OPTICAL CHARACTER RECOGNITION USING OPENCV AND TESSERACT

(IMAGE TO TEXT TO SPEECH)

SUBMITTED BY:

KARTHIK MADHETI: B190618CE

SAI TARUN UPPARI: B190610CS

UNDER THE GUIDANCE OF:

KISHAN MIRYALA

CHANDRALEKHA CHITTA

ABSTRACT:

Optical Character Recognition, shortly abbreviated as OCR, is the process of extracting text from an image. This project is nothing but a direct application of OCR on an image to extract text from it. This work is distinguished by four parts. The first part is obtaining the image. The second is the processing of the image using OpenCV. The third is passing the image to tesseract and obtaining the text from it. The final part is to read the text using pyttsx3. The programming language used for this project is python.

TABLE OF CONTENTS

1. INTRODUCTION

Optical Character Recognition, shortly abbreviated as OCR, is the process of extracting text from an image. Not only text from an image, but OCR is also a technology that enables you to convert different types of documents, such as scanned paper documents, PDF files, or images captured by a digital camera into editable and searchable data. However, there are several limitations of OCR that can result in an inaccurate or missing text which makes text-based classification difficult or impossible such as small font size of the text in documents, poor results with degraded documents, etc. so to increase the accuracy of ocr we use OpenCV to pre-process the images, or whatever they may be before performing ocr on them.

2. SYSTEM REQUIREMENTS:

Smartware requirements:

1. Python3
2. Opencv 4.2.0
3. Numpy 1.18.1
4. Pytesseract 0.3.4
5. Pyttsx3 2.87

Hardware requirements:

1. Mobilephone or a webcam with a resolution of at least 2MP or higher than that.

3. LITERATURE SURVEY:

3.1 What is **Computer Vision**?

Computer vision is a field of informatics, which teaches computers to see. It is a way the computer gathers and interprets visual information from the surrounding environment. Usually, the image is first processed on a lower level to enhance picture quality, for example removing noise and background (if necessary). Then the picture is processed on a higher level, for example detecting patterns and shapes, and thereby trying to determine, what is in the picture if the picture is being analyzed by machine learning. In our project, we do not have anything to do with computer vision except processing the image.

3.2 What is **OpenCV**?

Open-source computer vision, shortly abbreviated as Opencv is a library of programming functions mainly aimed at real-time computer vision. It mainly focuses on image processing, video capture, and analysis including features like face detection and object detection. We can do pretty much anything we want using OpenCV to the images and videos. **OpenCV**-**Python** is a library of **Python** bindings designed to solve computer vision problems using the python programming language.

3.3 What is **Numpy**?

NumPy, which stands for Numerical Python, is a library consisting of multidimensional array objects and a collection of functions for processing those arrays. Using NumPy, mathematical, and logical operations on arrays can be performed. As an image is digitally represented in the form of an n-dimensional array of pixel values, we need **NumPy** to process these arrays.

3.4 What is **OCR**?

Widely used as a form of data entry from printed paper data records – whether passport documents, invoices, bank statements, computerized receipts, business cards, mail, printouts of static-data, or any suitable documentation – It is a common method of digitizing printed texts so that they can be electronically searched, stored more compactly, displayed on line, and used in machine processes such as machine translation, text to speech, etc.

3.5 What is **Tesseract**?

Tesseract — is an optical character recognition engine with open-source code, this is the most popular and qualitative OCR-library. It was originally developed by Hewlett-Packard and is being developed by Google since 2005. Python-tesseract or pytesseract is an optical character recognition (OCR) tool for python.

4. CODE EXPLANATION:

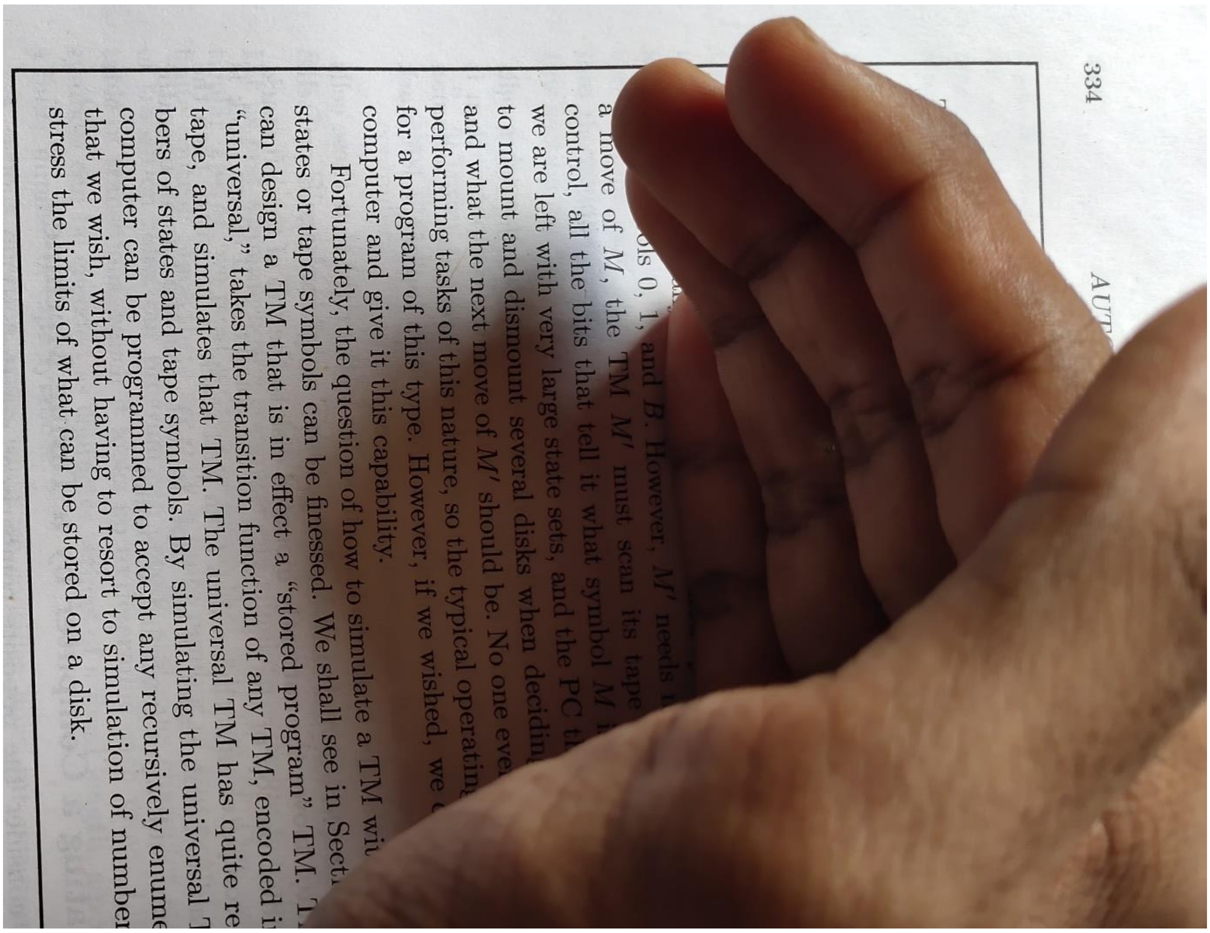
4.1 IMPORTING NECESSARY LIBRARIES

First, we import all the necessary libraries like OpenCV (cv2), NumPy, pytesseract, pyttsx3, and a few other libraries for computing.

