



**Prize Winner**

# **Programming, Apps & Robotics Year 7-8**

**Ania Choi**

**St Peter's Girls' School**



# Leukemia Detector

## 1. Introduction

This report is to summarise the development of a web application called Leukemia Detector engineered by Artificial Intelligence (AI) to detect signs of leukemia in human blood cells.

## 2. Science Idea

My Auntie Connie was diagnosed with leukemia in 2022. She discovered her disease at a late stage and passed away in 2023. This bad news shook me and I researched a lot about the leukemia. I am wondering if modern technology could be used, her disease would have been diagnosed earlier and she would have received a timely treatment.

After reading "The Song of the Cell" and "The Emperor of All Maladies" written by Siddhartha Mukherjee, I was inspired and began my journey to develop the Leukemia Detector.

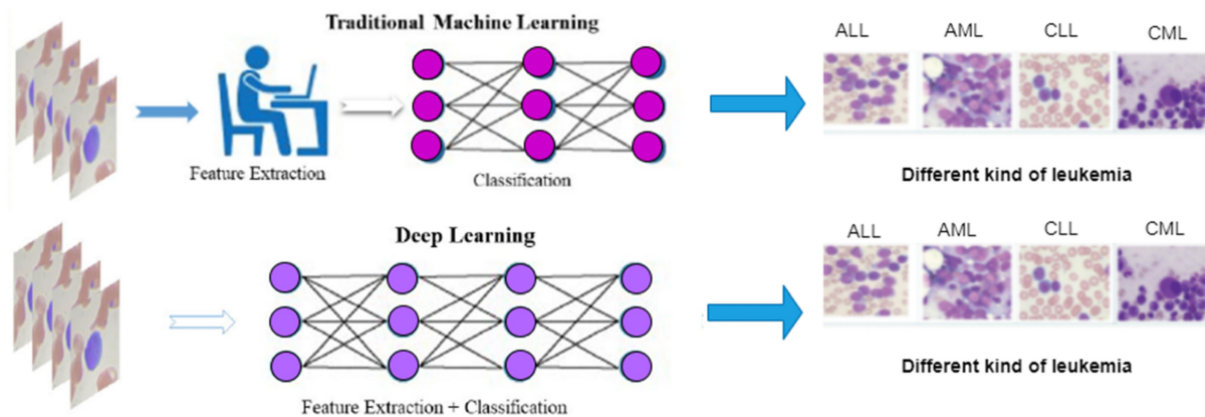
The main goal is to detect leukemia earlier using emerging technology such as AI. Digital photos of blood samples were used to develop a series of algorithms for Machine Learning to trace any abnormalities of blood cell images.

Hence, I develop Leukemia Detector which has ability to recognise the cancerous blood cells' image rapidly. This can greatly help scientists and oncologists to research leukemia to make the diagnosis effectively and patients can receive appropriate treatment in a timely manner.

## 3. Methodology

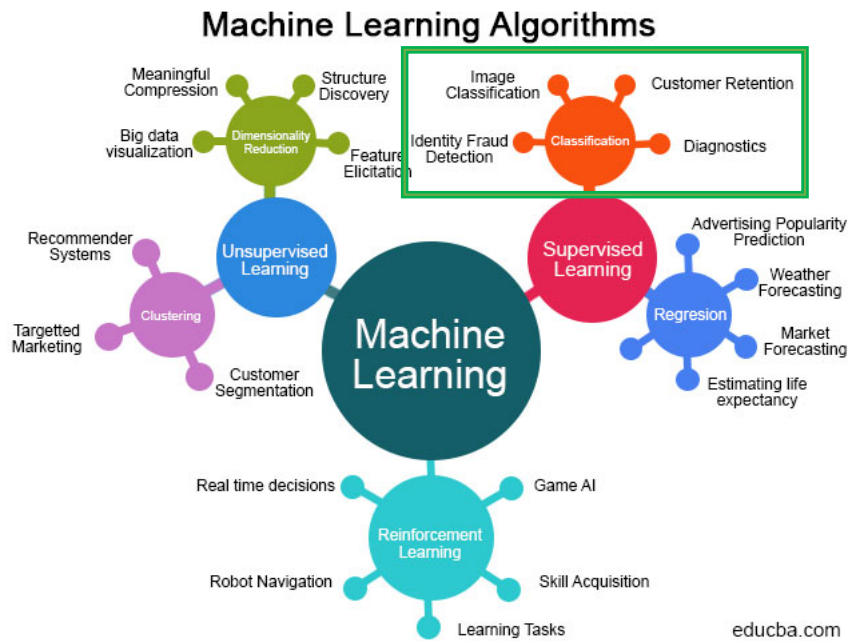
In developing Leukemia Detector, I make use of AI technology to analyse digital microscopic photos of blood samples from public domains. Since AI does not have any knowledge about leukemia prior to the instructions, I will train the AI engine by developing suitable algorithms and testing them with image data. The results are then generated through machine learning.

The AI algorithm is a wide area that consists of Machine Learning Algorithms and Deep Learning Algorithms. On a website of AI, Ridzuan<sup>9</sup> made a comparison between Deep Learning Algorithms and Machine Learning Algorithms. The website mentioned that Machine Learning Algorithms are algorithms that use large amounts of data for training computers. Deep Learning Algorithms are a branch of Machine Learning Algorithms that use a complex level of Artificial Neural Networks (ANNs) to complete the main process of Machine Learning and achieve high accuracy in tasks such as speech recognition, self-driving car and sentence generator. Though Machine Learning, AI can identify types of leukemia with training.



**Figure 1: Deep Learning vs Machine Learning**

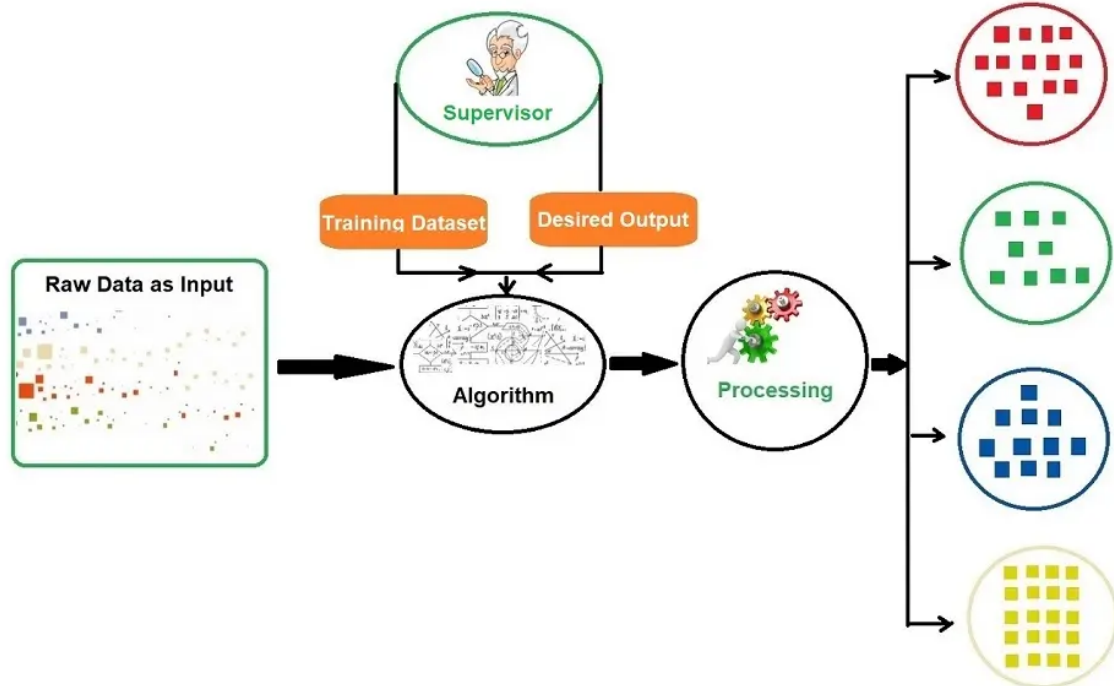
Machine Learning is divided into three sectors: Supervised Learning, Unsupervised Learning and Reinforcement Learning. Diagnosis is under the Supervised Learning section.



