CS361: Artificial Intellegence

(COVER SHEET)

Project Name: An Automated Optical Character Recognition of Handwritten English Letters using Decision Trees & Random Forests.

Team Information:

	ID [Ordered by ID]	Full Name [In Arabic]	Attendance [Handwritten Signature]	Final Grade
1	201901009	أحمد عيد فوزي سيد		
2	202000223	تسنيم سامح سليمان محمود		
3	202000664	كريم سلامه عبد الملك عثمان		
4	202000945	منة الله محمد سيد عيسوى		
5	202000958	مهند نصر الدين زغلول حسين		
6	202001066	يمنى محمد عبد القادر عبد الغني		

OCR Project Documentation

Table of Contents:

Additional Papers
Source Code
GUI Implementation

1.Project idea in details

Optical character recognition (OCR) technology is an efficient business process that saves time, cost and other resources by utilizing automated data extraction and storage capabilities.
Optical character recognition (OCR) is sometimes referred to as text recognition. It extracts and repurposes data from scanned documents, camera images and image-only pdfs.

OCR systems can use a combination of hardware and software to convert physical, printed documents into machine-readable text. Hardware — such as an optical scanner or specialized circuit board — copies or reads text; then, software typically handles the advanced processing.

OCR software can take advantage of artificial intelligence (AI) to implement more advanced methods of intelligent character recognition (ICR), like identifying languages or styles of handwriting. The process of OCR is most commonly used to turn hard copy legal or historical documents into pdf documents so that users can edit, format and search the documents as if created with a word processor.

How does optical character recognition work?
Optical character recognition (OCR) uses a scanner to process the physical form of a document. Once all pages are copied, OCR software converts the document into a two-color or black-and-white version. The scanned-in image or bitmap is analyzed for light and dark areas, and the dark areas are identified as characters that need to be recognized, while light areas are identified as background. The dark areas are then processed to find alphabetic letters or numeric digits. This stage typically involves targeting one character, word or block of text at a time. Characters are then identified using one of two algorithms — pattern recognition or feature recognition.

Pattern recognition is used when the OCR program is fed examples of text in various fonts and formats to compare and recognize characters in the scanned document or image file.
Feature detection occurs when the OCR applies rules regarding the features of a specific letter or number to recognize characters in the scanned document. Features include the number of angled lines, crossed lines or curves in a character. For example, the capital letter "A" is stored as two diagonal lines that meet with a horizontal line across the middle. When a character is identified, it is converted into an ASCII code (American

Standard Code for Information Interchange) that computer systems use to handle further manipulations.

2. Main functionalities

Optical Recognition has been one of the most challenging research areas in the field of image processing in recent years. Through going through multiple papers and searching across the internet and books we found that the main functionalities, that we're implementing, are the following stages:

1. Pre-processing

The pre-processing stage takes a raw image and performs some operations to make the algorithms and processing more efficient. These operations are:
1. Thresholding: Raw image either color or gray is converted into binary image.
2. Noise reduction: Various techniques like morphological operations are used to connect unconnected pixels, to remove isolated pixels, to smooth pixel boundaries.
3. Normalization: the process of translating data into the range [0, 1] (or any other range) or simply transforming data onto the unit sphere.

2. Segmentation

Segmentation is nothing but breaking the whole image into subparts to process them further.

Segmentation of image is done in the following sequence :
→ Line level Segmentation
→ Word level Segmentation
→ Character level Segmentation

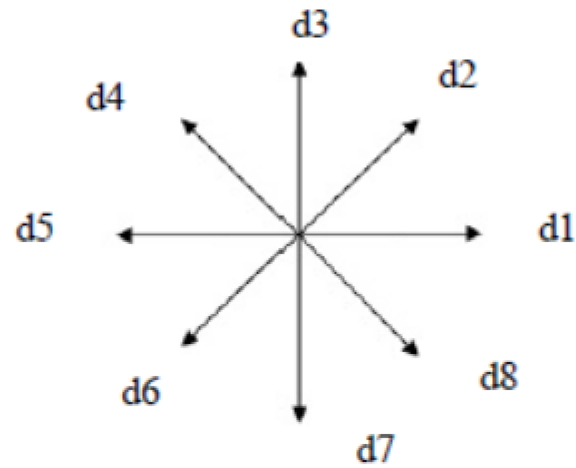
3. Feature extraction

The feature extraction stage is used to extract the most relevant information from the text image which helps us to recognize the characters in the text. The selection of a stable and representative set of features is the heart of pattern recognition system design. The classification stage uses the features extracted in the previous stage to identify the text segment according to preset rules.

Starts with,

1. Zoning: The frame containing the character is divided into several overlapping or non-overlapping zones and the densities of object pixels in each zone are calculated. Density is calculated by finding the number of object pixels in each zone and dividing it by total number of pixels.