4.2 READING THE IMAGE

Now we read the image using OpenCV. By default, OpenCV reads images in BGR format.

Input image:



334 AUT

ols 0, 1, and B. However, M′ needs
a move of M, the TM M′ must scan its tape
control, all the bits that tell it what symbol M
we are left with very large state sets, and the PC th
to mount and dismount several disks when decidin
and what the next move of M′ should be. No one eve
performing tasks of this nature, so the typical operatin
for a program of this type. However, if we wished, we c
computer and give it this capability.

Fortunately, the question of how to simulate a TM wit
states or tape symbols can be finessed. We shall see in Secti
can design a TM that is in effect a "stored program" TM. T
"universal," takes the transition function of any TM, encoded i
tape, and simulates that TM. The universal TM has quite re
bers of states and tape symbols. By simulating the universal T
computer can be programmed to accept any recursively enume
that we wish, without having to resort to simulation of number
stress the limits of what can be stored on a disk.

Now that we have read the image, we will work on basic background removal techniques to remove shadows from the image.

4.3 SHADOW REMOVAL:

Here, anything other than text is considered background. As we are going to work with a uniform threshold (will be explained later), we will have to process the image such that it contains uniform brightness all over the image. For this, we divide the original image with its background image. steps to be performed to remove shadows in an image are:

4.3.1 Split the image into individual planes and perform the operations on individual planes.

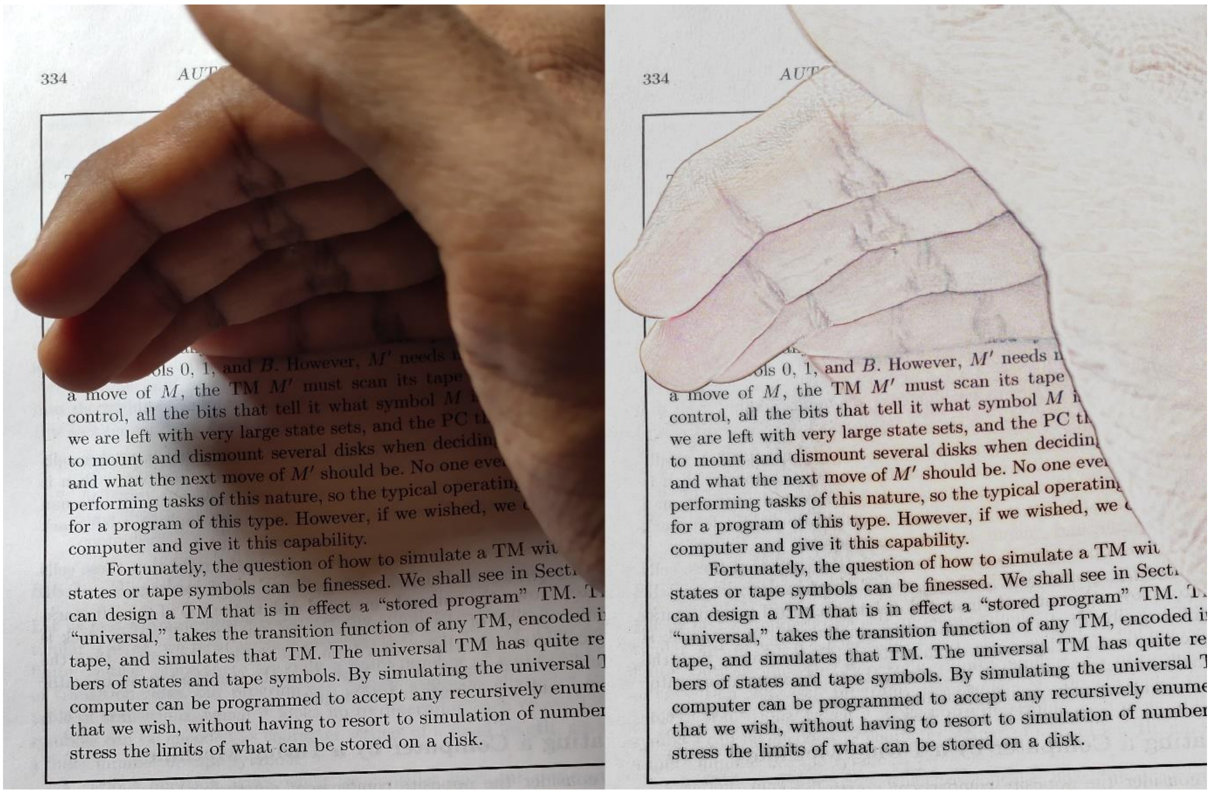
4.3.2 Perform a dilation process to spread the text to the background using a 7x7 kernel.

4.3.3 medianBlur the image with a kernel size of 21.

4.3.4 Merge the planes back. This is our background image.

4.3.5 Now divide the original image with the background image.

(i) Input image (ii) shadow-free image.



334 AUT

ols 0, 1, and B. However, M' needs
a move of M, the TM M' must scan its tape
control, all the bits that tell it what symbol M
we are left with very large state sets, and the PC t
to mount and dismount several disks when decidin
and what the next move of M' should be. No one eve
performing tasks of this nature, so the typical operatin
for a program of this type. However, if we wished, we
computer and give it this capability.
Fortunately, the question of how to simulate a TM wi
states or tape symbols can be finessed. We shall see in Sect
can design a TM that is in effect a “stored program” TM. T
“universal,” takes the transition function of any TM, encoded i
tape, and simulates that TM. The universal TM has quite re
bers of states and tape symbols. By simulating the universal T
computer can be programmed to accept any recursively enume
that we wish, without having to resort to simulation of number
stress the limits of what can be stored on a disk.

For the sake of clarity, we have performed the shadow removal techniques on a skew-corrected image. As you can see our output image does not contain any shadows and has a uniform brightness. This method is nothing but divide-blending the image-background and the image.

We have used the **blend_modes** library to divide the images just to avoid errors when other functions are used, and it is simple to use. The blend mode functions expect **NumPy float** arrays in the format as *[pixels in dimension 1, pixels in dimension 2,4]* an input. Both images need to have the same shape, where '4' represents the number of channels. The order of the channels should be *R, G, B, A*, where *A* is the alpha channel. All values should be *floats* in the range *0.0 <= value <= 255.0*. So an additional alpha channel is to be added to each of the two input images of the divide function.

Now that we have removed the shadows in the image, we will try to correct the skew in the images if present.

4.4 CORRECTING THE 2D SKEW:

Here comes the most interesting part of the project, correcting the skew of the image. Our project works best only with the images containing 2-dimensional skew only. We have not extended it to correct the 3-dimensional skew in images.

To correct the skew we have to determine the angle at which the text in the image is inclined. And for that, we make use of OpenCV's minAreaRect() function. The last output of the function is the angle at which the contour is inclined with horizontal. And for that, we have written a function ***get_median_angle().*** Before we find the angle, we have to know the characteristics of the image i.e. whether the image has white text on a dark background or dark text on a white background. So there is a need to get the correctly binarized image i.e. binarized image with dark text on white background.

4.4.1 Resizing without distortion:

As we are going to work with morphological transformations that fit for an image with dimensions 2000x3000 with 2000 being the width and 3000 the height of the image, we have to resize the image to that dimensions without any distortions, in simple words **resizing with ratios**.