**Figure 2: Machine Learning categories**

Supervised Learning sorts input data into categories. Classifications use algorithms to accurately assign test data into different groups. The process of Supervised Learning is shown in the diagram below.

## Supervised Learning

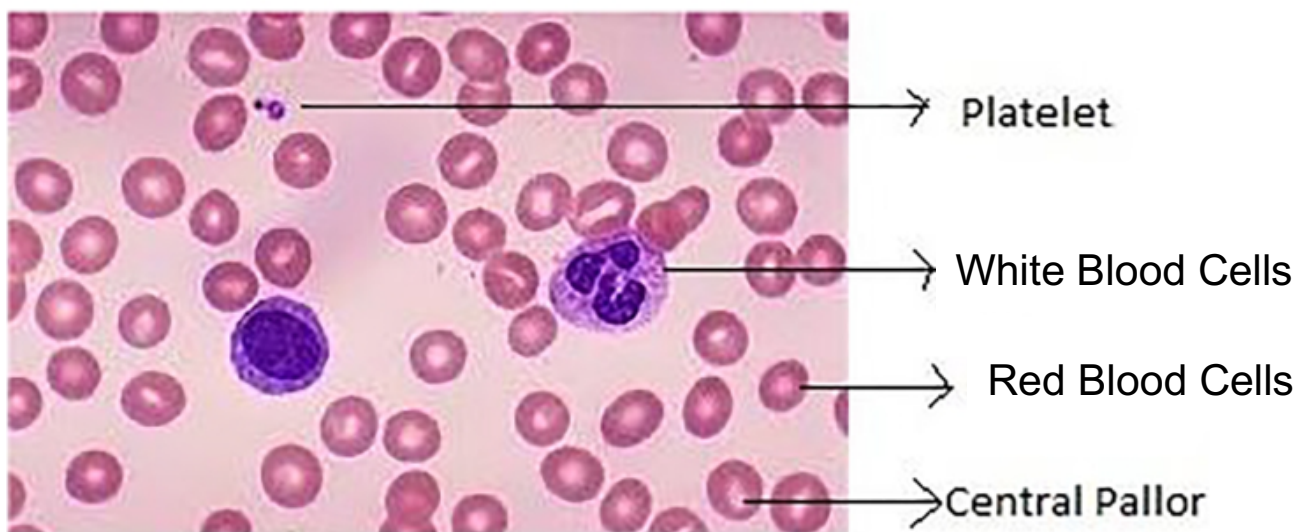


**Figure 3: Process of Supervised Learning**

The version of AI bot to power the Leukemia Detector is GPT-4o. It is more advanced and has more features than the previous versions. GPT-4o has ability to analyse photos and provide additional information about the questions being asked.

### 3.1. Blood Cell Biology

Prior to developing Leukemia Detector, I conducted literature review on human blood cell biology including types, function and how to define a healthy and normal cell. Three types of human blood cells are produced from the bone marrow. The function of each is listed below.

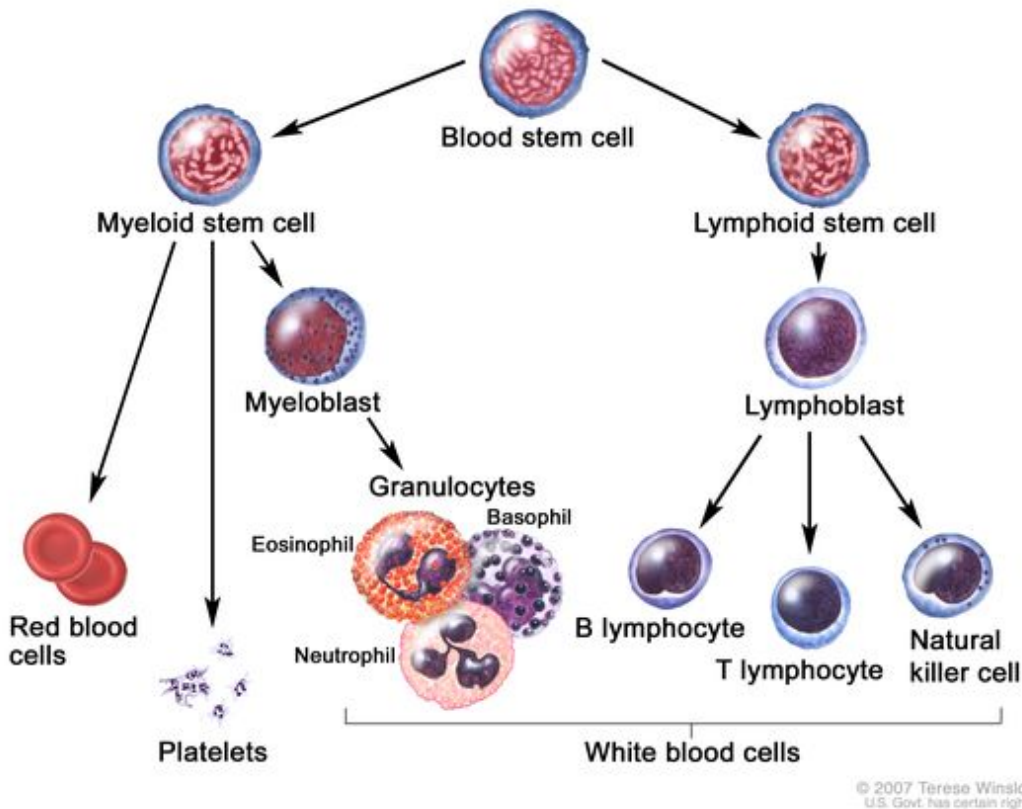


**Figure 4: Image of human blood cells**

Type	Functions
<b>Red Blood Cell</b>	<ul style="list-style-type: none"> <li>• Disk-like shape to have large surface area to volume ratio</li> <li>• Contains haemoglobin to carry oxygen</li> <li>• Contains no nucleus and mitochondria</li> <li>• Unable to reproduce by itself</li> </ul>
<b>White Blood Cell</b>	<ul style="list-style-type: none"> <li>• Defends the body from diseases by producing antibodies</li> <li>• Have different types of White Blood cells</li> </ul>
<b>Platelets</b>	<ul style="list-style-type: none"> <li>• Aid clotting of blood at the site of wound</li> <li>• Turns soluble fibrinogen into insoluble fibrin</li> </ul>

**Table 1: Types of human blood cells**

White blood cells defend the body from diseases and infection. They are made in the bone marrow and are found in lymph nodes and blood. In the bone marrow, white blood cells develop from blood stem cells and they continue to form into different types of white blood cells. The types of white blood cells include granulocytes (neutrophils, eosophils and basophils), monocytes and lymphocytes (T-cells and B-cells). Below is a diagram of how white blood cells form.