2. Projection Histogram Features: Projection histograms count the number of pixels in specified direction. There are three types of projection histograms. horizontal. vertical. Left diagonal and right diagonal.
3. Distance Profile Features: Profile counts the number of pixels (distance) from bounding box of character image to outer edge of character. In this approach, profiles of four sides left, right, top and bottom were used.
4. Background Directional Distribution (BDD) Features: To calculate directional distribution values of background pixels for each foreground pixel, we have used the masks for each direction. An example of sample:



5. Combination of various features: Each feature is used to form a feature vector hence if we use a combination of features then it will help us to derive the feature vectors with more elements which are helpful to increase the efficiency of recognition

4. Classification

In this process we start developing the algorithms that is going to train on classifying the characters into their places according to the features related to them decided by feature selection process.

3. Similar applications in the market

	Languages	Free Trial	Price	Deployment	Rate

ReadIRIS	130+ languages	30-day	$129	Windows, and Mac	9/10
ABBY FineReader	198+ languages	7-day	$117 / year	Windows, iOS, Android, and Mac.	9/10

ReadIRIS

ReadIRIS is a powerful and accurate OCR engine that can be used to convert scanned documents and images into editable and searchable text. It offers a wide range of features and options, making it a versatile and powerful OCR solution for a variety of needs.

ReadIRIS considered one of the best OCR software for scanned documents and invoices. It is a fast and accurate optical character recognition software that can recognize text in over 130 languages. It is easy to use and can be integrated into your workflow.

It has a multitude of options for converting scanned files into editable documents as well as modifying PDF files.

This document digitization software works on both Windows and Mac, and you can try out all the features for free for 30 days.

Key Features:

- 20% faster document processing
- Edit texts embedded in your images with OCR
- Convert Microsoft Office documents to PDF
- Annotate and comment
- Protect and sign PDFs
- Integration with printers (Twain scanners)

ABBYY FineReader PDF

ABBYY FineReader PDF is the best OCR engine that can help you convert PDFs into editable text files with ease. This powerful tool leverages AI-based OCR technology that can recognize text in over 198 languages, making it perfect for converting PDFs from all over the world.

Not only does it make converting PDFs easy, but it also offers a number of features that can make your life easier, such as the ability to convert scanned PDFs, the ability to convert PDFs with images, and the ability to convert password-protected PDFs.

This OCR solution has been in the market for 28 years with over 100 million installations including 17.000 corporate users.

This OCR application includes a 7-day free trial for individuals and a 30-day free trial for enterprises, as well as support for four distinct platforms, including Windows, iOS, Android, and Mac.

ABBYY's AI-based OCR and document-conversion technologies ensure high levels of accuracy and maintain layout and structure for further effortless editing.

This document digitization software is best for individuals and enterprises and is considered the best OCR software for invoices processing.

Key Features:

- Create, edit, and organize PDFs
- Collaborate on and approve PDFs
- High-quality editing and document comparison tools
- Sophisticated and flexible OCR settings/adjustments
- High levels of accuracy

Cons:

Small fonts processing needs improvements

Conversion to Microsoft Word and Excel needs improvements

4. An initial literature review of Academic publications

A. A STUDY ON OPTICAL CHARACTER RECOGNITION TECHNIQUES Narendra Sahu and Manoj Sonkusare

ABSTRACT:

Optical Character Recognition (OCR) is the process which enables a system to without human intervention identifies the scripts or alphabets written into the users' verbal communication. Optical Character identification has grown to be individual of the mainly flourishing applications of knowledge in the field of pattern detection and artificial intelligence. In our survey we study on the various OCR techniques. In this paper we resolve and examine the hypothetical and numerical models of Optical Character Identification. The Optical character identification or classification (OCR) and Magnetic Character Recognition (MCR) techniques are generally utilized for the recognition of patterns or alphabets. In general the alphabets are in the variety of pixel pictures and it could be either handwritten or stamped, of any series, shape or direction etc. Alternatively in MCR the alphabets are stamped with magnetic ink and the studying machine categorize the alphabet on the basis of the exclusive magnetic field that is shaped by every alphabet. Both MCR and OCR discover utilization in banking and different trade appliances. Earlier exploration going on Optical Character detection or recognition has shown that the In Handwritten text there is no limitation lying on the script technique. Hand written correspondence is complicated to be familiar through due to vdiverse human handwriting style, disparity in angle, size and shape of calligraphy. An assortment of approaches of Optical Character Identification are discussed here all along through their achievement.