4.4.2 Getting the correctly binarized image:

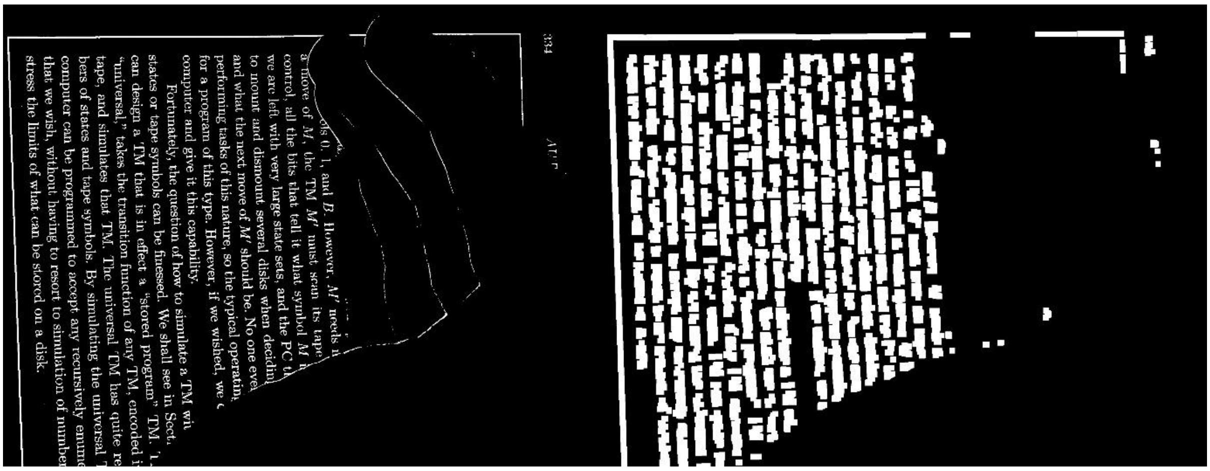
The image we pass to tesseract may contain white text on a dark background or vice versa. But tesseract only works with images of dark text on a white background. So there is a need to check the properties of the text and the background before passing it to tesseract. And for that, we use a naïve idea of calculating the number of contours in the binarized and inverted binarized image. the naïve idea is that a binarized image with a dark background has more number of contours than a binarized image with a white background. And we return the correctly binarized image with those conditions.

Here we used Otsu's binarization method which is a uniform threshold method. That is the reason we had to set the brightness of the image to a uniform level while removing shadows.

4.4.3 Determining the angle:

To determine the angle, we start by eliminating the noise in the inverted binarized image to a maximum extent by performing morphological operations on the binarized image. After performing the morphological operations we get the image with all the connected components. Then we find contours of those words (connected components) and use *minAreaRect()* function on each contour to get their orientation. We collect all the angles and calculate the median of those angles. The reason we find the median rather than considering the mean is that the median lies in the middle. When we want to use mean of the angles, we even consider the unwanted contours which we could not eliminate using those morphological operations. And this alters the angle resulting in the incorrect determination of skew. The outputs of the important steps are as follows:

(i) inverted-binary image. (ii) connected components image.



4.4.4 Correcting the angle:

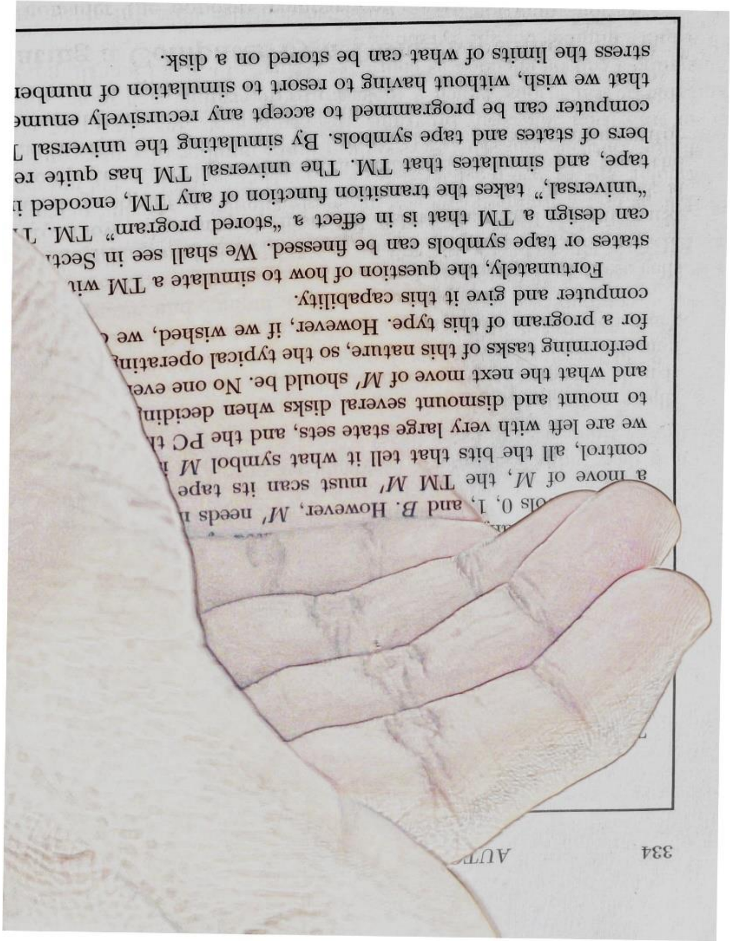
Now that we have determined the angle at which the text is oriented, we have to correct the angle before passing it to the *warpAffine()* function such that it aligns with any one of the axes.

The conditions to correct the angle are:

```
if 0 <= angle <= 90:
        corrected_angle = angle - 90
elif -45 <= angle < 0:
        corrected_angle = angle - 90
elif -90 <= angle < -45:
        corrected_angle = 90 + angle
```

Now we pass the corrected angle to the warpAffine() function to rotate our image. The output after rotating the image is as follows:

(i) rotated image:

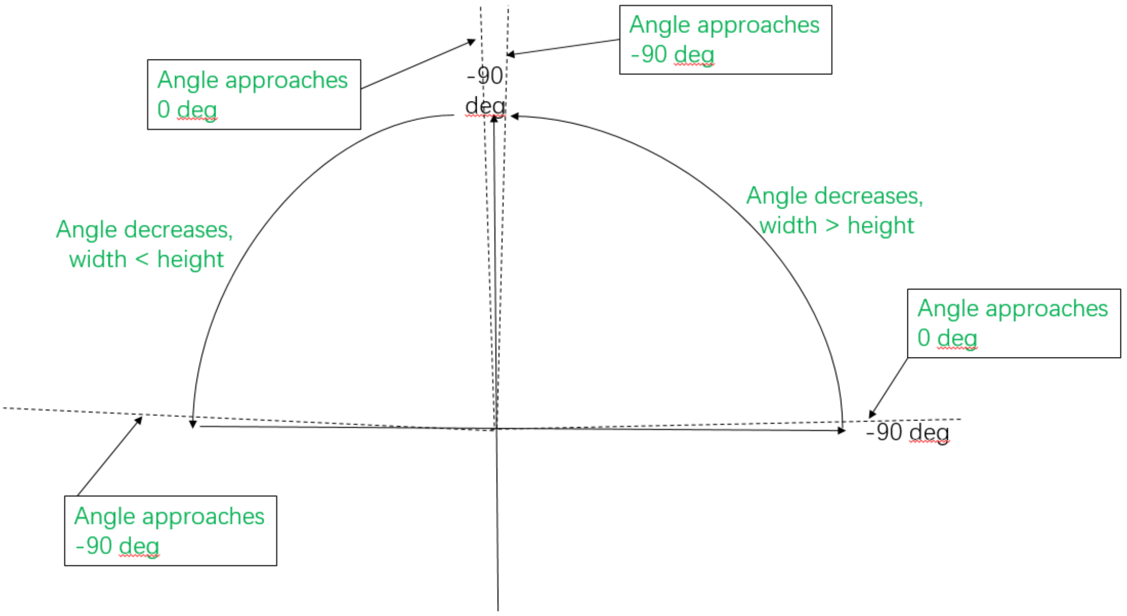


334 AUT

ols 0, 1, and B. However, M' needs n
a move of M, the TM M' must scan its tape
control, all the bits that tell it what symbol M i
we are left with very large state sets, and the PC th
to mount and dismount several disks when decidin
and what the next move of M' should be. No one eve
performing tasks of this nature, so the typical operatin
for a program of this type. However, if we wished, we c
computer and give it this capability.

Fortunately, the question of how to simulate a TM wi
states or tape symbols can be finessed. We shall see in Secti
can design a TM that is in effect a "stored program" TM. T
"universal," takes the transition function of any TM, encoded i
tape, and simulates that TM. The universal TM has quite re
bers of states and tape symbols. By simulating the universal T
computer can be programmed to accept any recursively enume
that we wish, without having to resort to simulation of number
stress the limits of what can be stored on a disk.

As you can see, the image is not rotated correctly. This is because of the ambiguous nature of the minAreaRect() function.



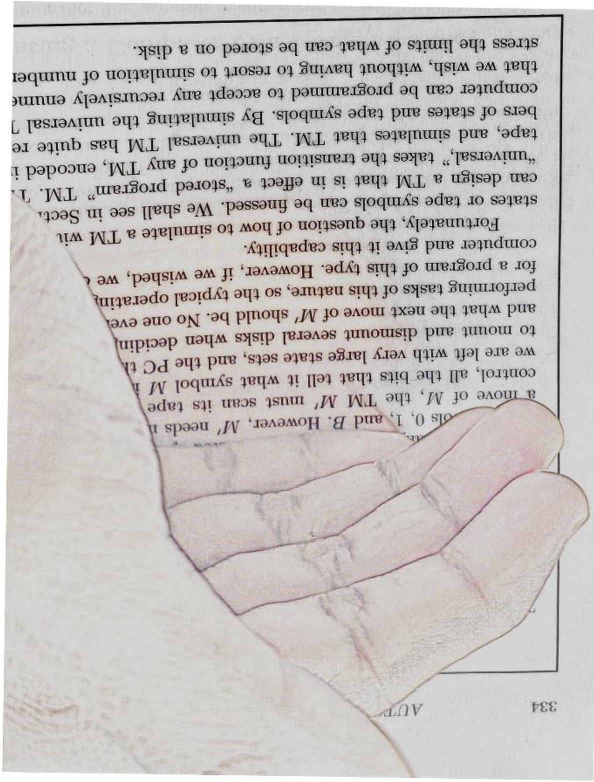
The above picture demonstrates the determination of the angle by the minAreaRect() function. Even though our text is oriented to 270 (+90 from 0) degrees, it determines the angle as -90 degrees approx because it is just calculating the orientation of the rotated rectangle.

To solve this issue we have come up with the pytesseract's *image_to_osd()* function which outputs the information about the orientation and script detection of the text in the image.

We pass the binarized image of the rotated image to the *image_to_osd()* function then find the correct orientation of the text. In our case, it is 180 degrees. We run a while loop till it rotates the image to 0 degrees.

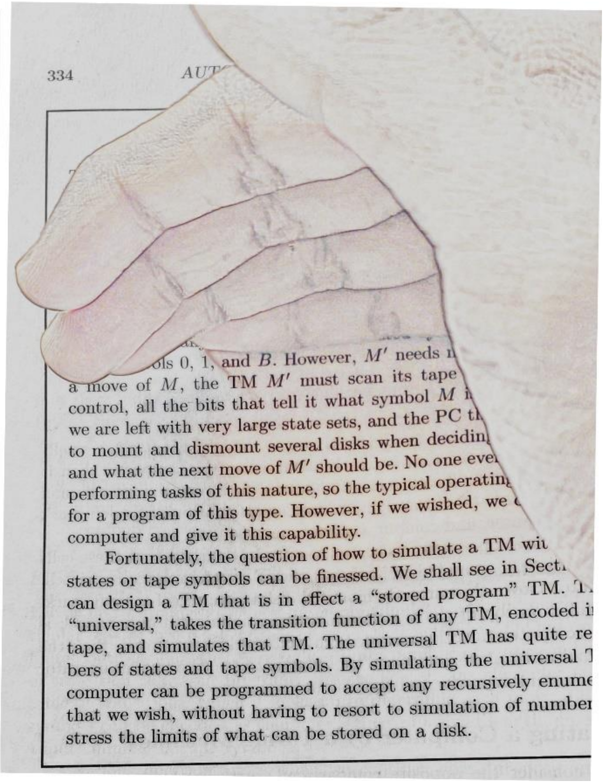
The reason we pass the binarized image is that Tesseract works best with images containing more visible characters. Otherwise, due to less clarity in the image, tesseract raises an exception: **TesseractError** or the 0 dpi exception.

(i) rotated image (ii) rotated image after using *image_to_osd()*



ols 0, 1, and B. However, M′ needs n
a move of M, the TM M′ must scan its tape
control, all the bits that tell it what symbol M i
we are left with very large state sets, and the PC t
to mount and dismount several disks when decidin
and what the next move of M′ should be. No one eve
performing tasks of this nature, so the typical operatin
for a program of this type. However, if we wished, we c
computer and give it this capability.
Fortunately, the question of how to simulate a TM wi
states or tape symbols can be finessed. We shall see in Sect
can design a TM that is in effect a "stored program" TM. T
"universal," takes the transition function of any TM, encoded i
tape, and simulates that TM. The universal TM has quite re
bers of states and tape symbols. By simulating the universal T
computer can be programmed to accept any recursively enume
that we wish, without having to resort to simulation of numbe
stress the limits of what can be stored on a disk.

334 AUT



334 AUT

ols 0, 1, and B. However, M' needs n
a move of M, the TM M' must scan its tape
control, all the bits that tell it what symbol M i
we are left with very large state sets, and the PC tl
to mount and dismount several disks when decidin
and what the next move of M' should be. No one eve
performing tasks of this nature, so the typical operatin
for a program of this type. However, if we wished, we c
computer and give it this capability.

Fortunately, the question of how to simulate a TM wi
states or tape symbols can be finessed. We shall see in Sect
can design a TM that is in effect a "stored program" TM. T
"universal," takes the transition function of any TM, encoded i
tape, and simulates that TM. The universal TM has quite re
bers of states and tape symbols. By simulating the universal T
computer can be programmed to accept any recursively enume
that we wish, without having to resort to simulation of numbe
stress the limits of what can be stored on a disk.

Now that we have the skew-corrected image, we resize this image to our original image dimensions to preserve clarity. Now we binarize the image and pas it to pytesseract's ***image_to_data()*** function. Returns result containing box boundaries, confidences, text, and other information.

