Applying Generative AI to Textbook Summarization in an Assistive Technology Setting

Elizabeth M. Manuel

Received September 13, 2024
Accepted August 06, 2025
Electronic access August 31, 2025

Computer vision has been used in the past to make text-based resources and scene text more accessible to the visually impaired. Readily available solutions don't presently exist for students who don't have access to textbooks in Braille, thus constraining their studies. This paper aims to establish how computer vision and Artificial Intelligence (AI) can be harnessed to make print resources, such as textbooks, more accessible to visually impaired students, by proposing a solution that involves the use of a vision-language model (VLM) to extract text from textbook pages, when images of them are clicked. This paper consists of a literature review and an analysis of the accuracy of vision-language models in detecting text from textbook pages. The outcomes of this paper will help develop solutions that can be implemented in smaller schools and homes for blind students, to aid in the process of their education, using Claude, the VLM that performed better in the experiment conducted. This paper aims to find viable, accessible solutions to replace a sighted individual, and provide for a more independent study process for students with disabilities.

Introduction

Background and Context

Visually-impaired students in smaller towns and cities, that don't have access to facilities, training and resources to adapt to the mainstream methods of teaching, are cut off from a world of possibilities, due to their disability. Research in the past has been directed towards finding assistive technology for the visually impaired, that caters to more general scenarios. More specifically, in the field of computer vision, it has been to recognize characters in more widely circulated print-based resources, such as newspapers[1] or books in a library[2], and in signs and other scene text[3–5]. OCR (Optical Character Recognition)(see Glossary) was used to convert more than 16 million newspaper pages to a digital form for Chronicling America[1]. It used a fine-tuned Faster-RCNN (Region-based Convolutional Neural Network)(see Glossary) model and was initially tested on World War 1-era Chronicling America pages.

One solution Adaptive Text Region Representation[3] proposes a two-step process: text proposal and proposal refinement. In the first step, a Text-RPN (Region Proposal Network which proposes all identifiable objects within the boundaries of an image[6] generates a text proposal, given the input image. The proposal is refined through a refinement network. At the same time, text/non-text classification, bounding box regression and RNN (Recurrent Neural Network) based adaptive text region representation are used to classify the contents of the image as text or non-text. The final output is the text regions labeled with polygons with an adaptive number of vertices to accurately demarcate the text area[3]. Another solution[4] proposes a unified text detection system. It integrates the traditional false character candidate removal, text line extraction, and text line verification into a single process making it an efficient model to detect text. It employs a similar character detection system as used in other proposed solutions[7], which proposes a solution to read and detect text in natural scenes. It uses OCR to text regions returned by AdaBoost (an algorithm that combines weak image features to make stronger ones) and is followed by the Extension and Binarization stage which detects text regions which are then inputs to the OCR reading stage.

Artificial Intelligence (AI) models have been used for Natural Language Processing (see Glossary), predictions and for forecasts[8]. Generative AI, in specific, has been used to create multimodal results, including text, images, videos, designs, etc.[9]

OCR and similar technologies have been used to detect text in already computerized files like PDF files. But the same cannot be employed for images of textbooks that have dense text and visualizations and that might not be captured in the best lighting conditions and from the best angle. Solutions like Rosetta, employ OCR to recognize text in images. Written by employees at Facebook Inc., this solution was implemented to extract text on a Facebook scale[10]. There are solutions that have been created to counter the drawbacks of using OCR. For example, constructing and evaluating two systems. The first, a two-stage pipeline consisting of text detection followed by a leading OCR engine. The second is a system based on generic object recognition[11]. An arbitrary orientation network (AON) provides a solution to this. It can capture features of irregular texts. Its architecture consists of four components: 1) the basal convolutional neural network (BCNN) module for low-level visual representation; 2)

the arbitrary orientation network (AON) for capturing the horizontal, vertical and character placement features; 3) the filter gate (FG) for combining four feature sequences with the character placement clues; 4) the attention-based decoder (Decoder) for predicting character sequence. The solution was tested on several regular and irregular datasets including but not limited to CUTE80, ICDAR 2015 and IIIT5K-Words [12].

The traditional methods, using OCR, or a variant of it, AON, etc., have the downside of not being accessible. Newer solutions include employing deep learning to get around the challenges caused due to backgrounds, varying text, in style, orientation and lighting, and other similar reasons. Deep learning provides a solution using image classification, object detection, semantic segmentation and sequence modeling [13].

Recent studies have explored the use of multimodal AI, tactile learning tools, and audio-based aids to support visually impaired students. AI-powered audio learning platforms [14], automated tactile graphic generation systems [15], and hybrid audio-tactile interfaces [16] are among the emerging solutions that go beyond traditional OCR methods. These advancements offer important context for this study, which seeks to evaluate vision-language models as accessible educational tools.

VLMs (see Glossary) like ChatGPT, Gemini and Claude now accept text as well as images as input. Being available publicly and for free, these make these more preferable solutions in comparison to the solutions used earlier.

Problem Statement and Rationale

Textbooks are dense, with visualizations and images wrapping the text. The dense nature of the text and visualizations in textbooks makes it difficult to readily convert to text, which can in turn be converted to an audio format. This makes it difficult for visually impaired students to use regular textbooks on a day-to-day basis.

I have been working with visually impaired students, in a more rural part of my city, for a year now, to help them learn to play the piano. From my observations, and after talking to their caregivers, I have come to realize that they do not have a viable solution in place to solve this problem. The reason for their lack of access to more resources, like textbooks in Braille, is due to their lack of financial resources and infrastructure. They presently have someone sit with the students when they study, to read out the contents of their textbooks to them. This is not a feasible solution in the long term and constrains their studies.

Significance and Purpose

Textbooks have the benefit of explaining concepts in great detail, as opposed to presentations or slides. This makes them highly valuable to students. The uniqueness of every subject's textbook poses an additional challenge to the possibilities students have to learn. Providing a reliable AI-based alternative that can interpret and read out dense textbook content would make education significantly more accessible to visually impaired students, especially those in under-resourced areas.

Objectives

This experiment aims to determine which VLM is better suited to replace the individual these students now require to study. Integrating the better of these models into a mobile app would reduce the dependency of these students on other individuals for their studies.

Scope and Limitations

The study focuses specifically on high school-level textbook content and compares two publicly available generative AI VLMs GPT-4o and Claude 3.5 Sonnet. A majority of the dataset consists of images from high school textbooks due to a lack of access to textbooks of other grade levels. Limitations include the scope of model evaluation, the size of the dataset and the variability in textbook layouts, languages, grade levels and image quality.

Theoretical Framework

This work draws on the ongoing development and evaluation of computer vision and multimodal generative AI systems. It operates under the assumption that newer models like VLMs have surpassed traditional OCR in both flexibility and accessibility, making them viable for real-time educational support applications.

Methodology Overview

Hence, we ran a proof of concept experiment to compare two publicly available generative AI models GPT-4o and Claude 3.5 Sonnet. Both these models are VLMs, which are different from LLMs (Large Language Model) (see Glossary) in the sense that they are multimodal models, taking images, and other forms of input in addition to text, and return outputs of different types as well. Both models used in this experiment accept both text and images as input. Their performance was assessed based on accuracy, time taken, and qualitative analysis of their outputs.