© 2007 Terese Winslow  
U.S. Govt. has certain rights

**Figure 5: The development of white blood cells**

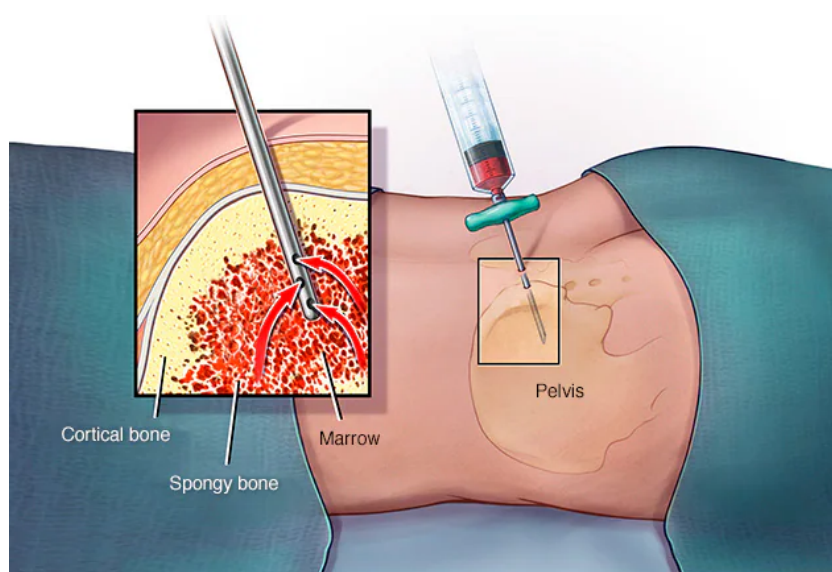
The role of white blood cells is to defend the body against infections. Different types of white blood cells have diverse methods to attack the infection and disease. Some white blood cells are directly attracted towards infection agents and engulf them. When the infection agents are in the cell, the white blood cells release chemicals that destroy the organism. Other white blood cells such as lymphocytes attacks the intruder by making antibodies. They detect antigens which are proteins on the surface of pathogens. These antigens are different from the body's cells. Once the lymphocytes detect antigens that do not belong to the body, they produce antibodies to attack the antigens.

### 3.2. Leukemia

The blood cells are made in the bone marrow and gone through various stages to reach their fully mature forms. Leukemia is known as blood cancer. It begins in the bone marrow. It arises from abnormal and immature white blood cells developed rapidly and uncontrolledly. In most cases, leukemia is developing (immature) white blood cells with no functions as normal cells. If someone has leukemia, the leukemia cells would take over space inside the bone marrow and there is no space for cells to develop into healthy different types of blood cells. The two most common types of leukemia are myeloid and lymphocytic. Some types of leukemia are more common in children while some types are more common in adults or elderly.

### 3.3. Tests of Leukemia

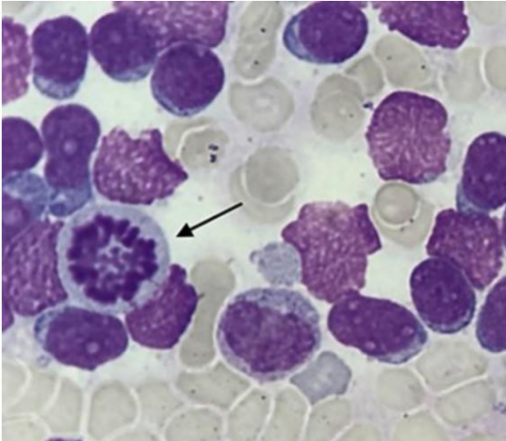
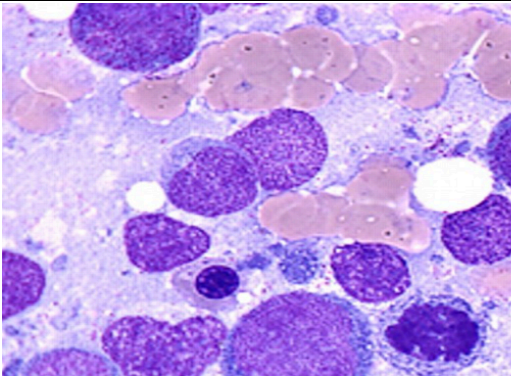
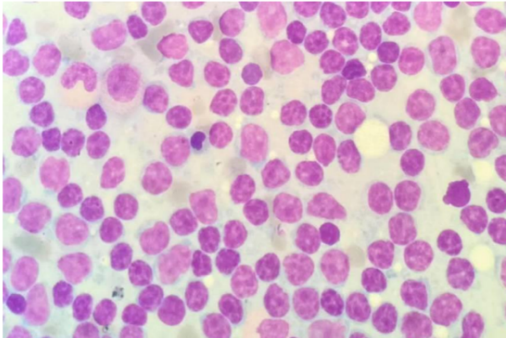
There are several traditional tests to detect leukemia. The basic tests are Complete Blood Count (CBC) and bone marrow biopsy. The doctor would insert a needle at the base of the spinal cord and capture a small piece of sponge-like tissue from the marrow. The process is shown below.

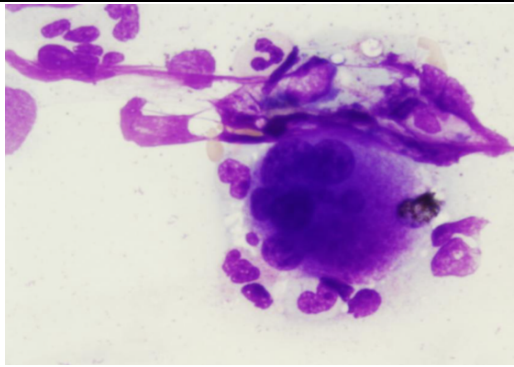
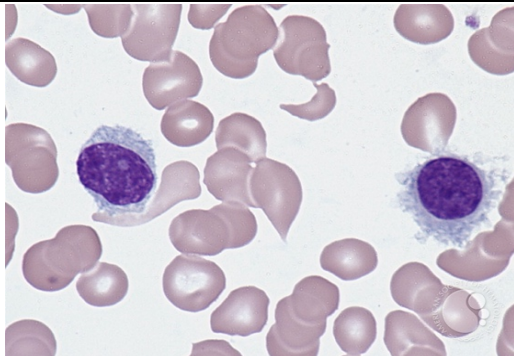


**Figure 6: Bone marrow biopsy procedures**

The Cytogenetic test is used to identify the Philadelphia chromosome. They also look for other changes known as additional chromosomal abnormalities (ACA) which are changes in size, shape and number in chromosomes. Another type of test is the fluorescence in situ hybridisation (FISH) test. Doctors dye cells in a fluorescent dye to look for the BCR-ABL1 gene on the chromosomes. FISH tests are generally faster to generate results than Cytogenetic tests. However, the waiting time for the results are about a month and there may be some uncertainties that need further testing to confirm.

Based on the literature review on the most common types of leukemia, I categorise each of the most common types of leukemia as criteria to form the basis of the algorithms. Samples are tabulated below.

#	Types	Samples of Blood Cells Images	Signs of Disease
1	Acute Lymphoblastic Leukemia (ALL)		<ul style="list-style-type: none"> <li>• Develop in immature white blood cells</li> <li>• Targets lymphocytes white blood cells</li> <li>• More white blood cells than other blood cells (over production of white blood cells)</li> </ul>
2	Acute Myeloid Leukemia (AML)		<ul style="list-style-type: none"> <li>• Develops in myeloid cells</li> <li>• Overproduction of white blood cells</li> <li>• Develop in immature white blood cells</li> </ul>
3	Chronic Lymphoblastic Leukemia (CLL)		<ul style="list-style-type: none"> <li>• Mature blood cells</li> <li>• Disease worsens over time (weaker body and immune system)</li> <li>• Begins in B lymphocytes</li> </ul>

#	Types	Samples of Blood Cells Images	Signs of Disease
4	Chronic Myeloid Leukemia (CML)		<ul style="list-style-type: none"> <li>• Large increase in granulocytes</li> <li>• Appears in mature cells</li> <li>• More white blood cell than other blood cells</li> </ul>
5	Hairy Cell Leukemia (HCL)		<ul style="list-style-type: none"> <li>• Hairy appearance</li> <li>• Originates from B lymphocyte</li> <li>• Often develop in mature cells</li> </ul>

**Table 2: Criteria to sort out leukemia types**

### 3.4. Programming Requirements

Leukemia Detector is developed based on the following programming requirements.