1. INTRODUCTION

Optical Character Identification is single of the majority enthralling and demanding areas of pattern recognition with a variety of realistic applications. The times departed by allow distinguish us that OCR expertise has been built by lots of researchers over a long period of time, consisting unreservedly of impressive like a worldwide human research network. In such an imperceptible discussion, people have made efforts, with "antagonism and collaboration," to advance the research effort. In this way, global symposiums and inductions are being determined to stimulate the improvement in the domain. For example the global induction on Frontiers in Handwriting detection and the International discussion on article psychoanalysis and Recognition determination play an explanation task in the intellectual and matter-of-fact arena.

All neural networks want data to be skilled. These data consist of input and target data. The guidance of one alphabet could be done with an input and an output vector. The length of the input vector will be the resolution of the input. I.e. if they worked with the fourth network, then the length will be 77 elements, every row coming after each other in it. The elements could have values 0 or 1. If the grid part of the image represented by the vector constituent has more than 50% coverage of the letter's features, after that the vector element's value is 1, otherwise it is 0. By checking this for all of the grid parts, we get the vector. During the primary neural network we have 10 numbers. This funds we will have 10 rows of this kind of vectors, every row representing a number. This describes the input matrix. This kind of data form is expected as inputs through Matlab in the neural network. For a network with N outputs, the output matrix will be an NxN dimension matrix, filled with zeros, and having ones on the main diagonal. This step of the recognition was done manually.

Until a few decades previously, explore within the domain of Optical Character identification was restricted to document images acquired with flatbed desktop scanners. The efficiency of such methods is incomplete as they aren't exchangeable because of huge dimension of the digital

scanners and the requirement of a computing method. Furthermore, the capturing speed of a digital scanner is dull than that of a high resolution digital camera. Recently, with the development of dispensation velocity and interior recollection of mobile devices such as high-end cell-phones, Individual Digital Assistants (IDA), smart phones, iPhones, iPods, etc. having built- in digital cameras, an innovative trend of exploration has emerged into representation. Researchers have dared to imagine of organization OCR applications on such strategy for having authentic instant outcome. A habitual big industry credential Reader (ICR), destined for repeated inhabitants of relevant contact in sequence starting a trade card also recognized as visiting card into the phone book of the diplomacy is an example of such applications. Conversely, computing in hand held approach engages a technique to of challenge. Since of the non contact environment of digital cameras connected to hand held machines, obtained descriptions terribly often undergo from skew and perception bend.

B. MaskOCR: Text Recognition with Masked Encoder -Decoder Pretraining

Pengyuan Lyu, Chengquan Zhang, Shanshan Liu, Meina Qiao, Yangliu Xu,

Liang Wu, Kun Yao, Junyu Han, Errui Ding, Jingdong Wang

Department of Computer Vision Technology (VIS), Baidu Inc.

Abstract:

In this paper, we present a model pre training technique, named MaskOCR, for text recognition. Our text recognition architecture is an encoder-decoder transformer: the encoder extracts the patch-level representations, and the decoder recognizes the text from the representations. Our approach pre trains both the encoder and the decoder in a sequential manner. (i) We pretrain the encoder in a self-supervised manner over a large set of unlabeled real text images. We adopt the masked image modeling approach, which shows the effectiveness for general images, expecting that the representations take on semantics. (ii) We pretrain the decoder over a large set of synthesized text images in a supervised manner and enhance the language modeling capability of the decoder by randomly masking some text image patches occupied by characters input to the encoder and accordingly the representations input to the decoder. Experiments show that the proposed MaskOCR approach achieves superior results on the benchmark datasets, including Chinese and English text images.

1 Introduction

Optical character recognition aims to recognize texts within a digital image, e.g., a scanned document, a photo of a document, a scene-photo, and so on. It has wide-range applications, such as visual search, document digitization, and so on. Optical character recognition generally consists of two tasks: text detection, localizing the text region, and text recognition, identifying the text from the localized region, which is the interest of this paper. There are three main pipelines for text recognition. (i) The character-based pipeline [48; 23; 29] localizes each character, performs character recognition, and then groups them into words. (ii) The word-based pipeline [21] performs word classification directly. (iii) The sequence-based pipeline [42; 43; 50; 12] regards text recognition as a sequence labeling problem, and is adopted by most deep learning methods.

We follow the sequence-based pipeline and adopt an encoder-decoder transformer for text recognition. The encoder is a ViT architecture [11], a sequence of self-attention and FFN blocks, for text image patch representation extraction. The decoder is formed with the DETR-style [3] decoder, a sequence of self-attention, cross-attention and FFN blocks, mapping the patch representations to a text with an expected role of language modeling [27]. Our main work lies in exploring the pretraining technology for text recognition. Our approach, calledMaskOCR, pre trains both the encoder and the decoder with the masking strategy in a sequential manner. We follow the self-supervised pre training framework and adopt a masked image modeling

∗Corresponding Author.

arXiv:2206.00311v1 [cs.CV] 1 Jun 2022

approach to pretrain the encoder for semantic patch representation learning. We divide the text image into a set of vertical patches, and randomly mask some patches that may contain a part of

some character, or some whole characters. We predict the representations of the masked patches from the visible patches in the representation space learned from the encoder, and map the predicted representations to the masked patch images. We pretrain the decoder in a supervised manner with the masking strategy for language modeling over synthesized text images. We fix the pretrained encoder and only update the decoder, so that this pretraining task explores the language rule and the encoder is not affected by the synthesized text image style that might be different from the downstream tasks. We validate the effectiveness of the proposed MaskOCR approach on the benchmark datasets, including Chinese and

English text images. The experiments show that our approach achieves superior results over previous text recognition methods.