Proof of Concept Experiment

Introduction:

Generative AI has been used in various fields due to its ability to process huge amounts of data and assist with several tasks. This technology has not been applied in the field of education, as of now. The text in textbooks varies across pages, with images,

illustrations and tables scattered throughout the text. The dense nature of the text makes it more difficult to detect the text in it. To better understand how feasible it is to use AI on a day-to-day basis to detect text from textbook pages, I compared different AI models and evaluated their performance in detecting and understanding text, graphics and images, and in summarizing the same for a student.

The models selected for this experiment are GPT-4o and Claude 3.5 Sonnet. The reason for choosing these models is that they are readily accessible. This makes them preferred because students are more likely to be able to use these frontend models.

Both of these models are VLMs, taking both images and text as input for processing. Whereas, LLMs specialize in dealing mostly with text.

OpenAI's GPT-4o was released as its flagship model. Presently, it stands at No. 1 on the LMSYS leaderboard[17]. It has been shown to surpass previous models in its accuracy with both audio and visual inputs[18]. Figure 1 shows the comparisons between GPT-4o and other similar models on various benchmarks.

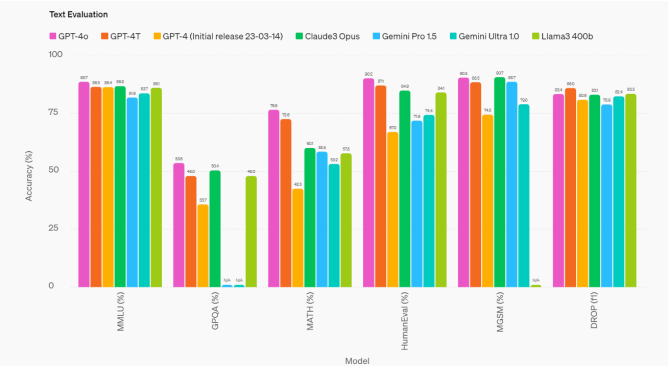


Fig. 1 [18]

Claude 3.5 Sonnet is Anthropic's strongest vision model yet, surpassing benchmarks set by previous Anthropic models.[19]

GPT-4o has a context window (see Glossary) of 128,000 tokens[20]. Claude 3.5, on the other hand, has a context window of 200, 000 tokens.[21]

The LMSYS Leaderboard is a result of crowdsourced evals on LLMs[17] It is a platform on which users can vote anonymously, and can choose between two anonymous chatbots. This makes sure the votes are unbiased. It uses LLM benchmarks, the Elo ranking system, and a human preference dataset. It uses the Bradley-Terry model for pairwise evaluations of models, to arrive at the final ranking.

The comparison carried out in this proof of concept experiment was between OpenAI's GPT-4o and Anthropic's Claude 3.5. According to the LMSYS leaderboard, the latest version of GPT-4o ranks 1st, with an arena score of 1277, while Claude 3.5 Sonnet ranks 6th[14].

A VLM leaderboard (OpenVLM Leaderboard), uses the results of the OpenSource Framework to rank VLMs[21]. According to this leaderboard, the latest version of GPT-4o ranks 1st, while Claude 3.5 Sonnet ranks 6th.

The MMMU benchmark[22]assesses the multimodal models on topics involving in-depth subject knowledge and reasoning. GPT-4o has set an MMMU benchmark of 69.1%[22]and Claude 3.5 Sonnet has a score of 68.3[19].

Considering that this experiment deals with both the recognition of text from images, as well as summarizing it, it is ideal to compare these two VLMs, to determine which serves this specific purpose better, both quality-wise and time-wise.

Data

I created my own dataset for this experiment. The dataset consisted of 50 images of various textbooks science, social science and language learning to ensure variation in the kind of data on each page. The approximate composition of the dataset was as follows: 37.5% Science, 37.5% Computer Science, 17.5% History and 17.5% Geography. The images were taken in different orientations, from different angles and in varied lighting conditions (around 55% in controlled indoor lighting and 45% in natural daylight) to ensure that the test is accurate. The images were taken using the 48 megapixel camera of a mobile phone, to resemble how students would take similar images.

Methods:

Two VLMs GPT-4o and Claude 3.5 Sonnet were used to detect text, produce summaries, and then to compare the two summaries produced to arrive at the best one.

An auto-evaluation was conducted in a new window of GPT-4o to ensure that there is no bias in the evaluation. (see Glossary) This was done to make for a more viable solution in the case of the usage of this solution in an app. Auto-evaluations have been used in the past to replace human evaluations, and to avoid bias.[23]

Comparison:

Prompts to the VLM

1. Text Recognition: Detect all text on the page in the image. Summarize diagrams or illustrations, if any. Do not summarize the text, return all of it. (along with the image)

2. Summary Generation: Generate a 150-word summary of the text on the page while adhering to the following:
   1. Every heading in the image is covered in the summary.
   2. Sections are described with no key terms left out.
   3. Illustrations are described.
   4. If the textbook is a mathematics textbook, include all formulae in the summary.
   5. If it is a science textbook, ensure that summaries of the

details depicted in diagrams and equations/formulae are mentioned.
6. If it is a history/geography/language learning textbook, ensure all headings are covered with description.

3. Auto-evaluation: Compare <1st summary> with <2nd summary>, and determine which one is better on the basis of the following criteria.
1. The summary encompasses every concept on the page.
2. Images and illustrations on the page are detected and described.
3. The summary gives a clear understanding of the topics on the page.

After passing the image to the VLM, and retrieving the text it returned, I conducted a manual check on the text and evaluated it. I read through the text to check the following:
1. If any text was left out.
2. If any text was incorrectly detected.
3. If images and other visualizations diagrams, mind maps, tables were detected correctly.
4. If the images in the textbook were recognized and described correctly, or to check how close its recognition of the image was to what the image depicted.

As shown in Table 1, Claude 3.5 Sonnet outperformed GPT-4o in 85% of test cases.

The model was then asked to create a summary and compare it with the summary generated by the second model on the basis of the following rubrics:

1. Every heading in the image is covered in the summary.

2. Sections are described with no key terms left out.

The quality of the generated summary is evaluated manually on the basis of the breadth and detail of the text on the textbook page.

The same cycle was repeated for every image in the data set.

The purpose of the auto-evaluation is to compare the summaries generated by the two models in an unbiased manner. Both models are given the same image, the same prompts and are asked to generate summaries. Both summaries are passed to both models and they are asked to compare the two on the basis of the criteria mentioned below. The results of the auto-evaluation are then compared to those of a human evaluation, to check if the summaries provide complete understanding of the contents of the page.

In addition to the auto-evaluation, the summaries were also evaluated by visually impaired children to assess real-world accessibility and effectiveness; the results of their feedback are summarized below.

On average, while analysing the content itself, Claude 3.5 Sonnet performed better across subjects. However, an important consideration to make is that it does not have a read aloud feature as of now, whereas GPT-4o does. One solution would be to use a third party text-to-speech solution when integrating Claude 3.5 Sonnet into an app. Use of a Claude model in the future which does have this feature available would also solve this problem.