- Programming language: Python
- Customised components: Gradio
- AI engine: Openai
- Editor: Mu Editor
- Image source: ASH Image Bank (<https://imagebank.hematology.org/>)

I use Python and Mu Editor because it is easy to understand and systematic. Python is a programming language and Mu Editor is an editor that assists me to write my code. Openai is a research laboratory to develop and direct AI which has been trained on massive data from a variety of sources. Such that the responses generated can be more accurate in response to users' input in a personalised and automated way.

Gradio is a Python package that has customizable components. It is easy to use to demo Machine Learning using a few lines of code. Similar to the use of Openai, I used Gradio to develop a function to generate various responses. I used these programming languages and platforms to help me create the Leukemia Detector.



In ASH Image Bank, there are many photos of various types of leukemia. I collected about 100 images and used these photos to train and test the AI engine. Sometimes, the AI engine may mistakenly recognise a type of leukemia from another one. I need to correct the AI engine and give the feedback which are accumulated in the system. Such that, the AI engine would not make the same mistake when it comes across the type of leukemia again.

### 3.5. Flowchart

The following flow chart is based on the criteria that was mentioned above. The purpose of the flow chart is to develop algorithms to train and test the AI.

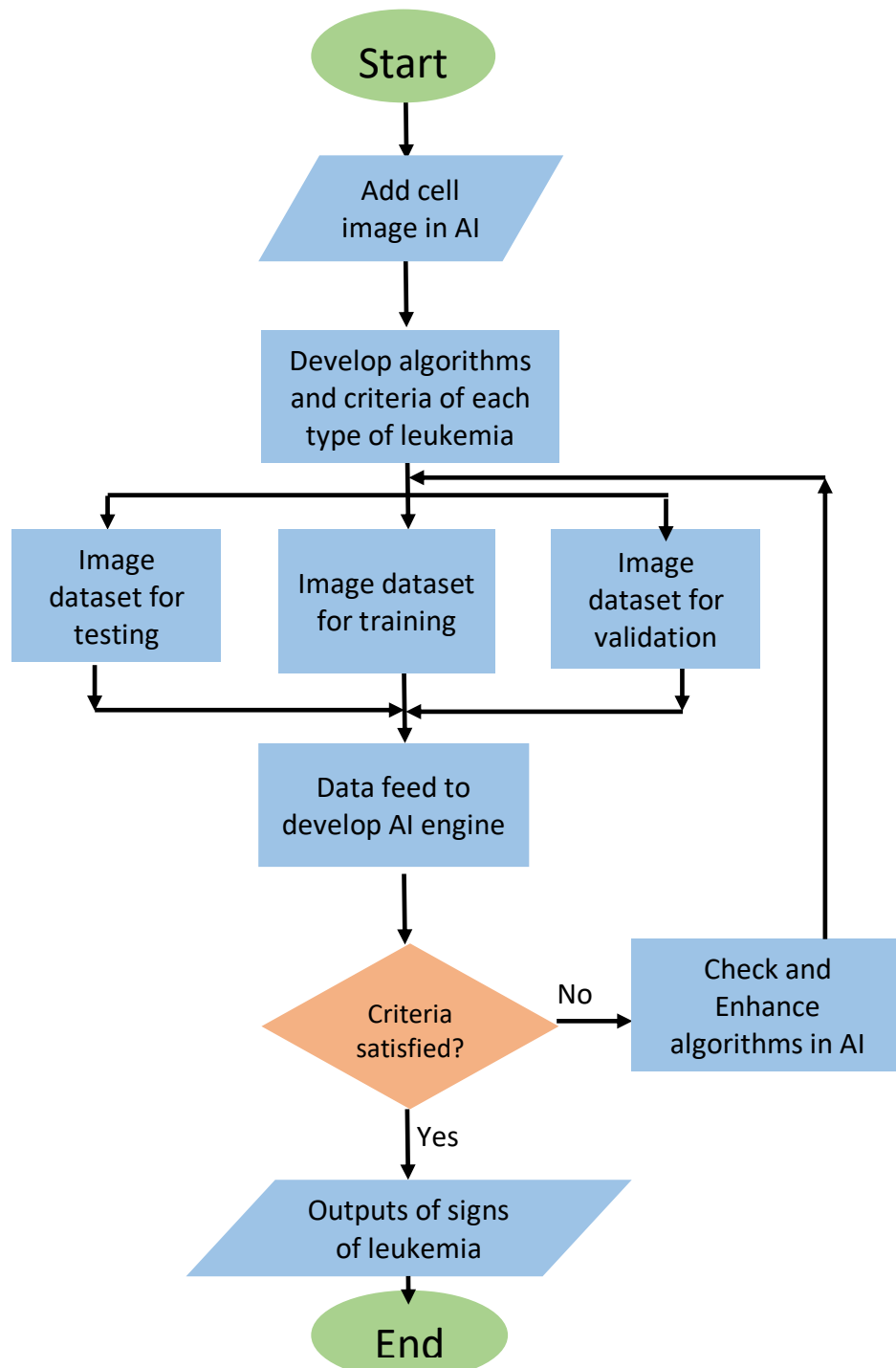
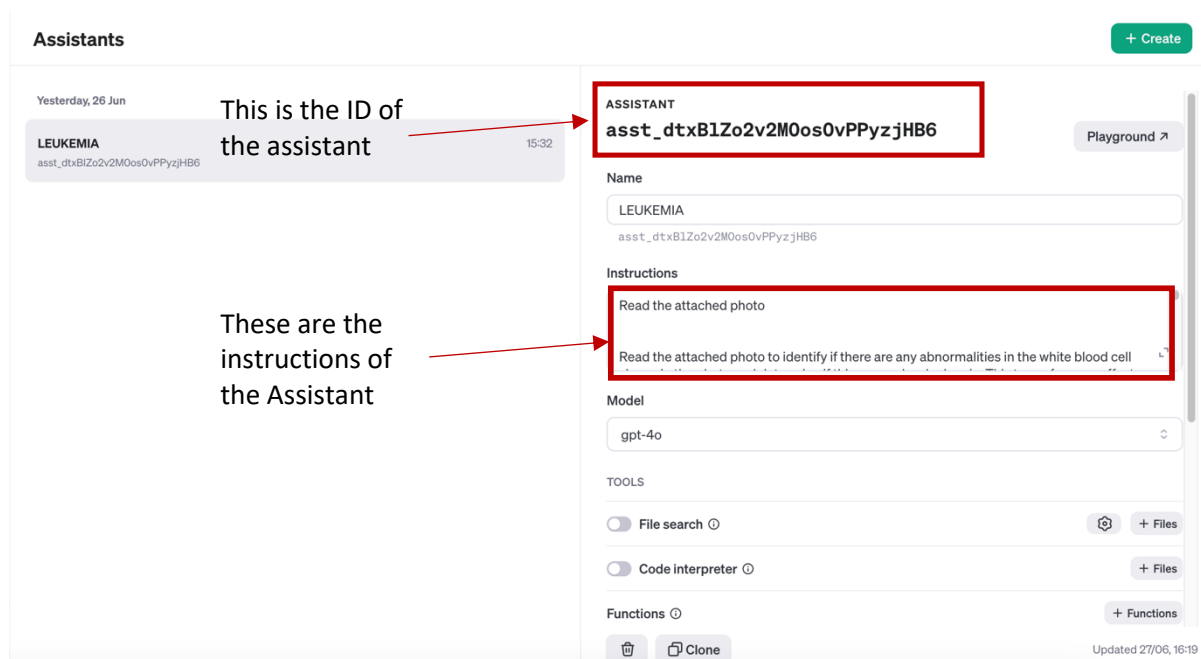


Figure 7: Flowchart of the algorithm of Leukemia Detector

### 3.6. Programming

A hard copy of the programme of Magic Herbot is attached in Appendix A and detailed explanations are found in this section. The steps to develop Magic Herbot are shown below with annotated screenshots.

