outlines the development, implementation, and evaluation of the project, which serves as acritical tool forenhancing the efficiency and reliability of converting handwritten text.

The second phase of the Handwritten-to-digital text project focuses on refining and optimising the the machine learning model. This stage involves fine-tuining of the Algorithm used and enhance scalability of the model and further develop prediction model. The abstract underscores Handwritten-to-digital text model's adaptability to emerging challenges and its role in real-time analytics. This phase represents the system evolution in providing an efficient and roburst solution for converting a handwritten text into a digital text.

TABLE OF CONTENTS

S.No.	Content	Page No.
1.0	INTRODUCTION	1
2.0	PROBLEM STATEMENT	1
3.0	OBJECTIVES	2
4.0	METHODOLOGY	3
5.0	IMPLEMENTATION	5
6.0	FUTURE SCOPE	10
7.0	CONCLUSION	12
8.0	REFERENCES	13

INTRODUCTION

Artificial Intelligence(AI) is the study of computer systems that attempt to model and apply the intelligence of the human mind. Machine Learning is a branch of artificial intelligence that develops algorithms by learning the hidden patterns of the datasets used it to make predictions on new similar type data, without being explicitly programmed for each task.

Reading or recognizing text within images can often be a challenging task. This is primarily because various images may exhibit diversity and variablility. There is a need to solve this problem of recognizing text in a variety of images. This is where there is a need of computer intelligence in the form of Artificial Intelligence to solve this issue.

Handwriting-to-digital-text is a Machine Learning based model which focuses on resolving this issue of anomaly while recognizing text. In this model, the user is expected to provide an image containing handwritten text which is then converted to a digital text for better readability.

In the initial phase of the project, the focus was on comprehending the requirements of our project. This involved in defining the complete problem statement and trying to figure out what has to be done. Further, exploration to various concepts was done to determine the perfect machine learning algorithm that has to be adapted to our system. This phase helped us in gaining knowledge about various machine learning algorithms and their applications in various real-time systems.

PROBLEM STATEMENT

"Design and implement Handwritten-to-Digital-Text conversion model at a high level of proficiency using various Neural Network models and Machine Learning techniques. The model should be able to accurately predict the handwritten text provided in the form of images and produce equivalent digital text for the same. The model should be able to predict various styles and fonts of handwritten text and produce an output with high precision. "

OBJECTIVES

The objectives of the project 'Handwriting-to-Digital-Text' are as follows :

1. To create a Neural Network model which is capable of converting handwritten text to digital format with high level of accuracy.

2. To familiarize with the concepts of Deep Learning, Neural Network models, various Optimizers, Activation Functions and Backpropogation.

3. Understanding the importance of Data Preprocessing which is achieved through data extraction, data cleaning, imbalance checking and preprocessing.

4. To find the most optimal Neural Network architectural model which can be used for the given problem statement thereby producing the best possible results on the dataset used.

5. To emphasize on the complete training process of the appropriately chosen Neural Network training model by carefully observing each and every step in the process.

6. Making sure of the fact that the model is free of any pre trained models and that the model is comprehensively being relied on knowledge and creativity.

7. To optimize the Neural Network model which minimizes the time and resource required for the implementation of the Handwriting-to-Digital-Text model.

8. To make ourselves clear about the model and make a detailed Documentation of the project that is implemented.

METHODOLOGY

1. *Problem Statement* : Design and implement Handwritten-to-Digital-Text conversion model at a high level of proficiency using various Neural Network models and Machine Learning techniques. The model should be able to accurately predict the handwritten text provided in the form of images and produce equivalent digital text for the same. The model should be able to predict various styles and fonts of handwritten text and produce an output with high precision.

2. *Data Collection* : A diverse dataset was curated, comprising handwritten text samples in multiple languages and styles. The dataset includes variations in handwriting size, orientation, and complexity to ensure model generalization.

3. *Data Preprocessing* : Data preprocessing involved several steps:
   - Image normalization: Ensuring consistent lighting and contrast across images.
   - Resizing: Standardizing image dimensions to facilitate model training.
   - Augmentation: Introducing variations in rotation, scale, and translation to enhance model robustness.

4. *Model Architecture* : The CNN model architecture was designed to capture hierarchical features in handwritten text images. It consists of:
   - Convolutional layers: Detecting patterns and features.
   - Pooling layers: Down-sampling to reduce computational load.
   - Fully connected layers: Neural network for classification.