Now we collect the texts that have a confidence value greater than 64 (confidence threshold value).

OCR confidence is a numerical value assigned by an OCR engine to a text component, indicating the degree to which the engine is certain that it has recognized the component correctly.

4.5 DRAWING THE BOUNDING BOXES:

From the output of the ***image_to_data()*** function, we have the coordinates of the bounding boxes of texts. Now we draw bounding boxes around the text and collect the texts into an empty list.

4.6 READING THE TEXT:

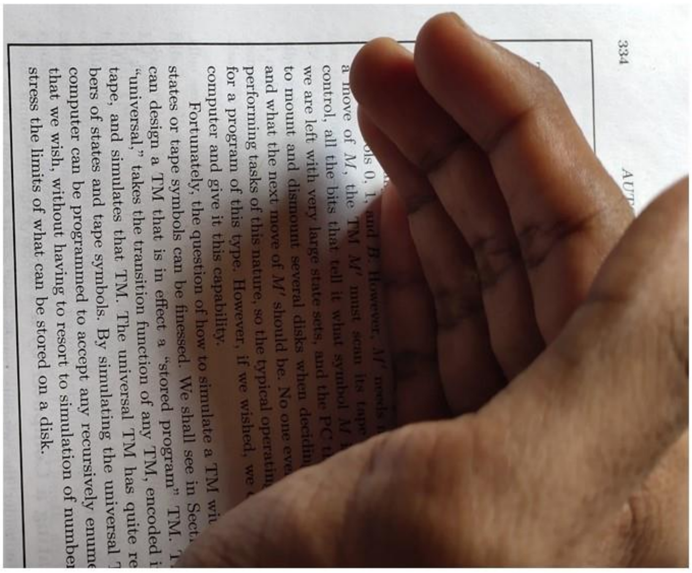
Join the elements of that text list and pass it to pyttsx3.init().say(string) function.

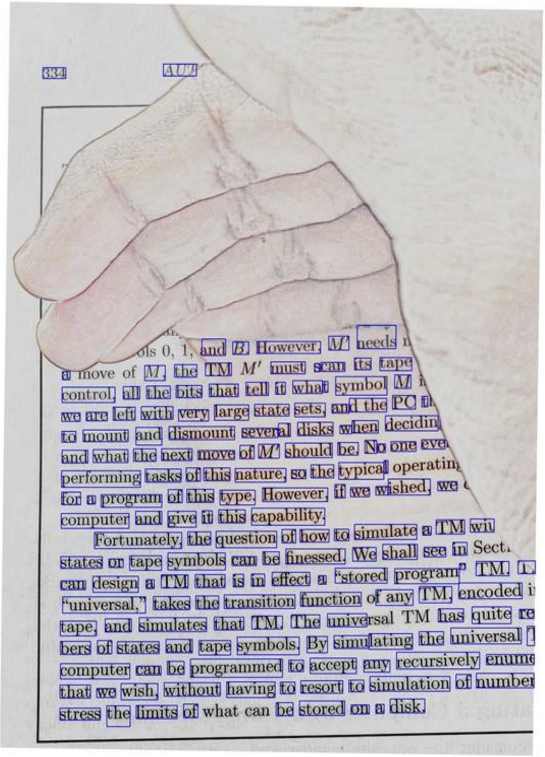
For tesseract, we have set the ocr-engine-mode(oem) and page-segmentation-mode(psm) to 1 and 12 respectively.

5. RESULTS:

1.

(i) input image (ii) output image

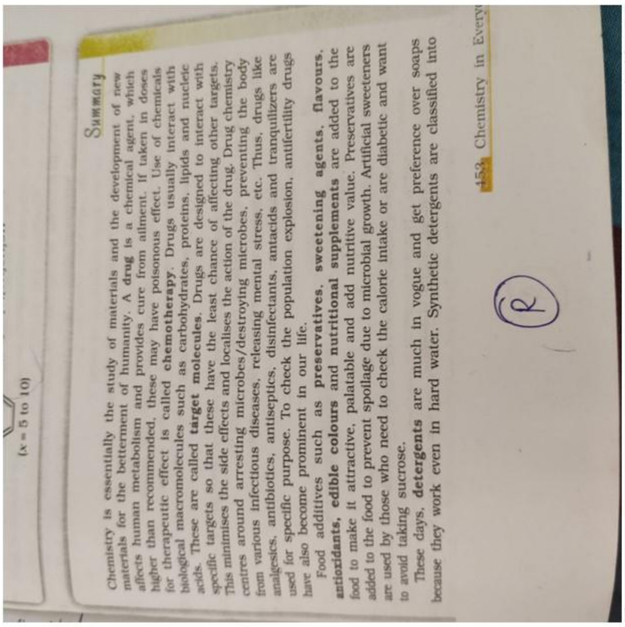


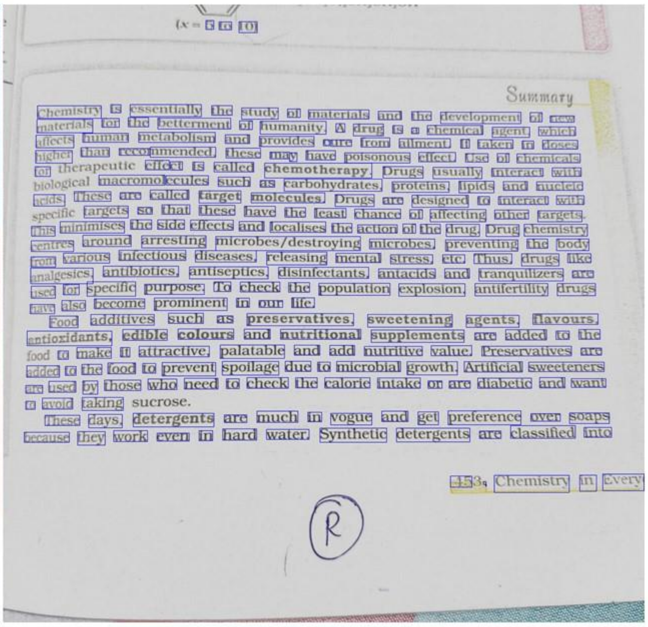


TEXT:

334 AUT and B. However, M’ needs can its tape a M, the TM must s control, all the bits that tell it wha t symbol M dthe PC t we are left with very large state sets, an hen decidin to mount and dismount sever al disks w and what the next move of M’ should be. N o one eve 1 performing tasks of this nature, so the typica for a program of this type. However, if we W ished, we computer and give it this capability. Fortunately, the question of how to simulate a TM wit We shall see in states or tape symbols can be finessed. » TM. ‘L. can design a TM that is in effect a “stored program fany TM, encoded “universal,” takes the transition function has quite re tape, and simulates that TM. The unive lating the universal ‘7 bers of states and tape symbols. By simu computer can be programmed to accept any recursively enume that we wish, without having to resort to simulation of number stress the limits of be stored on a disk.

2.