#### Step 1: Write the instructions



In Openai, there are multiples models, namely Chat, Assistant and Completion. The chat function provides text output in response text input. Completion models aim to answer complex questions and are suitable for businesses. Assistant models are user-friendly while providing accurate information.

The instructions are used to develop a set of algorithms for the AI engine to follow and generate results. AI algorithms work by taking in training data to help the algorithm to gain experience. The model that Leukemia Detector is using is the Assistant model. They obey rules and use natural language processing (NLP) and machine learning to fulfil the user's requests. Samples of the criteria that the Assistant follows are shown below.

## Sample A

Read the attached photo

Read the attached photo to identify if there are any abnormalities in the white blood cell shown in the photo and determine if this person has leukemia. This type of cancer affects myeloid and lymphoid cells. If the shape of the inner part (dark purple) of the cell is abnormal, this may be a sign of leukemia. There are a few features of an abnormal white blood cell:

1. Auer rods (appears in myeloid cells--Auer rods are distinctive needle like crystals that are specific for and virtually diagnostic of a myeloid (bone marrow-derived) neoplastic clone)
2. Blast cells that are not able to mature (Acute Lymphoblastic leukemia)

Leukemia can be split into four categories; acute lymphoblastic leukemia, acute myeloid leukemia, chronic lymphoblastic leukemia and chronic myeloid leukemia. Lymphoblastic and myeloid leukemia can be either acute or chronic. Acute leukemia is usually more invasive and faster; it is often found in children and teenagers. Chronic leukemia targets people aged over 60 and develops slower

## Sample B

### Acute Myeloid Leukemia

Healthy blood cells, like red and white cells and platelets, are formed by normal stem cells. AML develops when the genetic material of a developing stem cell in the bone marrow is damaged, resulting in an "acquired mutation." The damaged cell becomes a leukemic cell and multiplies into billions of cells called leukemic blasts. These leukemic blasts do not function normally.

- Block the production of normal cells
- Grow and survive better than normal cells

The following conditions are the results of AML.

1. Anaemia: a condition where there is a low number of red cells in the blood, which can cause fatigue and shortness of breath
2. Neutropenia: a condition where there is a low number of neutrophils (a type of white blood cell), so the immune system cannot effectively guard against infection.
3. Thrombocytopenia: a condition where there is a low number of platelets, which can cause bleeding and easy bruising with no apparent cause

## Step 2: Import libraries to be used in the programming

```
1 import gradio as gr
2 import os
3 import openai
4
5 # Initialize the client
6 client = openai.Client(
7     api_key=
8 )
9
10 mv_assistant = client.beta.assistants.retrieve("asst_dtxB1Zo2v2M0os0vPPvziHB6")
11
```

Import Gradio library for web app framework

Import the Openai library to generate responses

Define the assistant's name and ID

Apply Openai API key to use the AI function in this web app

## Step 3: Convert the uploaded image into 64bit for AI to read

```
16
17
18 def file_to_base64(file_path):==
19     with open(file_path, "rb") as file:
20         data = file.read()
21         return base64.b64encode(data).decode("utf-8")
22
```

Detect the image and convert it to 64 bit file for the computer to read

## Step 4: Add message to chatbot space

```
23
24 def add_message(history, message):
25     for x in message["files"]:
26         history.append((x, None))
27
28     if message["text"] is not None:
29         history.append((message["text"], None))
30     return history, gr.MultimodalTextbox(value=None, interactive=False)
31
32
```

Adds the user's message into the replying space. If the code detects a message (in blue), it will send it to the space.

## Step 5: Detect the information that the user gives (photo and text) and reply

```
32
33 def bot(history):
34     img = history[-2][0]
35     if len(history) > 1 else ""
36     msg = None
37     if isinstance(img, tuple):
38         file = client.files.create(file=open(img[0], "rb"), purpose="vision")
39         if history[-1][0] == "":
40             msg = {
41                 "role": "user",
42                 "content": [
43                     {"type": "image_file", "image_file": {"file_id": file.id}},
44                 ],
45             }
46     else:
47         msg = {
48             "role": "user",
49             "content": [
50                 {"type": "text", "text": history[-1][0]},
51                 {"type": "image_file", "image_file": {"file_id": file.id}},
52             ],
53
```

Looking for input in the chatbot space

If there is input, check if there is an image

Check the last input if there is no text (no text=only image)

If there is text, it means that there is both text and image.

## Step 6: Command for running the Openai

```
55
56     if msg is not None:
57         thread = client.beta.threads.create(messages=[msg])
58         run = client.beta.threads.runs.create(
59             thread_id=thread.id, assistant_id=my_assistant.id
60         )
61
62     while run.status != "completed":
63         run = client.beta.threads.runs.retrieve(thread_id=thread.id, run_id=run.id)
64
65
66
67     thread_messages = client.beta.threads.messages.list(thread_id)
68     # print(thread_messages)
69     reply = thread_messages.data[0].content[0].text.value
70     history[-1][1] = reply
71     return history
```

If there is a message, create job for Openai

Run the job for Openai

## Step 7: Create the interface using Gradio

```
73
74 with gr.Blocks(fill_height=True) as demo:
75     gr.Label(value="***** Leukemia Detector *****")
76     chatbot = gr.Chatbot(
77         elem_id="chatbot",
78         bubble_full_width=False,
79         scale=1,
80     )
81
82     chat_input = gr.MultimodalTextbox(
83         interactive=True,
84         placeholder="Please upload your blood sample image and describe your case and the cells...",
85         show_label=False,
86     )
87
88     chat_msg = chat_input.submit(
89         add_message, [chatbot, chat_input], [chatbot, chat_input]
90     )
91     bot_msg = chat_msg.then(bot, chatbot, chatbot, api_name="bot_response")
92     bot_msg.then(lambda: gr.MultimodalTextbox(interactive=True), None, [chat_input])
93
94
95     demo.launch(share=True)
96
```

Creating the features of the label

Instructions of what to input in the chatbot (user)

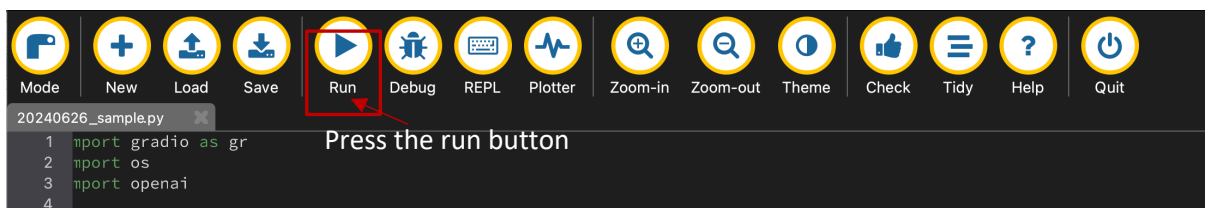
After the user replies, run Step 5

Create link to open in web browser

## 4. Results

The steps listed below are to execute Leukemia Detector to generate results.

### Step 1: Run the code



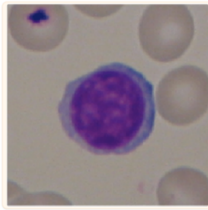
### Step 2: Copy and paste the link into the web browser

```
Running: 20240626_sample.py
Running on local URL: http://127.0.0.1:7860
Running on public URL: https://e6771dd70131a41916.gradio.live

This share link expires in 72 hours. For free permanent hosting and GPU upgrades, run `gradio deploy` from Terminal to deploy to Spaces (https://huggingface.co/spaces)
>>>
```



### Step 3: Upload an image and describe the case and the cells



is there a problem with this patient? The patient is a six year old boy.