C. Scene Text Telescope: Text-Focused Scene Image Super-Resolution

Jingye Chen, Bin Li*, Xiangyang Xue

Shanghai Key Laboratory of Intelligent Information Processing

School of Computer Science, Fudan University

Abstract:

Image super-resolution, which is often regarded as a preprocessing procedure of scene text recognition, aims to recover the realistic features from a low-resolution text image. It has always been challenging due to large variations in text shapes, fonts, backgrounds, etc. However, most existing methods employ generic super-resolution frameworks to handle scene text images while ignoring text-specific properties such as text-level layouts and character-level details.

In this paper, we establish a text-focused super-resolution framework, called Scene Text Telescope (STT). In terms of text-level layouts, we propose a Transformer-Based SuperResolution Network (TBSRN) containing a Self-Attention Module to extract sequential information, which is robust to tackle the texts in arbitrary orientations. In terms of character-level details, we propose a Position-Aware Module and a Content-Aware Module to highlight the position and the content of each character. By observing that some characters look indistinguishable in low-resolution conditions, we use a weighted cross-entropy loss to tackle this problem. We conduct extensive experiments, including text recognition with pre-trained recognizers and image quality evaluation, on TextZoom and several scene text recognition benchmarks to assess the super-resolution images. The experimental results show that our STT can indeed generate text-focused super-resolution images and outperform the existing methods in terms of recognition accuracy

1. Introduction

Scene text recognition (STR) has drawn much research interest of the computer vision community due to its various applications such as license plate recognition and ID card recognition [17, 23, 39]. While STR has made a big step forward with the development of deep learning, recognition

performance on low-resolution (LR) text images is still subpar [44]. LR text images exist in many situations, e.g., a Figure 1. The proposed STT generates a relatively clearer text image and pays more attention to character details compared with

interpolation methods and generic SR methods. “SR” and “HR” denote super-resolution and high-resolution, respectively. photo taken with a low-focal camera or a document image compressed to reduce disk usages. When handling LR text images, existing recognition or spotting methods usually employ interpolation methods, like bicubic and bilinear interpolations, to upsample the original images [4, 27, 37, 38]. As shown in Figure 1, the image upsampled by the interpolation method is still blurred, which indeed brings difficulties to existing recognition models.

In recent years, several works employ generic superresolution methods for text image super-resolution. For example, in [8], SRCNN [7] with a shallow network is used as the backbone. In [42], a Laplacian-pyramid backbone is employed to combine features from several middle layers to upsample low-resolution images. However, these methods are not suitable for processing text images [44] since they see text images as general ones without taking text-specific properties (e.g. text-level layouts and character-level details) into consideration. In contrast, there are few methods that take a part of these properties into account. For example, PlugNet [30] designs a multi-task framework, aiming to recognize and upsample text images in one model. In [44], a Text Super-Resolution Network (TSRN) containing

a horizontal and a vertical BLSTMs [11] is proposed to capture sequential information of text images. However, the BLSTMs are not suitable for capturing sequential information of inclined or curved text images. In this paper, we propose a text-focused super-resolution framework, called Scene Text Telescope. To tackle texts in arbitrary orientations, we propose a novel backbone, namely Transformer-Based Super-Resolution Network (TBSRN) to capture sequential information. We notice that the previous methods usually employ loss functions

that focus on every pixel of the image, which may suffer great disturbances from backgrounds. According to Inattentional Blindness [28, 29], when humans observe a text image, they will naturally pay more attention to text regions rather than backgrounds, i.e., there is no need to improve the quality of the whole image in the super-resolution task. Based on this fact, we put forward a Position-Aware Module and a Content-Aware Module to focus on the position and the content of each character. By observing that there are some confusable characters in the low-resolution situation (e.g.

in Figure 1, "c" and "e" look similar), we employ a weighted cross-entropy loss in the Content-Aware Module to address this problem. Since these two modules are only used as text-specific guidance when training, they will not bring additional time overhead in the test stage. We mainly evaluate our method on TextZoom [44], which contains LR-HR pairs captured from digital cameras.