(x = 5 to 10)

Summary

Chemistry is essentially the study of materials and the development of new materials for the betterment of humanity. A **drug** is a chemical agent, which affects human metabolism and provides cure from ailment. If taken in doses higher than recommended, these may have poisonous effect. Use of chemicals for therapeutic effect is called **chemotherapy**. Drugs usually interact with biological macromolecules such as carbohydrates, proteins, lipids and nucleic acids. These are called **target molecules**. Drugs are designed to interact with specific targets so that these have the least chance of affecting other targets. This minimises the side effects and localises the action of the drug. Drug chemistry centres around arresting microbes/destroying microbes, preventing the body from various infectious diseases, releasing mental stress, etc. Thus, drugs like analgesics, antibiotics, antiseptics, disinfectants, antacids and tranquilizers are used for specific purpose. To check the population explosion, antifertility drugs have also become prominent in our life.

Food additives such as **preservatives**, **sweetening agents**, **flavours**, **antioxidants**, **edible colours** and **nutritional supplements** are added to the food to make it attractive, palatable and add nutritive value. Preservatives are added to the food to prevent spoilage due to microbial growth. Artificial sweeteners are used by those who need to check the calorie intake or are diabetic and want to avoid taking sucrose.

These days, **detergents** are much in vogue and get preference over soaps because they work even in hard water. Synthetic detergents are classified into

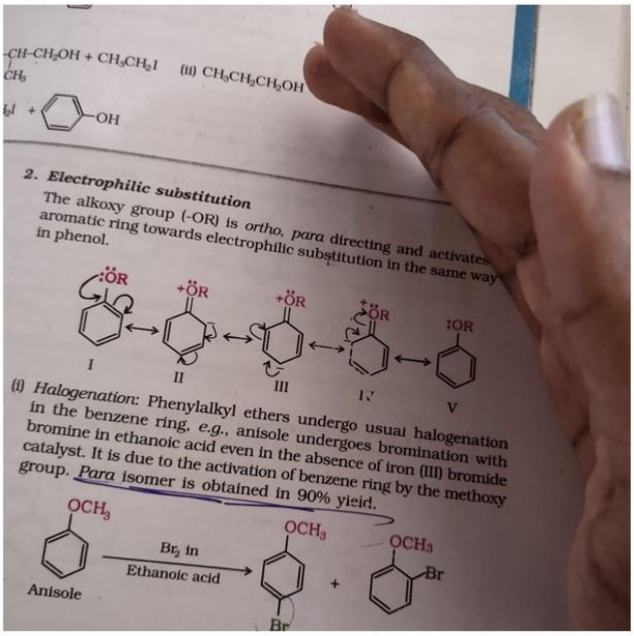
453 Chemistry in Every

R

TEXT:

5 to 10) Chemistry is cssentially the study of materials and the development of new materials for the betterment of humanity. A drug is a chemical agent. which affects human metabolism and provides c ure from ailment. If taken in doses mmended, these ma higher than recor y have poisonous effect. Use of chemicals for effe ct is called chemotherapy. Drugs usually interact with macromol ecules such as carbohydrates, proteins. lipids and nucleic acids. These are called target molecules. Drugs are designed to interact with targets so that these have the least chance of affecting other targets This minimises the side effects and localises the action of the drug. Drug chemistry centres around arresting microbes/destroying microbes, preventing the body from various infectious diseases, releasing mental stress, etc. Thus, drugs like analgesics. antibiotics, antiseptics, disinfectants, antacids and tranquilizers are used for specific purpose. To check the population explosion, antifertility drugs have also become prominent in our life. Food additives such as preservatives, sweetening agents, flavours, entioxidants, edible colours and nutritional supplements are added to the to make it attractive, palatable and add nutritive value. Preservatives are added to the food to prevent spoilage due to microbial growth. Artificial sweeteners are used by those who need to check the calorie intake or are diabetic and want io avoid taking These days, detergents are much in vogue and get preference over soaps because they work even in hard water. Synthetic detergents are classified into 45 “4 4 Chemistry in Every!

3.



-CH-CH$_2$OH + CH$_3$CH$_2$I (ii) CH$_3$CH$_2$CH$_2$OH
CH$_3$

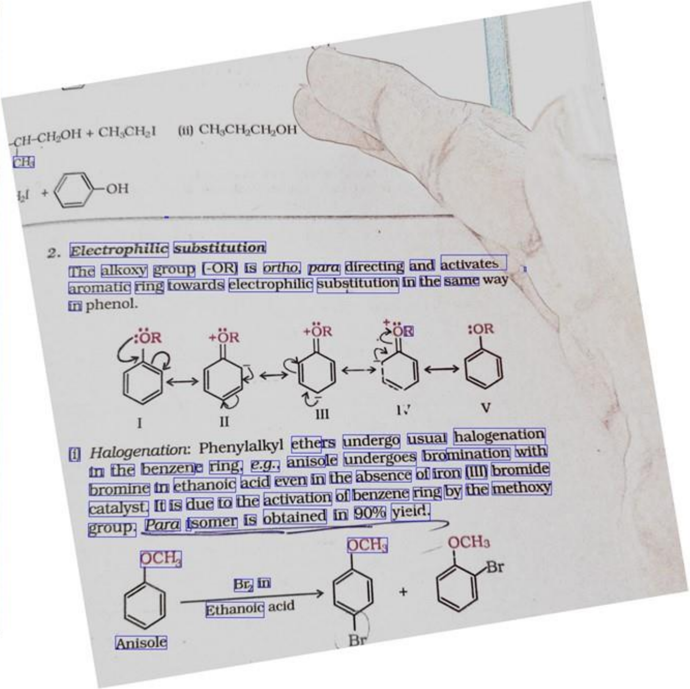
I + ⌬-OH

2. Electrophilic substitution

The alkoxy group (-OR) is *ortho, para* directing and activates aromatic ring towards electrophilic substitution in the same way in phenol.

:ÖR
+ÖR
+ÖR
+ÖR
:OR
I
II
III
IV
V

(i) *Halogenation:* Phenylalkyl ethers undergo usual halogenation in the benzene ring, *e.g.*, anisole undergoes bromination with bromine in ethanoic acid even in the absence of iron (III) bromide catalyst. It is due to the activation of benzene ring by the methoxy group. *Para* isomer is obtained in 90% yield.



-CH-CH$_2$OH + CH$_3$CH$_2$I (ii) CH$_3$CH$_2$CH$_2$OH
CH$_3$

I + ⌬-OH

2. Electrophilic substitution

The alkoxy group (-OR) is ortho, para directing and activates aromatic ring towards electrophilic substitution in the same way in phenol.

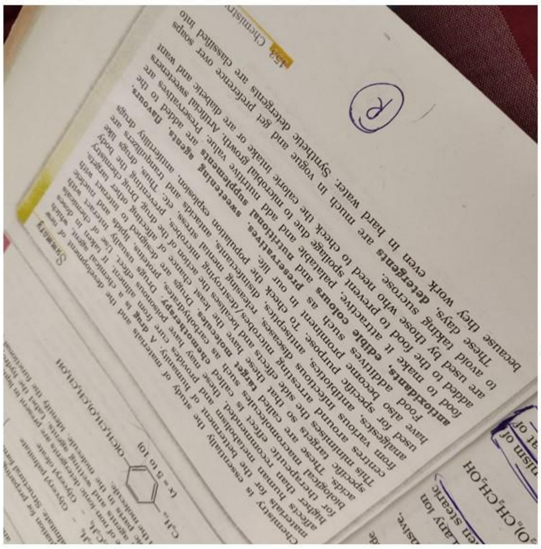
(i) Halogenation: Phenylalkyl ethers undergo usual halogenation in the benzene ring, e.g., anisole undergoes bromination with bromine in ethanoic acid even in the absence of iron (III) bromide catalyst. It is due to the activation of benzene ring by the methoxy group. Para isomer is obtained in 90% yield.