### Step 4: Generate the result

After the user submits the image and describes the problem, the AI will analyse the image using the instructions. A sample of the result is shown below.

Label \*\*\*\*\* Leukemia Detector \*\*\*\*\*

Chatbot

is there problem in this image

Yes, there is a problem in the image. The photo shows an abnormally high number of white blood cells compared to red blood cells, and many of the white blood cells appear to be immature (indicated by the "purple spots" not filling the whole cell). This combination is a characteristic of Acute Lymphoblastic Leukemia (ALL).

To summarize:

1. There are more white blood cells than red blood cells.
2. The purple area (presumably the nucleus) does not fill the entire cell, indicating these cells are immature.
3. The immature cells point toward the overproduction of lymphoblasts typical in ALL.

While this image suggests ALL based on the criteria provided, a comprehensive diagnosis would still require additional laboratory testing and clinical evaluation, including a bone marrow biopsy, chromosome analysis, and flow cytometry to distinguish between different types of leukemia.

Please upload your blood sample image and describe your case and the cells...



About 100 images of blood cells were used to train the AI engine and test the algorithms. About 75% of images with signs of leukemia were detected and recognised correctly. Through these results, Leukemia Detector shows promising results in identifying leukemia cells and categorising them in types. However, it is noted that the AI engine experienced difficulty when differentiating CLL and CML. This may be because the AI engine may have mistakenly recognised lymphoblasts for myeloid cells.

#### **4.1. Applications**

Leukemia Detector has potential applications for providing diagnosis to patients and making a rapid decision about what type of leukemia the patient has. Once the disease is detected earlier, the patient takes action swiftly to prevent it from getting worse. Another potential application is to combine clinic monitoring apps with Leukemia Detector. In such manner, the patient may know the process of diagnostics and ask the doctor about their disease. The layout is user-friendly and can be used by clinic officers who do not have medical knowledge.

The layout of the chatbot is simple and users can easily interpret how to use it. On each of the chatbot, there are some symbols that allows user to upload photos. Additionally, there are instruction for what to ask the AI. The patients would know what to ask the chatbot and receive accurate results. However, the aesthetics are not as appealing as the background colour is only light/dark (depending on day or night) and there are no decorative borders.

In addition to being a user-friendly web application, Leukemia Detector generates the results quickly and requires minimum cost. While most traditional tests require almost a few weeks to detect leukemia, Leukemia Detector needs short period of time to complete the image analysis. Leukemia Detector can help scientists and oncologists to carry out in-depth researches on the patterns and features of leukemia in an effective and efficient way.

#### **4.2. Evaluation**

While developing Leukemia Detector, my first attempt was to create a multi-functional chatbot that can also generate microscopic images of leukemia blood cells. However, due to my limited time and knowledge, I could not incorporate this feature into Leukemia Detector.

Given the above, the data from Openai cannot be verified. Hence, sometimes, it does not respond accordingly. The current version of Leukemia Detector uses blood sample photos to recognise the types of leukemia. However, the AI chatbot may confuse one type of leukemia with another type. For example, AI may confuse ALL with CLL because the slowly progressing disease of the chronic leukemia cannot be seen through one photo. The AI's responses are sometimes unreliable, so I plan to improve its accuracy by providing more information about how to distinguish similar types of leukemia.

When the diagnoses of the leukemia are incorrect, the AI does not input this into its instructions which makes the results unreliable. This can be improved by gathering more data and image information and allowing Openai to access web sources to expand its criteria and write an enhanced algorithm. Additional abilities can be added to the web application, such as suggesting treatment to the particular type of leukemia depending on its aggressiveness.



Relying on image analysis is not sufficient in diagnosing leukemia. To increase the accuracy of the results, image analysis should be combined with other methods and algorithms such as blood cell counts and genetic testing. Since Leukemia Detector diagnoses leukemia based on high-quality images, some images that are not clear may affect the result. Hence, more microscopic images would be collected to present a robust system that generates more reliable results.

## 5. Conclusion

My science idea is to develop a web application, Leukemia Deterctor to effectively detect signs of leukemia at earlier stage. Leukemia Detector engineered by AI is able to give a diagnosis to patients for leukemia and greatly help scientists and oncologists to do research in Leukemia more effectively. I will further improve Leukemia Detector to make it more robust and user-friendly.

## 6. Reference List

1. Siddhartha Mukherjee. (2023, December 5). The Song of the Cell.
2. Siddhartha Mukherjee. (2010, November 16). The Emperor of All Maladies.
3. Gradio. Available at: <https://gradio.app/> (Accessed on 20 March 2024)
4. Openai. Available at: <https://openai.com/> (Accessed on 19 March – 28 June 2024)
5. Hugging Face. Available at: <https://huggingface.co/> (Accessed 20 March 2024)
6. ASH Image Bank Available at: <https://imagebank.hematology.org/> (Accessed 1 April- 28 June 2024)
7. <https://www.tableau.com/data-insights/ai/algorithms#:~:text=So%20then%20what%20is%20an,to%20operate%20on%20its%20own.> (Accessed 21 March 2024)
8. <https://www.educba.com/artificial-intelligence-algorithm/> (Accessed 21 March 2024)
9. <https://deeplearningmy.com/deep-learning-algorithms-comparison/> (Accessed 22 March 2024)
10. <https://www.machinelearningnuggets.com/gradio-tutorial/#:~:text=Gradio%20is%20an%20open%2Dsource,as%20a%20link%20with%20Anyone.> (Accessed on 23 March 2024)
11. <https://www.techtarget.com/searchenterpriseai/definition/OpenAI> (Accessed on 23 March 2024)

12. <https://www.hematology.org/education/patients/blood-cancers/leukemia> (Accessed on 26 March 2024)
13. <https://www.mayoclinic.org/diseases-conditions/acute-lymphocytic-leukemia/symptoms-causes/syc-20369077> (Accessed on 24 March 2024)
14. <https://www.cancerresearchuk.org/about-cancer/chronic-myeloid-leukaemia-cml/getting-diagnosed/genetic-tests - :~:text=Doctors use the FISH test,assess your response to treatment.> (Accessed on 27 March 2024)
15. <https://www.cancer.org/cancer/types/chronic-lymphocytic-leukemia/about/what-is-ll.html> (Accessed 30 March 2024)
16. <https://www.mayoclinic.org/diseases-conditions/chronic-myelogenous-leukemia/symptoms-causes/syc-20352417> (Accessed 1 April 2024)
17. <https://www.blood.co.uk/news-and-campaigns/the-donor/latest-stories/functions-of-blood-its-role-in-the-immune-system/ - :~:text=They are attracted towards the,organism down and destroy it.> (Accessed 1 June 2024)
18. <https://veganhealth.org/white-blood-cells-in-vegans/> (Accessed 1 June 2024)
19. [https://www.researchgate.net/figure/Normal-white-blood-cells-The-cells-seen-in-the-background-are-red-blood-cells-identified\\_fig1\\_360195740](https://www.researchgate.net/figure/Normal-white-blood-cells-The-cells-seen-in-the-background-are-red-blood-cells-identified_fig1_360195740) (Accessed on 1 June 2024)
20. <https://histology.siu.edu/intro/bldcells.htm> (Access on 15 June 2024)
21. <https://www3.med.unipmn.it/did/will/atlashem/node1.htm> (Access on 15 June 2024)

## 7. Acknowledgement

Beloved Auntie Connie and family  
Mr G A Coulter – Digital Technologies Teacher  
Ms Lene Van Schalkwyk- Scientist in Residence