The errors noted in the text and graphics detection of the models has been summarised in the table below:

Results:

Overall, as seen in Table 1, Claude 3.5 Sonnet performed the task of detecting and summarizing the contents of a textbook page better. The text detection was performed better by GPT-4o, on average, and the summaries created by Claude 3.5 Sonnet were more comprehensive. Claude 3.5 Sonnet detected images and visualizations and described them more accurately.

Each conversion was timed, and averaged out to 35.155 seconds, per image, for GPT-4o and to 20.2405 seconds, per image, for Claude 3.5 Sonnet, making the latter significantly faster than the former.

Analysis:

An analysis suggests that GPT-4o is about 1.7x faster than Claude-3.5 Sonnet and has a lower latency[24]. However, the results of the above experiment show otherwise. This could be due to high traffic when the experiment was run. However, this requires further investigation.

On conducting a t-Test, a statistical test to compare if the difference in the means of two groups are significant, or might have happened by chance (see Glossary)[25] on the paired values, it is evident that Claude 3.5 Sonnet significantly outperformed GPT-4o.

$$t = \frac{\bar{d}}{s/\sqrt{n}}$$

Where: t = t-value,
$\bar{d}$ = mean of the differences between readings = 13.29 s,
s = standard deviation of the differences = 18.63 s,
n = number of readings = 40.

Further, on conduction a two-tailed test (see Glossary)[26] with code to find the p-value, the following is obtained.

```
from scipy import stats

t_stat = 4.51
df = 39
p_value = stats.t.sf(t_stat, df) * 2
```

Where, df refers to the degrees of freedom which is n - 1. On calculating, p = 0.000057, which implies that the observed

Table 1 Comparison of GPT-4o and Claude 3.5 Sonnet on Text and Image Summarization Tasks

Image No	**Time Taken**		**Auto-eval (Better summary)**	**Human Eval (Better summary)**	**Final Result**
	GPT-4o (X)	**Claude-3.5 Sonnet (Y)**			
1(Science)	24.74s	17.26s	Y	Y	Y
2	31.38s	23.65s	Y	Y	Y
3	47.43s	19.05s	Y	Y	Y
4	44.01s	25.00s	Y	Y	Y
5	52.16s	15.11s	Y	Y	Y
6	1m 10.81s	22.91s	Y	X	X
7	1m 11.26s	10.88s	Y	Y	Y
8	1m 13.75s	25.53s(figure captions recognized incorrectly)	Y	X	X
9	18.79s	22.10s	Y	Y	Y
10	23.67s	15.32s	Y	Y	Y
11	14.02s	29.13s	Y	Y	Y
12	17.45s	21.34s	Y	Y	Y
13	13.89s	23.89s	Y	Y	Y
14	17.22s (Reactions undetected)	19.87s	Y	Y	Y
15	26.43s	15.73s	Y	Y	Y
16 (CS)	29.24s	22.16s	Y	Y	Y
17	30.73s	23.38s	X	X	X
18	30.63s	23.70s	Y	Y	Y
19	37.43s	28.30s	Y	Y	Y
20	37.85s	23.15s (text incorrectly detected)	X	X	X
21	23.03s	24.67s	Y	Y	Y
22	16.11s	17.38s	Y	Y	Y
23	20.49s	17.92s	Y	Y	Y
24	22.17s	15.63s	Y	Y	Y
25	13.64s	19.78s	Y	Y	Y
26	15.61s	13.46s	Y	X	X
27	14.32s	11.47s	Y	Y	Y
28	14.46s	16.71s	Y	Y	Y
29	15.56s	13.82s	Y	Y	Y
30	17.40s	26.24s	Y	Y	Y
31(History)	1m 13.60s	25.70s	Y	X	X
32	59.12s	19.72s	Y	Y	Y
33	47.68s	22.36s	Y	Y	Y
34	50.32s	20.60s	Y	Y	Y
35	39.48s	11.04s	Y	Y	Y
36	43.81s	15.61s	Y	Y	Y
37	51.09s	13.05s	Y	Y	Y
38 (Geography)	33.74s	27.22s	Y	Y	Y
39	29.07s	26.90s	Y	Y	Y
40	27.83s	22.88s	Y	Y	Y

Table 2 Results of User-testing with Visually-impaired Students on Average

Subject — Criteria		Clarity	Readability	Informativeness
Science	GPT-4o	4	3	4
	Claude 3.5 Sonnet	-	4	4
Computer Science	GPT-4o	3.3	4	3.66
	Claude 3.5 Sonnet	-	3.66 (one grammatical error)	4.66
History	GPT-4o	4	4.5	4 (images described well)
	Claude 3.5 Sonnet	-	5	5(dates, names of people added to description; descriptive)
Geography	GPT-4o	4	4	5
	Claude 3.5 Sonnet	-	4	5

Table 3 Summary of the Errors Made by the VLMs

Error Type	Example	Frequency (%)
Misrecognized Text	Claude misread figure captions (Image 8); text detection was faulty (Image 20)	5%
Missing Visual Information	GPT-4o missed detecting reactions in a chemistry diagram (Image 14)	2.50%

difference in processing times between GPT-4o and Claude 3.5 Sonnet is highly statistically significant (meaning the probability of this difference having occurred by chance is 0.000057%). This means it is extremely unlikely that the difference occurred by chance, providing strong evidence that Claude 3.5 Sonnet processes the provided images faster than GPT-4o.

Cohen's d is a value that quantifies the largeness of the difference of the means between two groups in standard deviation units, here, in terms of the performance of both models.[27] On calculating Cohen's d,

$dc = \frac{M_1 - M_2}{sd}$

$dc = \frac{M_1 - M_2}{sd}$

Where:

M_1 = mean of performance of GPT-4o,

M_2 = mean of performance of Claude 3.5 Sonnet,

Standard deviation $sd = \sqrt{p(1-p)}$

Where: $p = \frac{34}{40} = 0.85$

Therefore, sd = 0.357.

On considering all readings, GPT-4o performed better in 6 readings, resulting in a binary M_1 value of 4/40 = 0.15, and Claude 3.5 Sonnet did better in 34, resulting in an M_2 value of 85/40 = 0.85.

Therefore,

$dc = \frac{0.15 - 0.85}{0.357} = -1.96$

This is a very large negative value, indicating that GPT-4o performed significantly worse than Claude 3.5 Sonnet.

Below is a figure illustrating the accuracy and average processing time of both models in a bar graph for comparison.

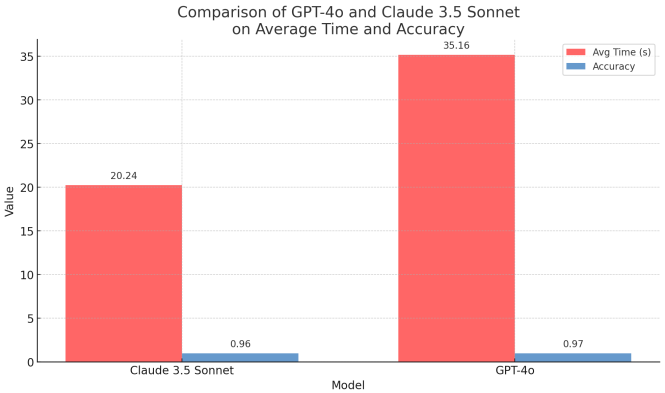


Fig. 2 Comparison of GPT-4o and Claude 3.5 Sonnet on accuracy and average time

Some other observations made during this experiment are that visualizations, including chemical reactions, were incorrectly detected, while formulae and equations were, for the most part, correctly detected. Claude 3.5 Sonnet did hallucinate, and return

content that did not exist on the input page. All of these errors together resulted in an error percentage of 7.5%.