Furthermore, we conduct several experiments on scene text recognition benchmarks to further verify the capabilities of our STT as a preprocessor. In this work, we employ some widely used recognition models (e.g. ASTER [38], MORAN [26], and CRNN [37]) and image quality metrics, to evaluate the generated SR images. The experimental results show that the proposed STT can indeed generate text-focused super-resolution images and outperform existing methods in terms of recognition accuracy. Contributions of the proposed STT can be concluded in three-fold:

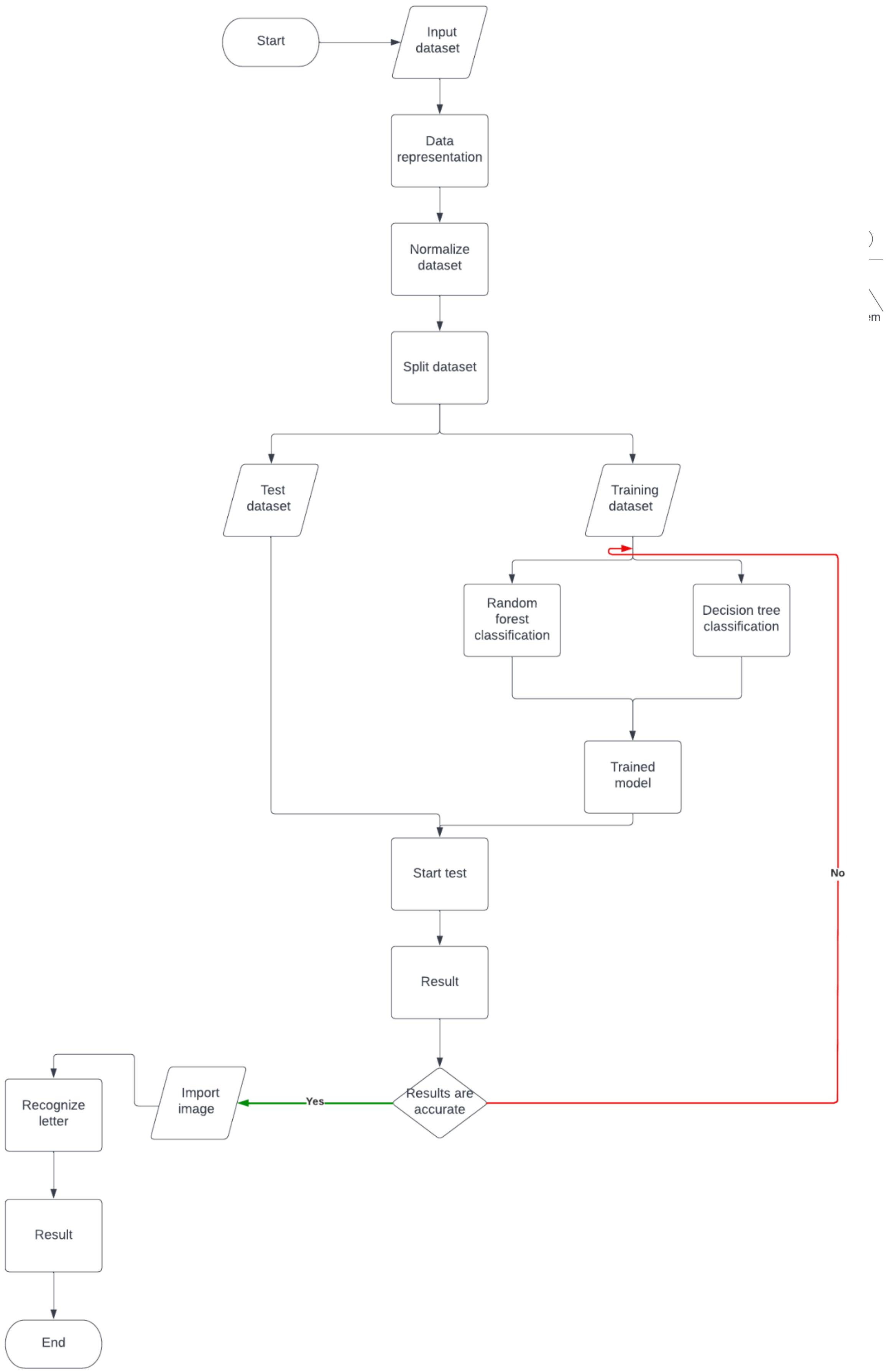
· We propose TBSRN to capture sequential information, which is more robust on texts in arbitrary orientations.

· A Position-Aware Module and a Content-Aware Module with a weighted cross-entropy loss are proposed to highlight the position and content of characters without bringing additional time overhead when testing.

· The proposed STT generates text-focused SR images and achieves higher recognition accuracy on pretrained recognizers than other existing methods.