## **Appendix A**

### **Programme of Leukemia Detector**

```
1 import gradio as gr
2 import os
3 import openai
4
5
6 # Initialize the client
7 client = openai.Client(
8     api_key="
9 )
10 my_assistant = client.beta.assistants.retrieve("asst_dtxBlZo2v2M0os0vPPyzjHB6")
11
12
13 def file_to_base64(file_path):
14     with open(file_path, "rb") as file:
15         data = file.read()
16         return base64.b64encode(data).decode("utf-8")
17
18
19 def add_message(history, message):
20     for x in message["files"]:
21         history.append((x, None))
22
23     if message["text"] is not None:
24         history.append((message["text"], None))
25     return history, gr.MultimodalTextbox(value=None, interactive=False)
26
27
28 def bot(history):
29     img = history[-2][0] if len(history) > 1 else ""
30     msg = None
31     if isinstance(img, tuple):
32         file = client.files.create(file=open(img[0], "rb"), purpose="vision")
33         if history[-1][0] == "":
34             msg = {
35                 "role": "user",
36                 "content": [
37                     {"type": "image_file", "image_file": {"file_id": file.id}},
38                 ],
39             }
40     else:
```



The image shows a code editor window with a toolbar at the top. The toolbar contains icons for Mode, New, Load, Save, Run, Debug, REPL, Plotter, Zoom-in, Zoom-out, Theme, Check, Tidy, Help, and Quit. Below the toolbar, the code editor displays Python code for a chatbot interface. The code includes logic for handling messages, creating threads, and updating the chat history. The code is as follows:

```
40     else:
41         msg = {
42             "role": "user",
43             "content": [
44                 {"type": "text", "text": history[-1][0]},
45                 {"type": "image_file", "image_file": {"file_id": file.id}},
46             ],
47         }
48     elif history[-1][0] != "":
49         msg = {
50             "role": "user",
51             "content": [
52                 {"type": "text", "text": history[-1][0]},
53             ],
54         }
55
56     if msg is not None:
57         thread = client.beta.threads.create(messages=[msg])
58         run = client.beta.threads.runs.create(
59             thread_id=thread.id, assistant_id=my_assistant.id
60         )
61
62         while run.status != "completed":
63             run = client.beta.threads.runs.retrieve(thread_id=thread.id, run_id=run.id)
64
65
66
67         thread_messages = client.beta.threads.messages.list(thread.id)
68         # print(thread_messages)
69         reply = thread_messages.data[0].content[0].text.value
70         history[-1][1] = reply
71     return history
72
73
74 with gr.Blocks(fill_height=True) as demo:
75     gr.Label(value="***** Leukemia Detector *****")
76     chatbot = gr.Chatbot(
77         elem_id="chatbot",
78         bubble_full_width=False,
79         scale=1,
80     )
```

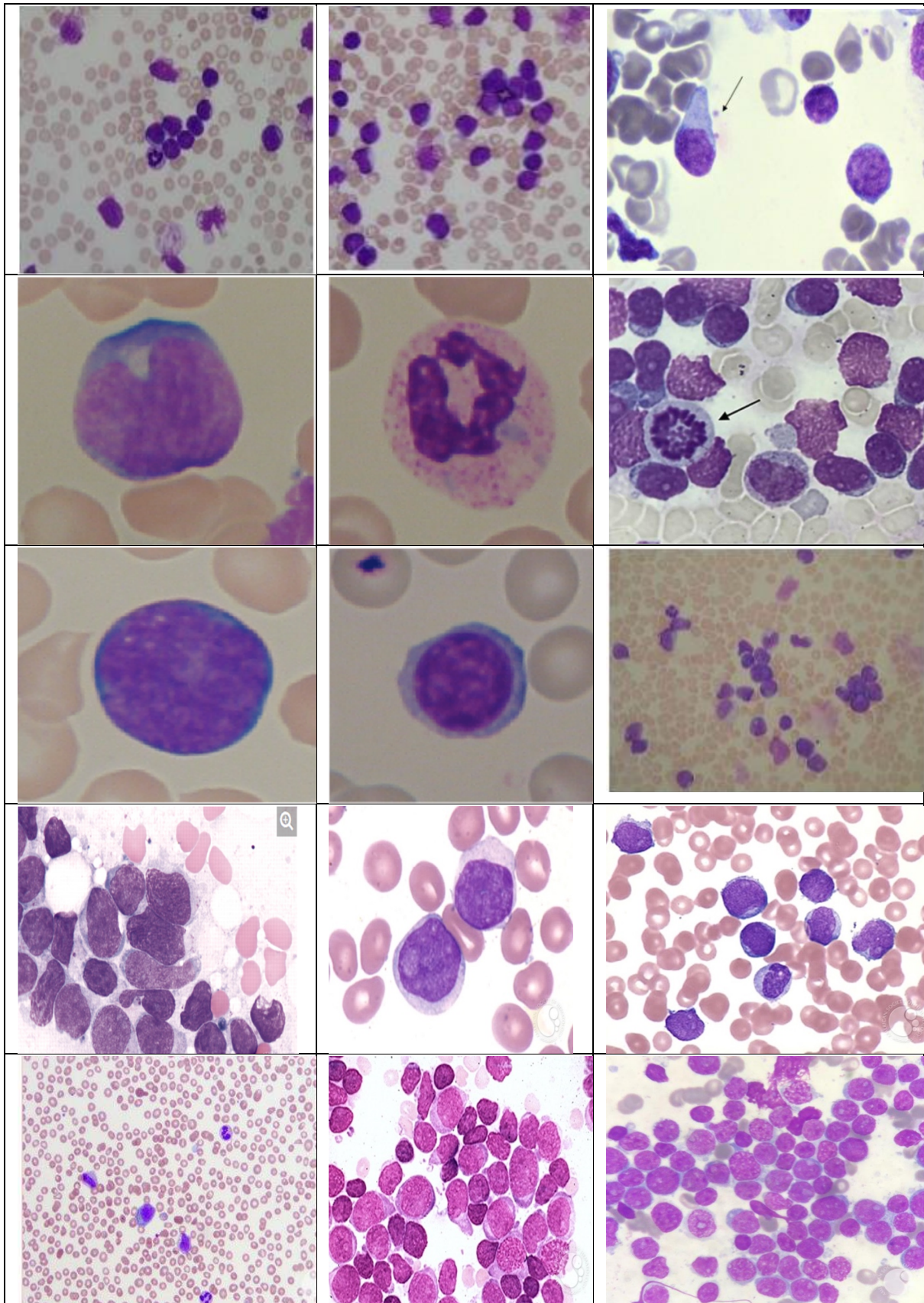


```
56     if msg is not None:
57         thread = client.beta.threads.create(messages=[msg])
58         run = client.beta.threads.runs.create(
59             thread_id=thread.id, assistant_id=my_assistant.id
60         )
61
62         while run.status != "completed":
63             run = client.beta.threads.runs.retrieve(thread_id=thread.id, run_id=run.id)
64
65
66
67         thread_messages = client.beta.threads.messages.list(thread.id)
68         # print(thread_messages)
69         reply = thread_messages.data[0].content[0].text.value
70         history[-1][1] = reply
71     return history
72
73
74 with gr.Blocks(fill_height=True) as demo:
75     gr.Label(value="***** Leukemia Detector *****")
76     chatbot = gr.Chatbot(
77         elem_id="chatbot",
78         bubble_full_width=False,
79         scale=1,
80     )
81
82     chat_input = gr.MultimodalTextbox(
83         interactive=True,
84         placeholder="Please upload your blood sample image and describe your case and the cells...",
85         show_label=False,
86     )
87
88     chat_msg = chat_input.submit(
89         add_message, [chatbot, chat_input], [chatbot, chat_input]
90     )
91     bot_msg = chat_msg.then(bot, chatbot, chatbot, api_name="bot_response")
92     bot_msg.then(lambda: gr.MultimodalTextbox(interactive=True), None, [chat_input])
93
94
95 demo.launch(share=True)
```