Here is a line graph showing the accuracy of the VLMs across grade levels. Due to the restricted size of the dataset, accuracy in elementary and middle school textbook images were 100%. However, on expanding the dataset, more cases with errors are bound to occur.

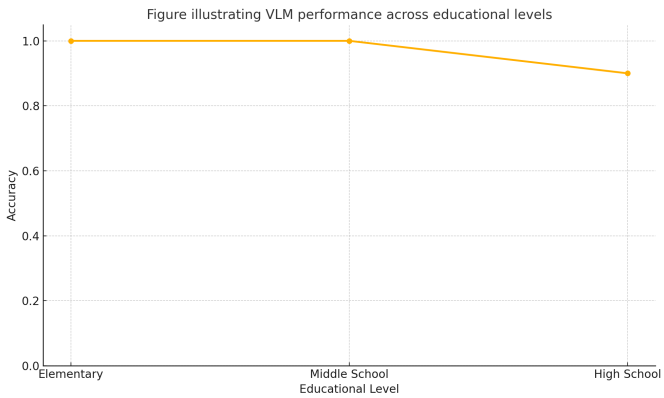


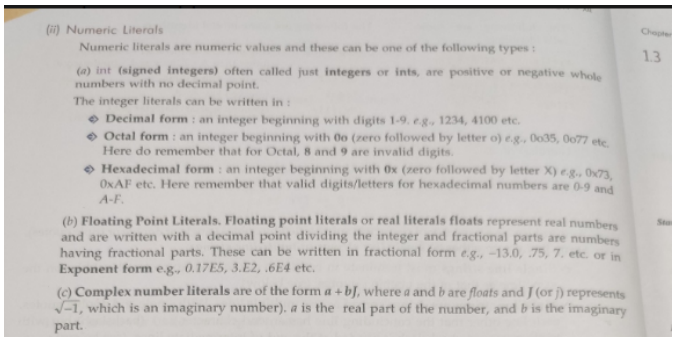
Fig. 3 Figure illustrating VLM performance across educational levels

Another observation is that GPT-4o returned formulae in a better formatted way than Claude 3.5 Sonnet did.

A few limitations that were observed through the course of this experiment include:

1. Claude 3.5 Sonnet did hallucinate, resulting in incorrect information, thus lowering the factuality of the output. This may be because it had read the text incorrectly.

Another test that was conducted was to find high confidence sections for a given image input, to detect patterns in the regions of low confidence. Prompt given to VLM: Give me the high


(ii) Numeric Literals

Numeric literals are numeric values and these can be one of the following types :

(a) int **(signed integers)** often called just **integers** or **ints**, are positive or negative whole numbers with no decimal point.

The integer literals can be written in :

- **Decimal form** : an integer beginning with digits 1-9. *e.g.*, 1234, 4100 etc.
- **Octal form** : an integer beginning with **0o** (zero followed by letter o) *e.g.*, 0o35, 0o77 etc. Here do remember that for Octal, **8** and **9** are invalid digits.
- **Hexadecimal form** : an integer beginning with **0x** (zero followed by letter X) *e.g.*, 0x73, 0xAF etc. Here remember that valid digits/letters for hexadecimal numbers are 0-9 and *A-F*.

(b) **Floating Point Literals. Floating point literals** or **real literals floats** represent real numbers and are written with a decimal point dividing the integer and fractional parts are numbers having fractional parts. These can be written in fractional form *e.g.*, –13.0, .75, 7. etc. or in **Exponent form** e.g., *0.17E5, 3.E2, .6E4* etc.

(c) **Complex number literals** are of the form $a + bJ$, where a and b are *floats* and J (or j) represents $\sqrt{-1}$, which is an imaginary number). a is the real part of the number, and b is the imaginary part.

Fig. 4 Image from a CS textbook

confidence and low confidence regions (high confidence having probability > 0.7 and low confidence being < 0.7) on recognizing the text in the below image. Put it in a paragraph with the high confidence text in an ordinary font, and the low confidence parts in bold.

VLM response: (ii) Numeric Literals Numeric literals are numeric values and these can be one of the following types:

a. int (signed integers) often called just integers or ints, are positive or negative whole numbers with no decimal point. The integer literals can be written in:

   ⊕ Decimal form: an integer beginning with digits 1–9, e.g., 1234, 4100 etc.

   ⊕ Octal form: an integer beginning with `0x` (zero followed by letter x) e.g., `0x8`, `0x7` etc. Here we understand that the Octal, 8 and 9 are invalid digits/letters for hexadecimal numbers, so we get an error.

   ⊕ Hexadecimal form: an integer beginning with `0x` (zero followed by letter x) e.g., `0x73`, `0xA1` etc. Here we understand that the digits 0–9, letters A–F are valid digits/letters for hexadecimal numbers, so we get 0–15.

b. Floating Point Literals. Floating point literals or real literals floats represent real numbers and are written with a decimal point dividing the integer and fractional parts are numbers. Having fractional parts. These can be written in fractional form e.g., -15.0, .75, 7. etc. or in Exponent form e.g., 0.1E2, 3.62, .614 etc.

c. Complex number literals are of the form $a + bj$, where a and b are floats and j (or J) represents $\sqrt{-1}$, which is an imaginary number. a is the real part of the number, and b is the imaginary part.

On analysis, it is evident that the low-confidence parts are mainly the technical examples, mathematical symbols, and specific code notation where OCR might struggle with precision in detection. Future experiments could look into methods to describe the formulae in a way that is comprehensive for a listener, rather than a reader.

While VLMs do provide opportunities to improve accessibility, there are also several ethical implications to consider.

One important consideration is hallucination, when the model generates inaccurate information. In an educational setting, this is especially problematic as it can mislead or misinform students. Also, biases in the models training data can result in inconsistent performance across different languages, subjects or groups of people. To minimise the risk of users being misled by these problems, including confidence scores and grounding the model in the textbook data provided could help the models produce more accurate outputs.

The hallucination could potentially be corrected by engineering the prompt given to the model and by grounding the model

in the data provided and the problem of the low confidence sections could be addressed through training the model, and fine tuning it to deal with more mathematical symbols

2. Batching input images increases the latency of the model, as well as lowers its accuracy. This could be corrected by further experiments and research, looking at and targeting the specific causes for this occurrence.

3. This experiment is that it solely focuses on testing the models capabilities in English, but expanding to other languages (Spanish, Mandarin, etc.) would expand the applicability of this model.