5. Proposed Solution

These Diagrams (Use Case and Flow Chart) Shows The Processes and the user view we planned to implement. They briefly describe the way we implemented our code.


Start
Input dataset
Data representation
Normalize dataset
Split dataset
Test dataset
Training dataset
Random forest classification
Decision tree classification
Trained model
Start test
Result
Results are accurate
Yes
No
Import image
Recognize letter
Result
End

6. The Dataset employed

We decided to use a Kaggle large dataset to train the model in the most efficient way possible. The EMNIST dataset is a set of handwritten character digits derived from the NIST Special Database 19 and converted to a 28x28 pixel image format and dataset structure that directly matches the MNIST dataset.

Dataset link:
https://www.kaggle.com/datasets/crawford/emnist

Dataset Details:

There are six different splits provided in this dataset and each are provided in two formats:

1. Binary (see emnist*source*files.zip)
2. CSV (combined labels and images)
    - Each row is a separate image
    - 785 columns
    - First column = class_label (see mappings.txt for class label definitions)
    - Each column after represents one pixel value (784 total for a 28 x 28 image)

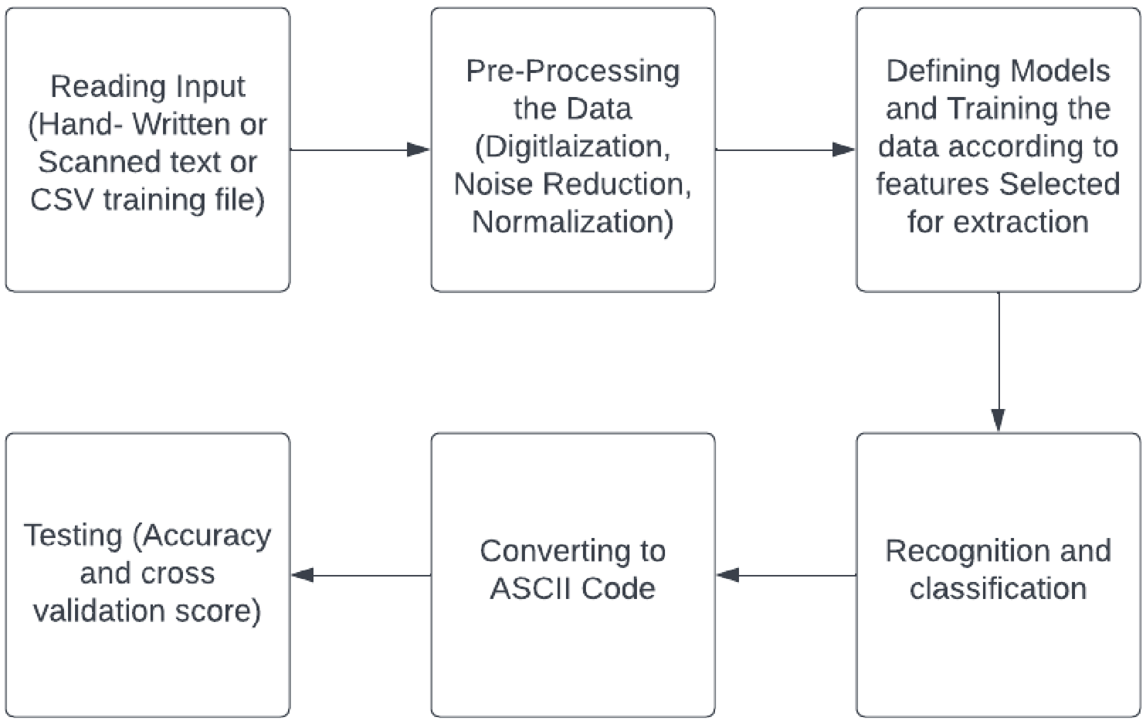
Letters dataset used:

Letters datasets

The EMNIST Letters dataset merges a balanced set of the uppercase and lowercase letters into a single 26-class task.

- train: 88,800
- total: 103,600
- classes: 37 (balanced)

7. Details of the algorithm(s)/approach(es) that will be



This Block Diagram simply describes the process of developing both algorithms.

Random Forest Algorithm:

Random Forest Algorithm is a supervised machine learning algorithm which is extremely popular and is used for Classification and Regression problems in Machine Learning. We know that a forest comprises numerous trees, and the more trees more it will be robust.
The following steps explain the working Random Forest Algorithm:

Step 1: Select random samples from a given data or training set.

Step 2: This algorithm will construct a decision tree for every training data.

Step 3: Voting will take place by averaging the decision tree.

Step 4: Finally, select the most voted prediction result as the final prediction result.

This combination of multiple models is called Ensemble. Ensemble uses two methods:

1. Bagging: Creating a different training subset from sample training data with replacement is called Bagging. The final output is based on majority voting.
2. Boosting: Combing weak learners into strong learners by creating sequential models such that the final model has the highest accuracy is called Boosting. Example: ADA BOOST, XG BOOST.