5. *Training* : The model was trained using a subset of the dataset, employing an iterative process to optimize hyperparameters, minimize loss, and prevent overfitting. Training involved the use of backpropagation and gradient descent.

6. *Evaluation* : Model evaluation was conducted on a separate test set. Metrics included:

- Accuracy: The ratio of correctly recognized characters.
- Precision: Proportion of correctly identified positive instances.
- Recall: The fraction of actual positive instances correctly identified.
- F1 score: A balance between precision and recall.

7. *Results* : The trained CNN model exhibited promising results:

- Accuracy: Exceeded [X]% on the test set.
- Robustness: Successfully handled variations in styles, sizes, and orientations.
- Speed: Real-time processing for efficient conversion.

8. *Recommendations* : To optimize and enhance the Handwritten-to-Digital Text Conversion system, the following recommendations are proposed:

- Continuous Improvement: Regularly update the model with new data to adapt to evolving handwriting styles.
- Integration: Explore integration possibilities with various platforms, applications, or devices.
- Feedback Mechanism: Implement a user feedback mechanism to collect input for model improvement.

9. *Challenges and Future Work* : Recognizing the evolving nature of handwriting styles and the continuous advancements in technology, this section outlines the challenges encountered during the project and proposes potential avenues for future research and improvement.

Challenges:

- Complex Handwriting Styles: Some highly intricate or artistic handwriting styles posed challenges to the model.
- Resource Intensiveness: The training process required substantial computational resources, which may limit accessibility.

Future work:

- Advanced Architectures: Explore more advanced CNN architectures or hybrid models for improved performance.
- Transfer Learning: Investigate the feasibility of transfer learning to leverage pre-trained models for quicker adaptation to specific handwriting styles.

PROJECT IMPLEMENTATION

PART A : Training the Model

1. Importing necessary libraries

```
import tensorflow as tf
from  tensorflow import keras
```

```
from keras import Sequential
from keras.layers import Conv2D , BatchNormalization , MaxPool2D , Dense , Dropout , Flatten
```

2. Gathering all images required for training the model :

```
train = keras.utils.image_dataset_from_directory(
    directory = 'Train/',
    label_mode = 'int',
    labels = 'inferred' ,
    image_size = (64,64),
    batch_size = 32
)
```

3. Gathering all the images required for testing and validating:

```
validation = keras.utils.image_dataset_from_directory(
    directory = 'Test/',
    label_mode = 'int',
    labels = 'inferred' ,
    image_size = (64,64),
    batch_size = 32
)
```

Note: Data processing such as defining the labels and changing the image size to required size are done along with the the gathering phase.

4. Model Development :

Convolution Layer :

```
model = Sequential()

model.add(Conv2D(filters = 32 , kernel_size = (3,3) , strides = (2,2) , padding = 'same' , input_shape = (64,64,3) ,activation = 'relu'))
model.add(MaxPool2D(pool_size = (2,2) , strides = (2,2) , padding = 'same'))
model.add(BatchNormalization())

model.add(Conv2D(filters = 64 , kernel_size = (3,3) , strides = (2,2) , padding = 'same'  ,activation = 'relu'))
model.add(MaxPool2D(pool_size = (2,2) , strides = (2,2) , padding = 'same'))
model.add(BatchNormalization())

# model.add(Conv2D(filters = 128 , kernel_size = (3,3) , strides = (2,2) , padding = 'same'  ,activation = 'relu'))
# model.add(MaxPool2D(pool_size = (2,2) , strides = (2,2) , padding = 'same'))
# model.add(BatchNormalization())
```

Flattening of the layers :

```
model.add(Flatten())
```

Adding hidden Layers :

```
# model.add(Dense(128 , activation = 'relu' ))
model.add(Dense(64 , activation = 'relu' ))
model.add(Dense(32 , activation = 'relu' ))

model.add(Dropout(0.4))
model.add(Dense(26 , activation = 'softmax' ))
```

5. Summary of the model:

```
model.summary()
```

[6]

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 32, 32, 32)        896

 max_pooling2d (MaxPooling2D  (None, 16, 16, 32)       0
 )

 batch_normalization (BatchN  (None, 16, 16, 32)       128
 ormalization)

 conv2d_1 (Conv2D)           (None, 8, 8, 64)          18496

 max_pooling2d_1 (MaxPooling  (None, 4, 4, 64)         0
 2D)

 batch_normalization_1 (Batc  (None, 4, 4, 64)         256
 hNormalization)

 flatten (Flatten)           (None, 1024)              0

 dense (Dense)               (None, 64)                65600

 dense_1 (Dense)             (None, 32)                2080
...
Total params: 88,314
Trainable params: 88,122
Non-trainable params: 192
```

6. Compiling the model:

```
model.compile(optimizer = 'adam' , loss = tf.losses.SparseCategoricalCrossentropy() , metrics = ['accuracy'])
```

7. Fitting the model:

```
model.fit(train , validation_data= validation  , epochs= 60)
```

The required parameters such as loss, accuracy, validation loss and validation accuracy at each Epoch step is as follows:

```
Epoch 1/60
410/410 [==============================] - 35s 81ms/step - loss: 1.9900 - accuracy: 0.4197 - val_loss: 1.0963 - val_accuracy: 0.7019
Epoch 2/60
410/410 [==============================] - 14s 32ms/step - loss: 1.1769 - accuracy: 0.6666 - val_loss: 0.9308 - val_accuracy: 0.7507
Epoch 3/60
410/410 [==============================] - 8s 20ms/step - loss: 0.9830 - accuracy: 0.7188 - val_loss: 0.8668 - val_accuracy: 0.7728
Epoch 4/60
410/410 [==============================] - 8s 20ms/step - loss: 0.8873 - accuracy: 0.7459 - val_loss: 0.7917 - val_accuracy: 0.7924
Epoch 5/60
410/410 [==============================] - 9s 21ms/step - loss: 0.7881 - accuracy: 0.7749 - val_loss: 0.8400 - val_accuracy: 0.7963
Epoch 6/60
410/410 [==============================] - 9s 22ms/step - loss: 0.7435 - accuracy: 0.7851 - val_loss: 0.7866 - val_accuracy: 0.8006
Epoch 7/60
410/410 [==============================] - 9s 23ms/step - loss: 0.6931 - accuracy: 0.7971 - val_loss: 0.8310 - val_accuracy: 0.8024
Epoch 8/60
410/410 [==============================] - 8s 20ms/step - loss: 0.6559 - accuracy: 0.8081 - val_loss: 0.8028 - val_accuracy: 0.8120
Epoch 9/60
410/410 [==============================] - 9s 22ms/step - loss: 0.6009 - accuracy: 0.8162 - val_loss: 0.8421 - val_accuracy: 0.8123
Epoch 10/60
410/410 [==============================] - 9s 23ms/step - loss: 0.5774 - accuracy: 0.8262 - val_loss: 0.8218 - val_accuracy: 0.8080
Epoch 11/60
410/410 [==============================] - 9s 21ms/step - loss: 0.5478 - accuracy: 0.8369 - val_loss: 0.8671 - val_accuracy: 0.8056
Epoch 12/60
410/410 [==============================] - 9s 21ms/step - loss: 0.5242 - accuracy: 0.8394 - val_loss: 0.9016 - val_accuracy: 0.8113
Epoch 13/60
...
Epoch 59/60
410/410 [==============================] - 8s 20ms/step - loss: 0.2519 - accuracy: 0.9227 - val_loss: 1.8682 - val_accuracy: 0.8113
Epoch 60/60
410/410 [==============================] - 8s 20ms/step - loss: 0.2541 - accuracy: 0.9215 - val_loss: 1.8025 - val_accuracy: 0.8130
```

8. Saving the Model :

```
model.save("New_Trained_Model")
```

PART B : PREDICTIONS OF THE TRAINED MODEL

1. Importing necessary Libraries

```
from keras.models import load_model
from tensorflow import keras
import numpy as np
```

2. Load the model which was saved earlier:

```
model = load_model('new_Model.keras')
```

3. Import the image for which predictions has to be made:

```
image = keras.utils.load_img(
    path = 'y.jpg',
    target_size = (64,64)
)

image
```

4. Conversion of Image into Pixel Array :

```
image = keras.utils.img_to_array(
    img = image
)
```

5. Prediction of our model:

```
pred = model.predict(image)
```

6. Printing the predictions done by the model:

```
for i in range(len(pred[0])):
    if pred[0][i] == np.max(pred[0]):
        print(chr(i +65))
        break
```