4. This solution does require constant internet access to use the model.

Here is a table to compare the proposed solution with existing solutions:

Conclusion

The results of the experiment show that Claude 3.5 Sonnet serves this specific use of generative AI better. The consistent accuracy of the model shows that it can in fact be used in the proposed solution, with further training and grounding, to further minimise errors in detection and hallucination. The integration of it into a mobile app would make employing this solution easier for visually impaired students.

Future work:

Due to limitations in accessing textbooks in other languages, this study has been limited to English, but could be extended to other languages in the future. Additionally, this solution presently would not cater to students with textbooks in languages that the model does not recognize. This is another issue that would need to be addressed in future, to prevent the model from hallucinating when it is asked to process information in an unknown language.

The requirement of access to the internet constantly makes the solution less feasible for students in rural areas with limited, or no, access to high-speed internet. A solution to getting around this problem would be to conduct a larger-scale experiment on similar lines, to compare on-device VLMs including Nano 1.0[28], thus optimizing this solution for lower-infrastructure areas. These models do not require internet connection for use, due to their smaller size, thus making them a more viable choice. Their latency is also reduced, due to their smaller size. This makes these on-device VLMs perfect for comparison using an experimental structure similar to the one used in this paper, to determine which model performs better.

In the case of integrating into an app, a few factors that would have to be considered are: availability (cost-free access to VLM), availability across multiple languages, ability to return and accept audio files, safety in using the model, the factuality of the model, as well as locale of the user. To allow for offline usage, on-device VLMs, which are smaller in size, would be preferable.

Another important improvement to make would be to train the VLM on several varieties of text inputs, with various dense or graphic text areas in it.

One other limitation is the latency in both models, some of which exceed 15-20 seconds. In a real-world setting, these could be inconvenient to a student. One solution to fix this, is to avoid batching images, and to send images in one at a time, to reduce the load on the VLM. Also, in the future, on-device VLMs would work faster, due to their smaller size, and hence, lower latency.

References

1 B. C. G. Lee, J. Mears, E. Jakeway, M. Ferriter, C. Adams, N. Yarasavage, D. Thomas, K. Zwaard and D. Weld, *The Newspaper Navigator Dataset: Extracting And Analyzing Visual Content from 16 Million Historic Newspaper Pages in Chronicling America*, 2020.

2 S. Mishra, *Assistive Technologies for Visual Impairment Enhancing Access to Library Resources*, 2023.

3 X. Wang, Y. Jiang, Z. Luo, C. Liu, H. Choi and S. Kim, *Arbitrary Shape Scene Text Detection with Adaptive Text Region Representation*, 2019.

4 S. Tian, Y. Pan, C. Huang, S. Lu, K. Yu and C. L. Tan, *Text Flow: A Unified Text Detection System in Natural Scene Images*, 2015.

5 Z. Zhang, W. Shen, C. Yao and X. Bai, *Symmetry-Based Text Line Detection in Natural Scenes*, 2015.

6 T. Karmakar, *Region Proposal Network (RPN) — Backbone of Faster R-CNN*, https://medium.com/@codeplumber/region-proposal-network-rpn-backbone-of-faster-r-cnn-4a744a38d7f9, 2018.

7 X. Chen and A. L. Yuille, *Detecting and Reading Text in Natural Scenes*, 2004.

8 *What are AI models?*, https://www.hpe.com/in/en/what-is/ai-models.html#:~:text=AI%20models%20can%20be%20used,and%20robotics%20and%20control%20systems.

9 *Gartner Experts Answer the Top Generative AI Questions for Your Enterprise*, https://www.gartner.com/en/topics/generative-ai.

10 F. Borisyuk, A. Gordo and V. Sivakumar, *Rosetta: Large scale system for text detection and recognition in images*, 2018.

11 K. Wang, B. Babenko and S. Belongie, *End-to-end scene text recognition*, 2011.

12 Z. Cheng, Y. Xu, F. Bai, Y. Niu, S. Pu and S. Zhou, *AON: Towards Arbitrarily-Oriented Text Recognition*, 2018.

13 S. Long, X. He and C. Yao, *Scene Text Detection and Recognition: The Deep Learning Era*, 2021.

Table 4 Comparative Evaluation of Assistive Tools Across Key Metrics

Assistive Tool	Speed	Cost	Accessibility	Accuracy
Claude 3.5 Sonnet	Real-time	Free of cost (internet required)	High	92%
Braille Textbooks	Slow	High cost	Moderate-low	100%
Human-read Audio	Moderate	Variable cost	Moderate	95% (mostly accurate)

14 C. Yang and P. Taele, *AI for Accessible Education: Personalized Audio-Based Learning for Blind Students*, 2025.

15 A. Khan *et al.*, *TactileNet: Bridging the Accessibility Gap with AI-Generated Tactile Graphics*, 2025.

16 M. Maćkowski and P. Brzoza, *Accessible Tutoring Platform Using Audio-Tactile Graphics Adapted for Blind Users*, 2022.

17 *LMSYS Chatbot Arena Leaderboard*, https://huggingface.co/spaces/lmsys/chatbot-arena-leaderboard, Accessed: 30th August 2024.

18 *Hello GPT-4o*, https://openai.com/index/hello-gpt-4o/, 2024.

19 *Claude 3.5 Sonnet*, https://www.anthropic.com/news/claude-3-5-sonnet, 2024.

20 S. M. Kerner, *GPT-4o explained: Everything you need to know*, https://www.techtarget.com/whatis/feature/GPT-4o-explained-Everything-you-need-to-know, 2024.

21 *OpenVLM Leaderboard*, https://huggingface.co/spaces/opencompass/open_vlm_leaderboard, Accessed: 30th August 2024.

22 *MMMU*, https://github.com/MMMU-Benchmark/MMMU.

23 C. Lin and E. Hovy, *Manual and Automatic Evaluation of Summaries*, 2002.

24 Hendrix, *Claude 3.5 Sonnet vs. GPT 4o: which is better?*, https://medium.com/@hendrix_56915/claude-3-5-sonnet-vs-gpt-4o-which-is-better-f4c4fe3a8f16#:~:text=In%20benchmark%20evaluations%2C%20Claude%203.5,math%2C%20and%20reasoning%20over%20text., 2024.

25 *An Introduction to t Tests*, https://www.scribbr.com/statistics/t-test/#:~:text=What%20is%20a%20t%2Dtest,means%20is%20different%20from%20zero.

26 *What Is a Two-Tailed Test? Definition and Example*, https://www.investopedia.com/terms/t/two-tailed-test.asp.