Decision Tree Algorithm:

A decision tree is a non-parametric supervised learning algorithm, which is utilized for both classification and regression tasks. It has a hierarchical, tree structure, which consists of a root node, branches, internal nodes and leaf nodes.

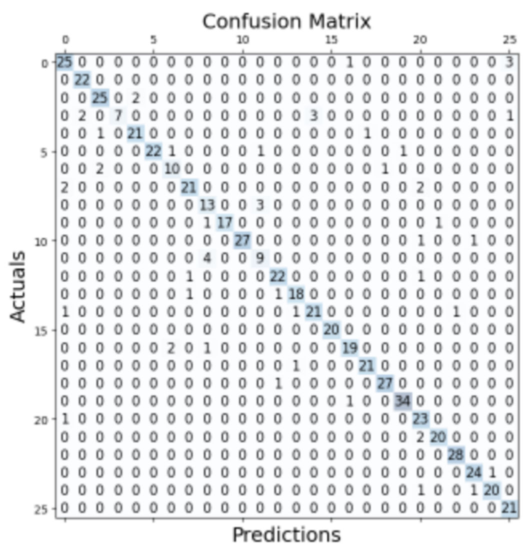
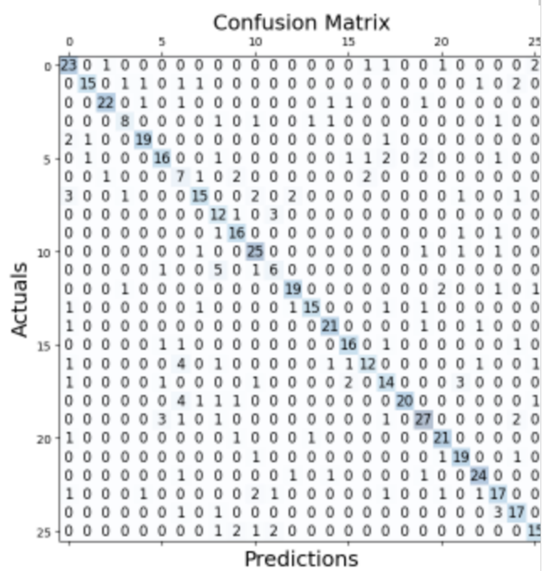
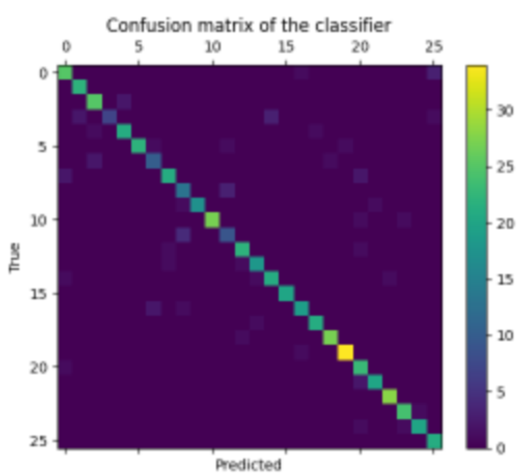
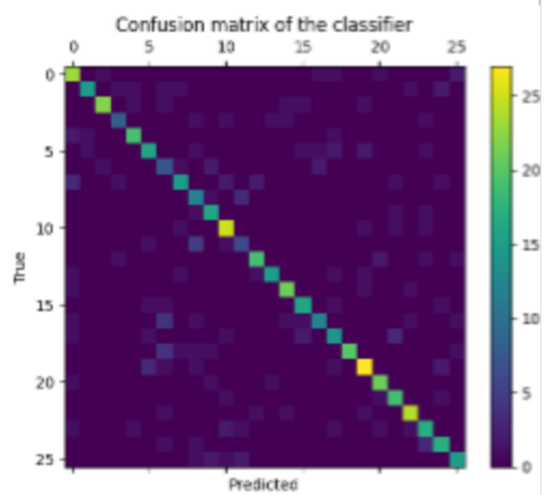
Decision tree learning employs a divide and conquer strategy by conducting a greedy search to identify the optimal split points within a tree. This process of splitting is then repeated in a top-down, recursive manner until all, or the majority of records have been classified under specific class labels. Whether or not all data points are classified as homogenous sets is largely dependent on the complexity of the decision tree. Smaller trees are more easily able to attain pure leaf nodes—i.e. data points in a single class. However, as a tree grows in size, it becomes increasingly difficult to maintain this purity, and it usually results in too little data falling within a given subtree. When this occurs, it is known as data fragmentation, and it can often lead to overfitting. As a result, decision trees have preference for small trees, which is consistent with the principle of parsimony