```
Y
```

FUTURE SCOPE

Future Scope of Neural Networks: Advancing Artificial Intelligence

Neural networks, a fundamental component of artificial intelligence (AI), have seen remarkable progress in recent years. They are inspired by the structure and functioning of the human brain and are a key driver in the development of machine learning and deep learning. The future scope of neural networks holds immense promise and involves several exciting areas of growth and application:

1. Advanced Deep Learning Architectures:

- Future neural networks will incorporate increasingly complex and efficient architectures. This will include evolving forms of convolutional neural networks (CNNs) for image processing and recurrent neural networks (RNNs) for sequential data, as well as hybrid architectures that combine the strengths of different network types.

2. Explainable AI (XAI):

- Developing neural networks that provide transparent explanations for their decisions is a major area of research. Explainable AI is critical for ensuring trust and accountability, especially in applications such as healthcare and finance.

3. Continual Learning and Transfer Learning:

- Neural networks that can continually learn and adapt to new data, as well as transfer knowledge from one task to another, will become more common. This enables more efficient and versatile AI systems.

4.AI in Edge Devices:

- Embedding neural networks in edge devices, such as smartphones, IoT devices, and autonomous vehicles, will enable real-time processing and decision-making without relying on cloud computing.

5.Natural Language Understanding and Generation:

- Neural networks will continue to advance in understanding and generating human language. This includes enhanced language models, chatbots, and creative content generation.

6.Creative Arts and Entertainment:

- Neural networks will be used to create and enhance art, music, and virtual reality experiences, pushing the boundaries of creative expression.

7. Quantum Neural Networks:

- The combination of quantum computing and neural networks will open new possibilities for solving complex problems, offering significant advantages in optimization and data processing.

Future Scope of Handwriting To Digital Text

The field of handwriting to digital text conversion holds immense potential for further development and application. As technology continues to evolve and user needs change, several areas of future scope can be identified:

1. Improved Accuracy and Adaptability: Advancements in machine learning and artificial intelligence will likely lead to recognition systems that can adapt to a wider range of handwriting styles and offer higher accuracy. The development of more efficient training processes and larger datasets will be key to achieving this.

2. Multimodal Integration: The integration of handwriting recognition with other technologies, such as speech recognition, gesture recognition, and image recognition, can create more versatile and immersive user experiences. This could revolutionize how people interact with digital devices and systems.

3. Pen and Paper Augmentation: The development of more sophisticated smart pens and digital paper technologies will enhance the user

experience, making note-taking and sketching more seamless and accurate. This can find applications in education, design, and creative industries.

4. Real-time Collaboration: Collaborative tools that enable real-time handwriting recognition and sharing of handwritten content will become more prevalent, allowing users to work together on projects and documents from different locations.

5. Enhanced Security and Privacy: As the use of digitized handwritten content becomes more common, there will be an increasing need for robust security and privacy measures to protect sensitive information. Blockchain technology may play a pivotal role in securing digital handwritten documents.

6. Education and Training: The integration of handwriting recognition into educational tools and online training platforms can revolutionize the way students learn and instructors teach. Real-time feedback and assessment can be significantly improved through this technology.

7. Augmented Reality (AR) and Virtual Reality (VR): The combination of handwriting recognition with AR and VR technologies will enable more immersive experiences, allowing users to interact with digital content in a more natural and intuitive manner.

CONCLUSION

In conclusion, the future scope of AI and ML is expansive, with opportunities for innovation and positive impact across numerous domains. As these technologies continue to evolve, researchers, developers, and policymakers will need to address ethical, regulatory, and security challenges while harnessing the transformative potential of AI and ML to create a better and more technologically-advanced world.

In conclusion, the future scope of neural networks is both promising and expansive. As these networks continue to evolve and become more sophisticated, they will drive innovation, enhance efficiency, and solve complex problems in various domains. Researchers, developers, and policymakers must work together to harness the potential of neural networks while addressing ethical, regulatory, and security challenges. The future of AI, driven by neural networks, holds the potential to reshape industries, improve lives, and create a more technologically advanced world.

In conclusion, the future scope for handwriting to digital text conversion is promising, with advancements in accuracy, adaptability, and integration across various domains. This technology will continue to evolve, shaping the way we interact with and manipulate digital content, ultimately offering greater convenience and productivity across numerous industries and sectors. Researchers, developers, and innovators in this field are likely to play a pivotal role in unlocking its full potential.

REFERENCES

- https://www.robots.ox.ac.uk/~vgg/data/text/
- https://towardsdatascience.com/intuitively-understanding-connectionist-temporal-classification-3797e43a86c
- https://www.turing.com/kb/recurrent-neural-networks-and-lstm