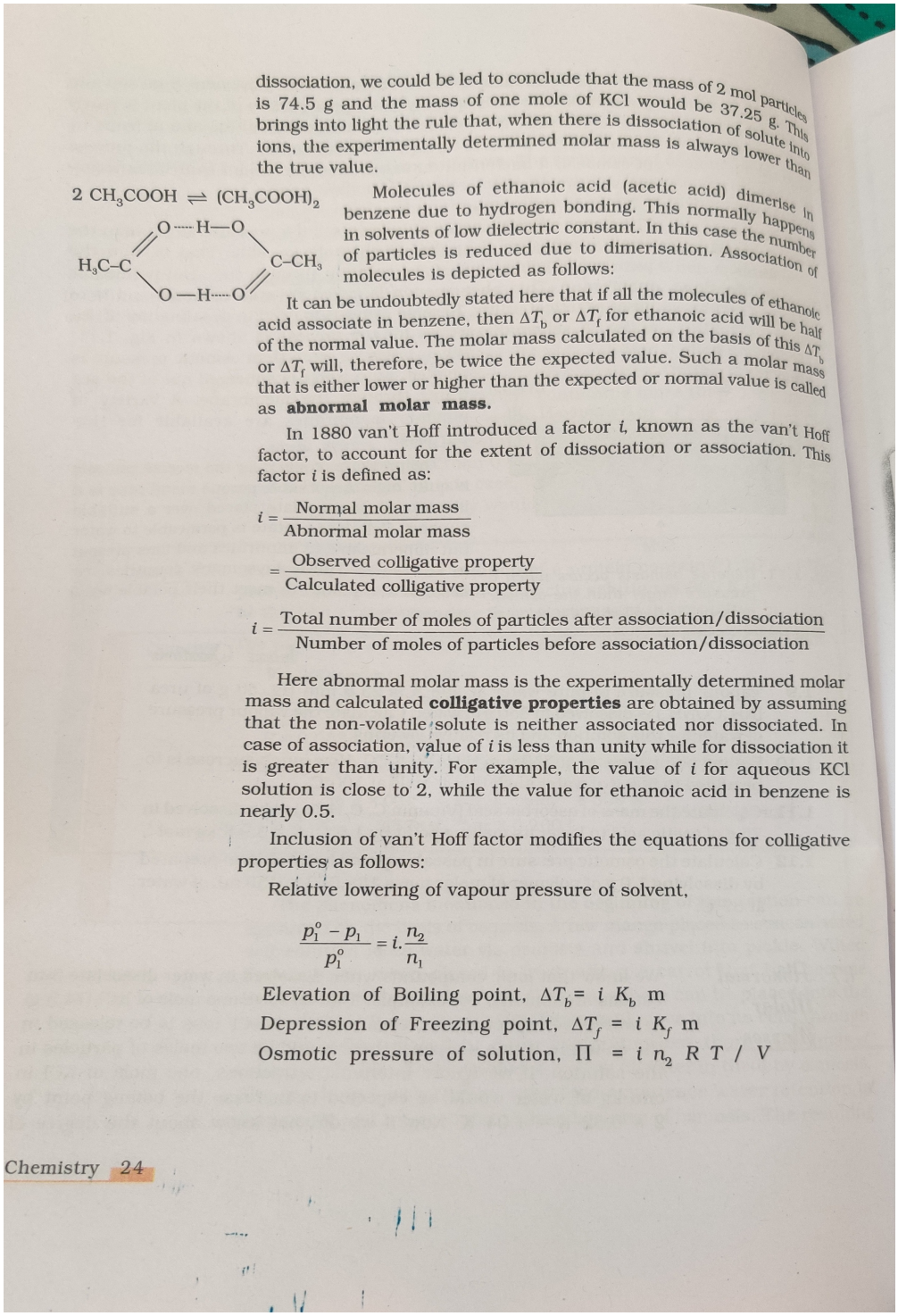
27 *Cohens d*, https://en.wikiversity.org/wiki/Cohen%27s_d#:~:text=Cohen's%20d%20is%20an%20effect,the%20comparison%20between%20two%20means.

28 *Nano 1.0*, https://deepmind.google/models/gemini/nano/.

Glossary

1. OCR: Optical Character Recognition. A technology used to convert images of text into machine-readable text.
2. RCNN: Region-based Convolutional Neural Network. A type of deep learning model used to find and identify objects in an image by first proposing possible regions where the objects might be, and then analyzing those regions to classify whats inside them.
3. Natural Language Processing: A field of artificial intelligence that helps computers understand, interpret, and generate human language, so they can interact with us more naturally.
4. VLM: Vision-Language Model. An AI model that can understand and work with both images and text together, allowing it to answer questions about pictures, describe images, or combine visual and written information in its responses.
5. LLM: Large Language Model. An AI model trained on huge amounts of text so it can understand and generate human-like language, often used for answering questions, writing text, or holding conversations.
6. Token context window: Token context window here refers to the amount of text (measured in tokens, which are chunks of words) a model can consider at once when generating responses. A larger context window means the model can understand and reference more information from the conversation or document at a time.
7. Auto-evaluation: Auto-evaluation is an automated process where one model evaluates another models performance based on criteria like completeness and accuracy, without the need for human intervention.
8. t-Test: statistical test to compare if the difference in the means of two groups are significant, or might have happened by chance.
9. Two-tailed test: A test used to find whether there is a significant difference between two groups from the mean on either side of it.

Appendix I: Dataset Samples



dissociation, we could be led to conclude that the mass of 2 mol particles is 74.5 g and the mass of one mole of KCl would be 37.25 g. This brings into light the rule that, when there is dissociation of solute into ions, the experimentally determined molar mass is always lower than the true value.

$2\ CH_3COOH \rightleftharpoons (CH_3COOH)_2$

Molecules of ethanoic acid (acetic acid) dimerise in benzene due to hydrogen bonding. This normally happens in solvents of low dielectric constant. In this case the number of particles is reduced due to dimerisation. Association of molecules is depicted as follows:

It can be undoubtedly stated here that if all the molecules of ethanoic acid associate in benzene, then ΔT_b or ΔT_f for ethanoic acid will be half of the normal value. The molar mass calculated on the basis of this ΔT_b or ΔT_f will, therefore, be twice the expected value. Such a molar mass that is either lower or higher than the expected or normal value is called as **abnormal molar mass.**

In 1880 van't Hoff introduced a factor *i*, known as the van't Hoff factor, to account for the extent of dissociation or association. This factor *i* is defined as:

$$i = \frac{\text{Normal molar mass}}{\text{Abnormal molar mass}}$$

$$= \frac{\text{Observed colligative property}}{\text{Calculated colligative property}}$$

$$i = \frac{\text{Total number of moles of particles after association/dissociation}}{\text{Number of moles of particles before association/dissociation}}$$

Here abnormal molar mass is the experimentally determined molar mass and calculated **colligative properties** are obtained by assuming that the non-volatile solute is neither associated nor dissociated. In case of association, value of *i* is less than unity while for dissociation it is greater than unity. For example, the value of *i* for aqueous KCl solution is close to 2, while the value for ethanoic acid in benzene is nearly 0.5.