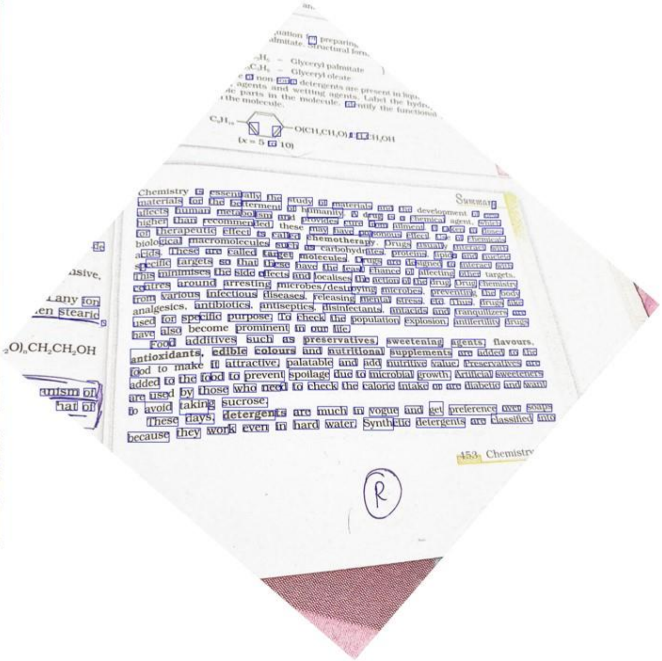
OCH$_3$ Anisole Br$_2$ in Ethanoic acid OCH$_3$ Br + OCH$_3$ Br

TEXT:

CHs ~ Electrophilic substitution The alkoxy group (-OR) is ortho, para directing and activates \ aromatic ring towards electrophilic substitution in the same in R ethe rs undergo usual halogenation (i) le undergoes bro mination with in the benzen e ring, e.g., aniso the absence of iron (Ill) bromide bromine in ethanoic acid even in the activation of benzene ring by the methoxy catalyst. It is due to isomer is obtained in 90% yieid. group. Para } OCH; OCH, Br, in Ethanoic Anisole

4.

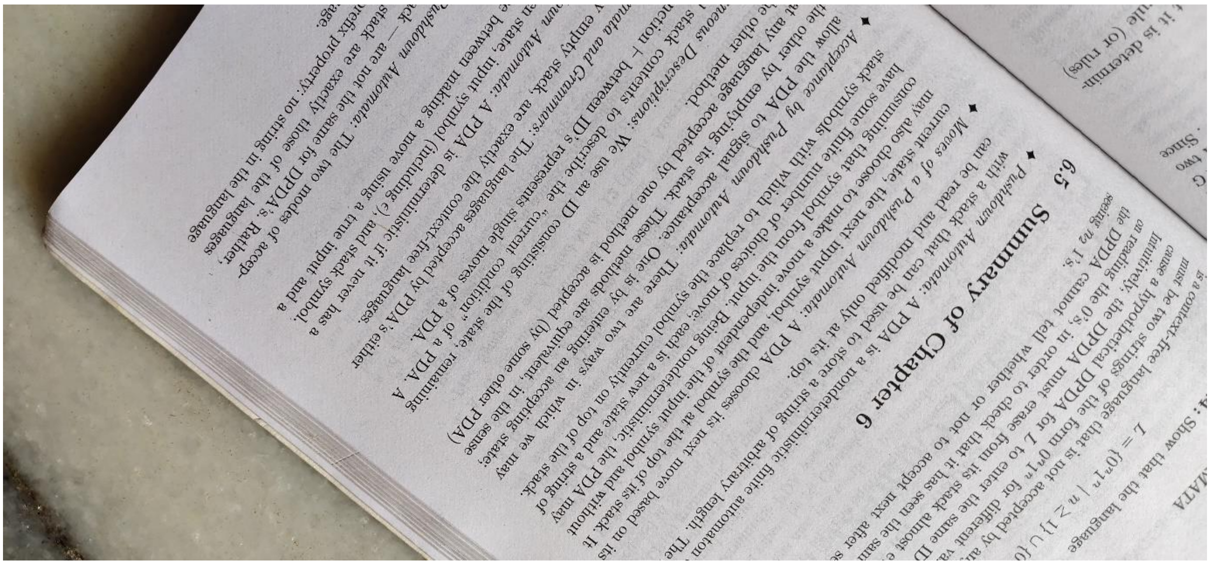


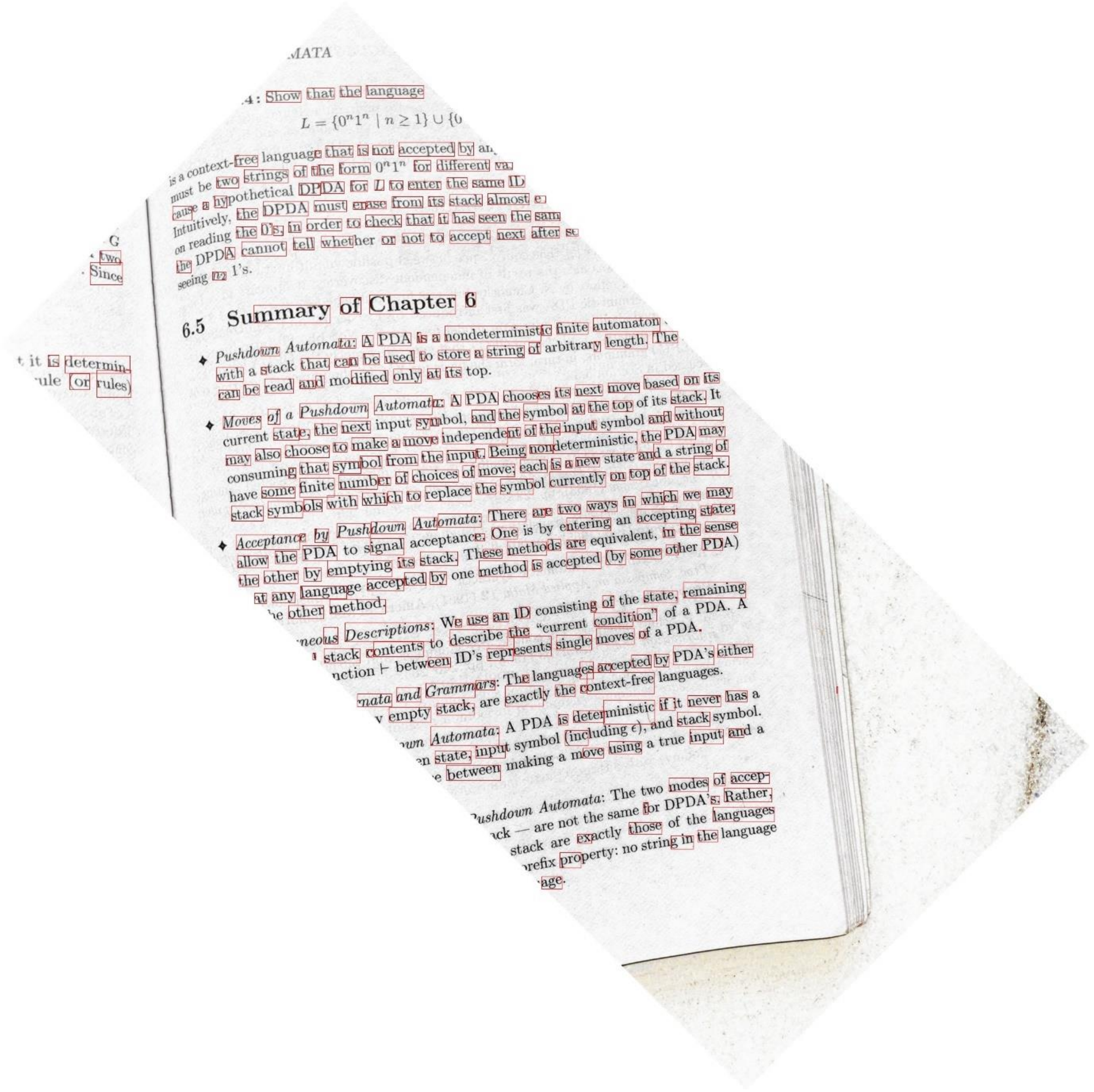


TEXT:

Or Of for fe Vete \ 4 ¢ the to is essent ially the materials for the be tterment of Study of Materials 4 affects human meta bol umanity A and the of new ism an Tug is a chemica) higher than recommen ded, Provides cure fr which may have poi om ailment. if te Ken in doses fortherapeutic effect is call Sonous effect. Us of chemicals gical macromolecules su ed chemotherap Drugs usualt de ch as imteract with cids. These are called tar get molecules. 5 ates, proteins lipid S and nucleic Tugs are de Signed to interact with p ecific targets so that th ese have the leas t chance of This minimises the side e ffects and localises th: affecting other € action of the drug Drug chemistry ce ntres around arresting microbes/destr oying microbes preventin the body ion from various infectious diseases antibiotics releasing mental stress etc. Thus drugs like antiseptics disinfectants Stearic cific purpose. To check the antacids and tranquilizers are S used for spe population explosion, antifertility drugs also prominent in our life. have d additives such as preservatives, sweetening agents, Foo edible colours and nutritional supplements are added to the antioxidants it attractive, palatable and add nutritive value. Preservatives are fo the fo od to prevent spoilage due to microbial growth. Artificial sweeteners added to d to check the calorie intake or are diabetic and want anism of those who nee are used by hat of g sucrose. t avoid takin ts are much in vogue and get preference over soaps These days. detergen etic detergents are classified into k even in hard water. Synth: pecause they WOF

5.





MATA

.4: Show that the language

$$L = \{0^n1^n \mid n \ge 1\} \cup \{0$$

is a context-free language that is not accepted by an
must be two strings of the form 0^n1^n for different va
cause a hypothetical DPDA for L to enter the same ID
Intuitively, the DPDA must erase from its stack almost e
on reading the 0's, in order to check that it has seen the sam
the DPDA cannot tell whether or not to accept next after se
seeing n_2 1's.

G
two
Since

t it is determin-
rule (or rules)

6.5 Summary of Chapter 6

- ✦ *Pushdown Automata*: A PDA is a nondeterministic finite automaton
with a stack that can be used to store a string of arbitrary length. The
can be read and modified only at its top.

- ✦ *Moves of a Pushdown Automata*: A PDA chooses its next move based on its current state, the next input symbol, and the symbol at the top of its stack. It may also choose to make a move independent of the input symbol and without consuming that symbol from the input. Being nondeterministic, the PDA may have some finite number of choices of move; each is a new state and a string of stack symbols with which to replace the symbol currently on top of the stack.

- ✦ *Acceptance by Pushdown Automata*: There are two ways in which we may allow the PDA to signal acceptance. One is by entering an accepting state; the other by emptying its stack. These methods are equivalent, in the sense
at any language accepted by one method is accepted (by some other PDA)
he other method.

neous Descriptions: We use an ID consisting of the state, remaining
stack contents to describe the "current condition" of a PDA. A
nction ⊢ between ID's represents single moves of a PDA.

mata and Grammars: The languages accepted by PDA's either
y empty stack, are exactly the context-free languages.

wn Automata: A PDA is deterministic if it never has a
en state, input symbol (including ϵ), and stack symbol.
e between making a move using a true input and a

ushdown Automata: The two modes of accep-
ck — are not the same for DPDA's. Rather,
stack are exactly those of the languages
refix property: no string in the language
age.

TEXT:

Show that the language free e that is not accepted by he form for different va. two strings of t DA for L to enter the same TD DP a hy DPDA must er ase from its stack almost e. caus the check that it has seen the sam the 0" s, in order to to accept next after st or not two A cannot tell whet Since abe ng mmery of Chapter 6 s a nondeterminist ic finite automaton an ta: A PDA i y length. The is determin. n be used t store a string of with § that Ca! dified only at its (or rules) d pased on its can be yead an a: A PDA choos eg its next move stack. Pushdown Automat the symbol at the top d without Moves of w and 4 an stati e, the next sy e independe nt of the the PDA may also choose to make a mov 4. Being non deterministic, dastring of may bol from the inpu each is a new an g that sym er of choices of move} of the stack. finite numb the symb' currently on top some ols with whi ch to replace whi ch we may stack symb There ar e two ways in accepting st ate; down Aut omata: ntering in the sense Acceptanc e by Push e. One ds are PDA ignal ese metho (by some ot PD. allow the emptying i ts stack. epted the other by ced by method js qt any langue accep’ remaining g of state, other method. fe use an D condit! on” the us Descriptions: describe. moves of . 1 gtack ¢ ontents een repr esents single es ac ted bY PDA’s either ars e ontext-free exacth the © = mata and Gramm never has empty stack, js deter ministic ifi stack Automata: (ine using input an state, inpw Ove petween modes of accep- 5. Rather, f languages ex those gin the pro age

6. POSSIBLE IMPROVEMENTS AND LIMITATIONS:

6.1 Spell-checker:

As you can read the results, you might have noticed that there are a few spelling mistakes in the text. We can correct these spelling mistakes using the ***pyspellchecker*** library.

6.2 Correcting the 3d tilt:

As you can see, the 4th and 5th results are not very accurate. This is because tesseract fails to work effectively when there is a skew in the image, be it 2-dimensional or 3-dimensional. We were successful in correcting the 2d skew in the images. But if we can correct even the minute 3d skew also, then there is a chance to increase the accuracy of the ocr.

6.3 Making the text lines horizontal (effective 2d deskewing):

If you take a look at the pictures carefully, you might notice the incorrect recognition of some words. This is more like breaking the word into pieces even if the word is perfect to be read by tesseract. E.g. “macromolecules” is broken down to “macromol” and “ecules”. We think that we can try to eliminate these kinds of errors by making the lines of text horizontal (do not get confused, it is not the skew in the image). Refer to result no. 5.

6.4 Images with multiple skews:

We can even have images containing multiple skews to be read by ocr. But we haven’t considered such kind of images in our project.

6.5 Images with multiple backgrounds:

Our code fails with images containing multiple backgrounds, colored (single or multi) backgrounds. E.g. page covers, images with less text, images with white text on a dark background.

7. CONCLUSION:

We were able to pre-process the image using OpenCV to a maximum extent. And extract text from it using tesseract and read it aloud using pyttsx3.

8. REFERENCES:

www.docs.opencv.org

www.stackoverflow.com

www.github.com

www.pypi.org