8. Details and development

Development platform: Visual Studio Code.
Programming Language: Python.
Version: python3
Training and testing: through K Fold Splitting technique
Libraries used:

- ☑ pandas
- ☑ numpy
- ☑ matplotlib.pyplot
- ☑ Sklearn.model_selection
- ☑ Sklearn.ensemble
- ☑ Sklearn.tree
- ☑ Sklearn.metrics
- ☑ Sklearn,preprocessing
- ☑ Sklearn.datasets
- ☑ Cv2
- ☑ Tkinter
- ☑ PIL

9. Experiments & Results

<table>
<tr><th>Algorithm</th><th>Random Forest</th><th>Decision Tree</th></tr>
<tr><td>Training Accuracy</td><td>100%</td><td>100%</td></tr>
<tr><td>Testing Accuracy</td><td>90.86%</td><td>74.62%</td></tr>
<tr><td>Cross Validation Score</td><td>88.50%</td><td>70.10%</td></tr>
<tr><td>Confusion Matrix</td><td>
</td><td>
</td></tr>
<tr><td>Confusion Matrix Classifier</td><td>
</td><td>
</td></tr>
</table>

10. Analysis, Discussion, and Future Work

Analysis of the results, what are the insights?

As clearly obvious in the data provided, random forests is a better algooritm and achieves better results in this particular field with such a large amount of dataset, as decision trees is not that efficient when it comes to large and complex data.

What are the advantages / disadvantages?

Advantages:

Based on the ensemble learning technique and creating sub-trees for every combination possible random forests reduces the overfitting problem for the decision trees. It also successfully recognizes characters with 90.86% accuracy.

Disadvantages:

The code unfortunately doesn't have a perfect prediction for paragraphs.

Why did the algorithm behave in such a way?

The main obstacle for decision trees is that it as a tree grows in size, it becomes increasingly difficult to maintain this purity, and it usually results in too little data falling within a given subtree. When this occurs, it is known as data fragmentation, and it can often lead to overfitting. As a result, decision trees have preference for small trees, which is consistent with the principle of parsimony

On the other way, random forests tend be slower when it deals with a larger set of data. That's why a more complex and deep algorithm could be better for producing better results.

What might be the future modifications you'd like to try when solving this problem?

it needs to be improved through either using a better more deep algorithm like Artificial Neural Networks, or using larger set of data with different set of features selected.

0. Resources

Idea and main functionalities

https://towardsdatascience.com/segmentation-in-ocr-10de176cf373

Similar Applications

https://theecmconsultant.com/best-ocr-software/

Literature Review

https://www.researchgate.net/publication/313334780_A_Study_on_Optical_Character_Recognition_Techniques

https://paperswithcode.com/paper/maskocr-text-recognition-with-masked-encoder

https://paperswithcode.com/paper/scene-text-telescope-text-focused-scene-image

Diagrams

https://lucid.app/lucidchart/91783ea7-49c9-4866-b42d-56d306d61902/edit?invitationId=inv_bbf5c303-7425-4e96-a3ac-cc5287bc1813

https://lucid.app/lucidchart/162104a6-211f-43d2-9347-3786d7f766c0/edit?viewport_loc=-435%2C41%2C2844%2C1151%2C0_0&invitationId=inv_c6555e49-b629-4b0b-8bc7-f87730588fe6

https://lucid.app/lucidchart/566305c7-3c7a-4db1-bf24-0fd010b008da/edit?viewport_loc=-1008%2C189%2C4206%2C1702%2C0_0&invitationId=inv_9aef859a-bd04-46b4-99d4-db9118efab81

Proposed Solution and useful resources

https://www.simplilearn.com/tutorials/machine-learning-tutorial/random-forest-algorithm#:~:text=A%20Random%20Forest%20Algorithm%20is,more%20it%20will%20be%20robust.

https://www.ibm.com/eg-en/topics/decision-trees#:~:text=A%20decision%20tree%20is%20a,internal%20nodes%20and%20leaf%20nodes.

https://towardsdatascience.com/a-short-introduction-to-model-selection-bb1bb9c73376

https://www.youtube.com/watch?v=gJo0uNL-5Qw&t=587s

https://www.youtube.com/watch?v=BnzUTg_nbpE

Additional Papers

https://acit2k.org/ACIT/images/stories/year2014/month1/ACIT2017_Proceeding/187_blinded.pdf

https://www.ijera.com/papers/Vol%201%20issue%204/BQ01417361739.pdf

Source Code

https://github.com/Yomna-Abdelghany/OCR-English-Letters-AIP-ID51-/blob/main/ocr.ipynb

GUI Implementation

https://github.com/Yomna-Abdelghany/OCR-English-Letters-AIP-ID51-/blob/main/OCR_GUI.py