Inclusion of van't Hoff factor modifies the equations for colligative properties as follows:

Relative lowering of vapour pressure of solvent,

$$\frac{p_1^o - p_1}{p_1^o} = i.\frac{n_2}{n_1}$$

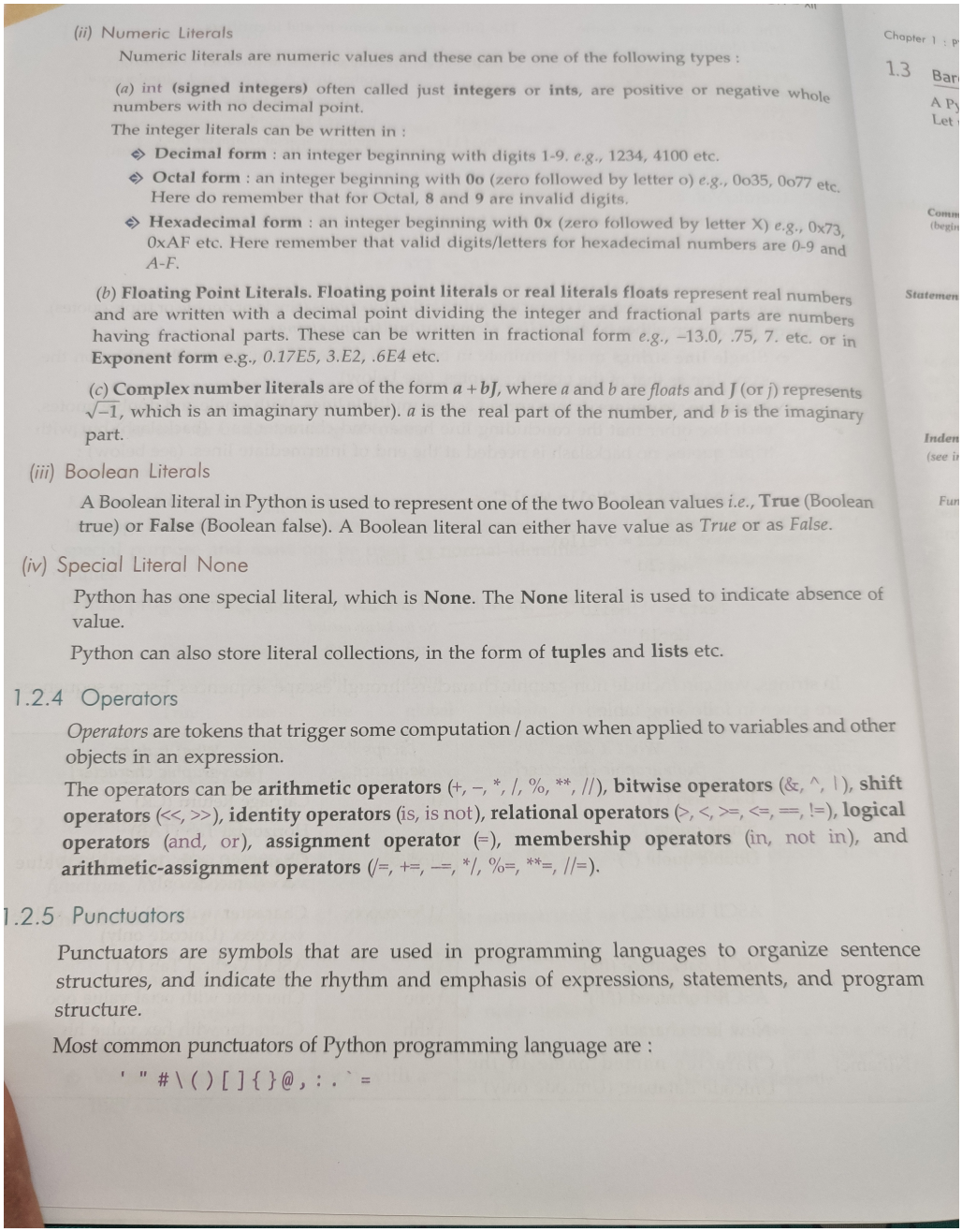
Elevation of Boiling point, $\Delta T_b = i\ K_b\ m$

Depression of Freezing point, $\Delta T_f = i\ K_f\ m$

Osmotic pressure of solution, $\Pi = i\ n_2\ R\ T\ /\ V$

Chemistry 24

1. Science



(ii) Numeric Literals

Numeric literals are numeric values and these can be one of the following types :

(a) int **(signed integers)** often called just **integers** or **ints**, are positive or negative whole numbers with no decimal point.

The integer literals can be written in :

- **Decimal form** : an integer beginning with digits 1-9. *e.g.*, 1234, 4100 etc.
- **Octal form** : an integer beginning with **0o** (zero followed by letter o) *e.g.*, 0o35, 0o77 etc. Here do remember that for Octal, **8** and **9** are invalid digits.
- **Hexadecimal form** : an integer beginning with **0x** (zero followed by letter X) *e.g.*, 0x73, 0xAF etc. Here remember that valid digits/letters for hexadecimal numbers are 0-9 and *A-F*.

(b) **Floating Point Literals. Floating point literals** or **real literals floats** represent real numbers and are written with a decimal point dividing the integer and fractional parts are numbers having fractional parts. These can be written in fractional form *e.g.*, –13.0, .75, 7. etc. or in **Exponent form** e.g., *0.17E5, 3.E2, .6E4* etc.

(c) **Complex number literals** are of the form $a + bJ$, where a and b are *floats* and J (or j) represents $\sqrt{-1}$, which is an imaginary number). a is the real part of the number, and b is the imaginary part.

(iii) Boolean Literals

A Boolean literal in Python is used to represent one of the two Boolean values *i.e.*, **True** (Boolean true) or **False** (Boolean false). A Boolean literal can either have value as *True* or as *False*.

(iv) Special Literal None

Python has one special literal, which is **None**. The **None** literal is used to indicate absence of value.

Python can also store literal collections, in the form of **tuples** and **lists** etc.

1.2.4 Operators

Operators are tokens that trigger some computation / action when applied to variables and other objects in an expression.

The operators can be **arithmetic operators** (+, –, *, /, %, **, //), **bitwise operators** (&, ^, |), **shift operators** (<<, >>), **identity operators** (is, is not), **relational operators** (>, <, >=, <=, ==, !=), **logical operators** (and, or), **assignment operator** (=), **membership operators** (in, not in), and **arithmetic-assignment operators** (/=, +=, –=, */, %=, **=, //=).

1.2.5 Punctuators

Punctuators are symbols that are used in programming languages to organize sentence structures, and indicate the rhythm and emphasis of expressions, statements, and program structure.

Most common punctuators of Python programming language are :

' " # \ ( ) [ ] { } @ , : . ` =

2. Computer Science

had earlier prepared a charter of Demands or a Manifesto for the German Communist League. This document was later given the shape of *Communist Manifesto* published in 1848. Another famous book of Marx and Engels is *Das Kapital*. Marx could complete only the first volume before his death in 1883. The remaining two volumes were published by Engels several years later.

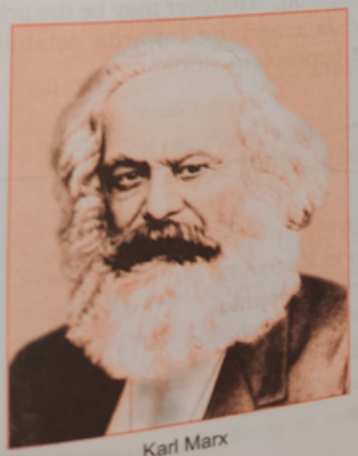
Karl Marx considered capitalist society as a society divided between two classes — the working class which produces all value; and the owning and employing class, which without producing anything, exploits the value or profits. This leads to a class struggle between the working class and the employing class. Eventually, this class struggle leads to a crisis, causing the collapse of the capitalist system. The working class would take over power, organise production for its own benefit as a class. This new society would be '*socialist*' in nature, a society without exploitation. The system behind this society is known as '*Marxist Socialism*' or 'Socialism' as conceived by Karl Marx himself. In a socialist society, private property in the means of production would be headed by co-operative ownership. A socialist economy would not base production on the creation of private profits, but would instead base production and economic activity on the criteria of satisfying human needs, i.e., production would be carried out directly for use. Eventually, Socialism would give way to a Communism, i.e., a classless, stateless system

Frederick Engels

based on common ownership and free access and maximum freedom to individuals to develop their own capacities and talents.

After the death of Karl Marx a new democratic Socialism came to the scene. Bernstein did not agree with the principle of class struggle advocated by Marx. He said that socialist objectives should be achieved through democratic means.

The revolutionary movement in Russia overthrew the Czarist regime in 1917. Lenin and Trotsky were the chief organisers of this Communist Revolution. On October 1, 1949, a Communist regime was established in China under the leadership of Mao Tse-tung. In India too socialist parties emerged. In 1934 Socialist party was established in India under the leadership of Acharya Narendra Dev, Achyut Patwardhan,



Karl Marx

Lenin

3. History

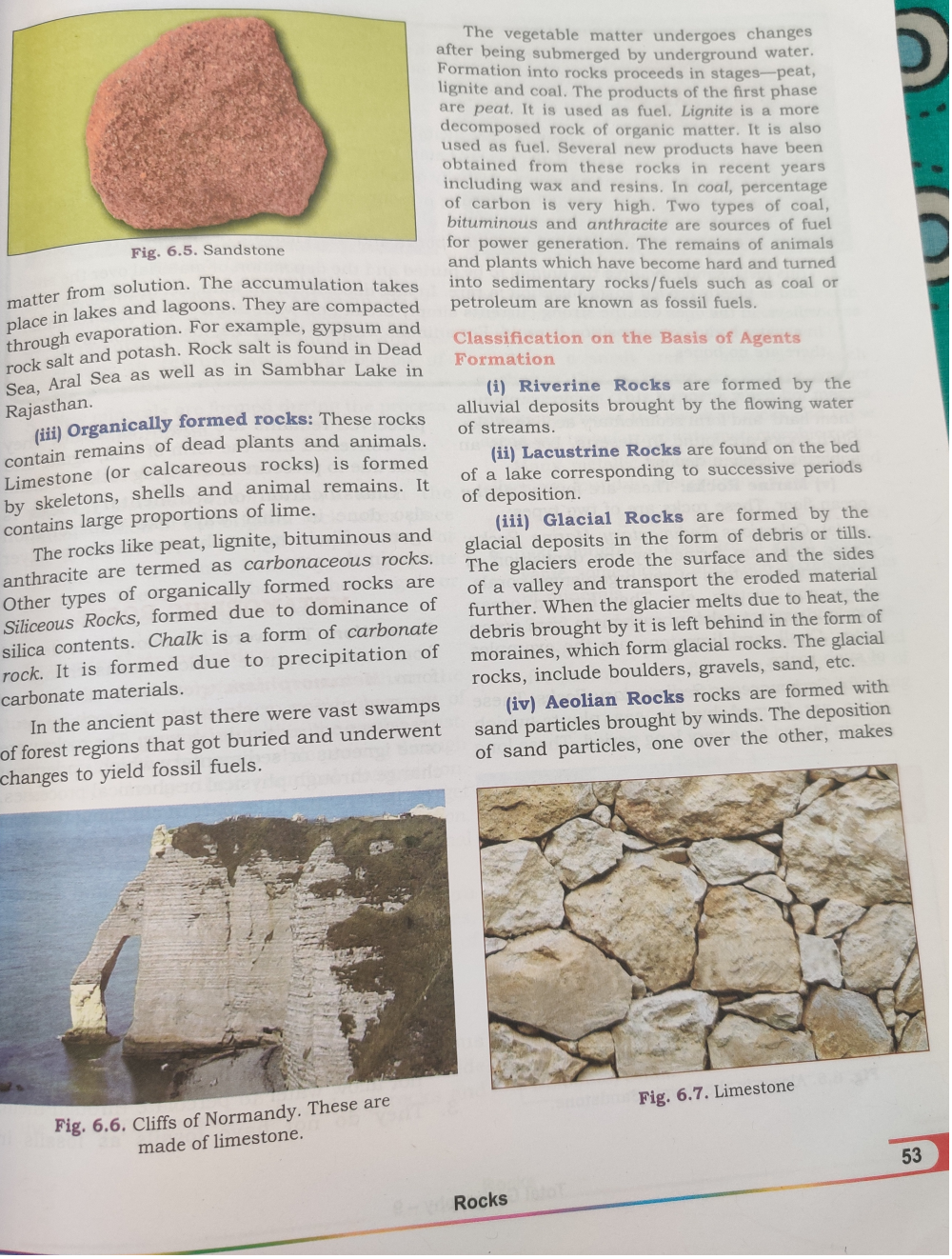


Fig. 6.5. Sandstone

matter from solution. The accumulation takes place in lakes and lagoons. They are compacted through evaporation. For example, gypsum and rock salt and potash. Rock salt is found in Dead Sea, Aral Sea as well as in Sambhar Lake in Rajasthan.

(iii) Organically formed rocks: These rocks contain remains of dead plants and animals. Limestone (or calcareous rocks) is formed by skeletons, shells and animal remains. It contains large proportions of lime.

The rocks like peat, lignite, bituminous and anthracite are termed as *carbonaceous rocks*. Other types of organically formed rocks are *Siliceous Rocks*, formed due to dominance of silica contents. *Chalk* is a form of *carbonate rock*. It is formed due to precipitation of carbonate materials.

In the ancient past there were vast swamps of forest regions that got buried and underwent changes to yield fossil fuels.

The vegetable matter undergoes changes after being submerged by underground water. Formation into rocks proceeds in stages—peat, lignite and coal. The products of the first phase are *peat*. It is used as fuel. *Lignite* is a more decomposed rock of organic matter. It is also used as fuel. Several new products have been obtained from these rocks in recent years including wax and resins. In *coal*, percentage of carbon is very high. Two types of coal, *bituminous* and *anthracite* are sources of fuel for power generation. The remains of animals and plants which have become hard and turned into sedimentary rocks/fuels such as coal or petroleum are known as fossil fuels.

Classification on the Basis of Agents Formation

(i) Riverine Rocks are formed by the alluvial deposits brought by the flowing water of streams.

(ii) Lacustrine Rocks are found on the bed of a lake corresponding to successive periods of deposition.

(iii) Glacial Rocks are formed by the glacial deposits in the form of debris or tills. The glaciers erode the surface and the sides of a valley and transport the eroded material further. When the glacier melts due to heat, the debris brought by it is left behind in the form of moraines, which form glacial rocks. The glacial rocks, include boulders, gravels, sand, etc.

(iv) Aeolian Rocks rocks are formed with sand particles brought by winds. The deposition of sand particles, one over the other, makes

Fig. 6.6. Cliffs of Normandy. These are made of limestone.

Fig. 6.7. Limestone

53

Rocks

4. Geography