## **Appendix B**

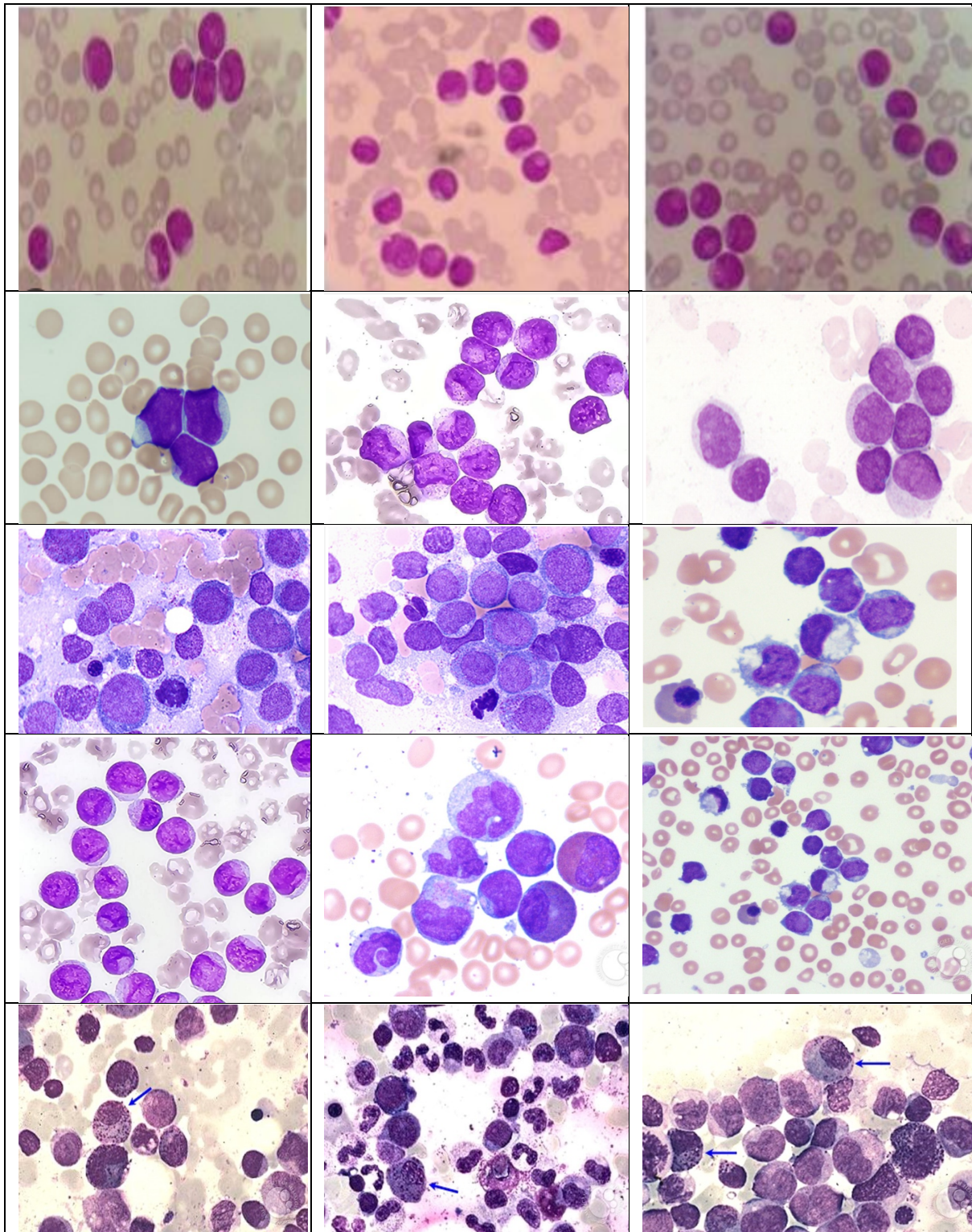
### **Images Samples of Blood Cells**

**Images Samples of Acute Lymphocytic Leukemia**

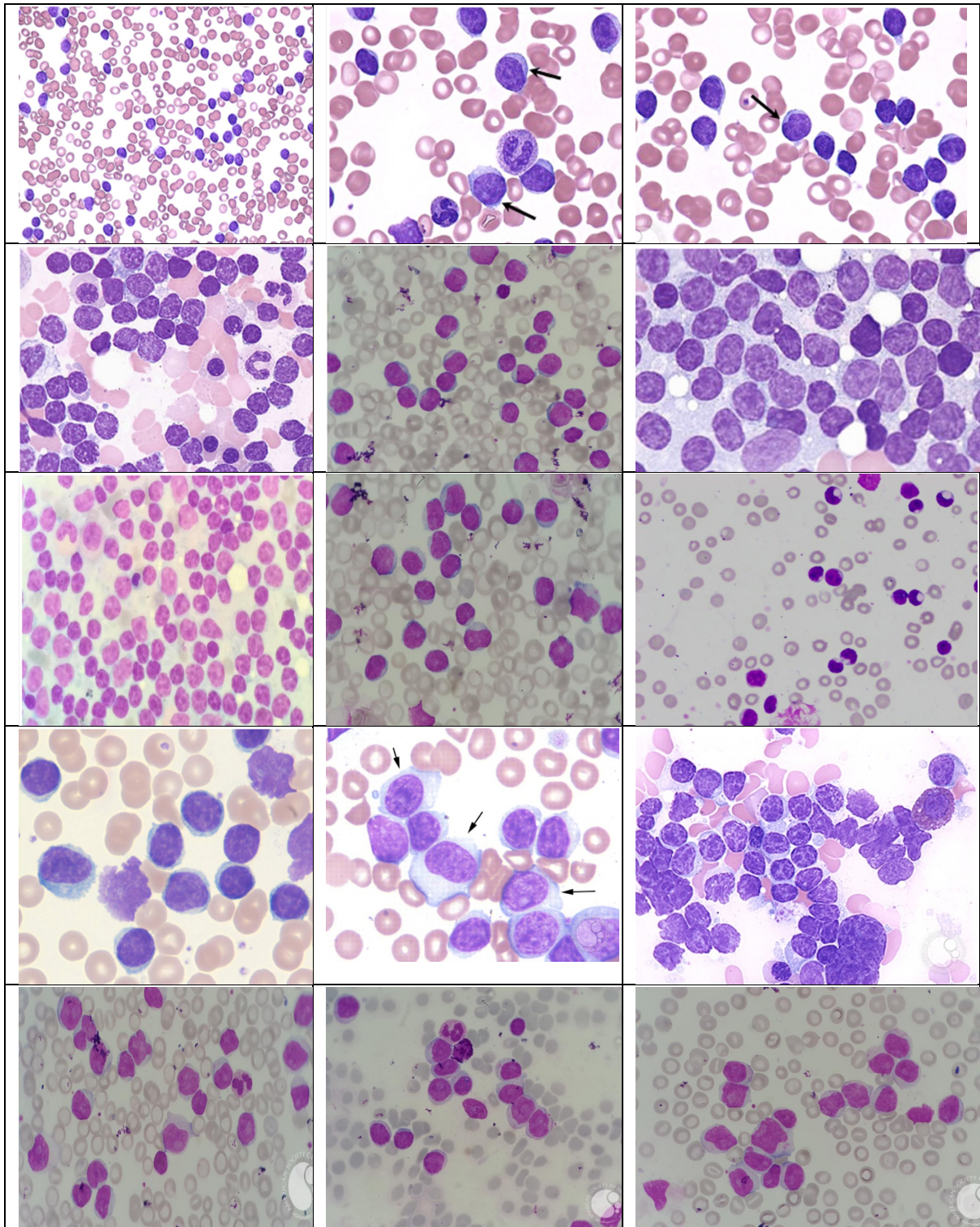




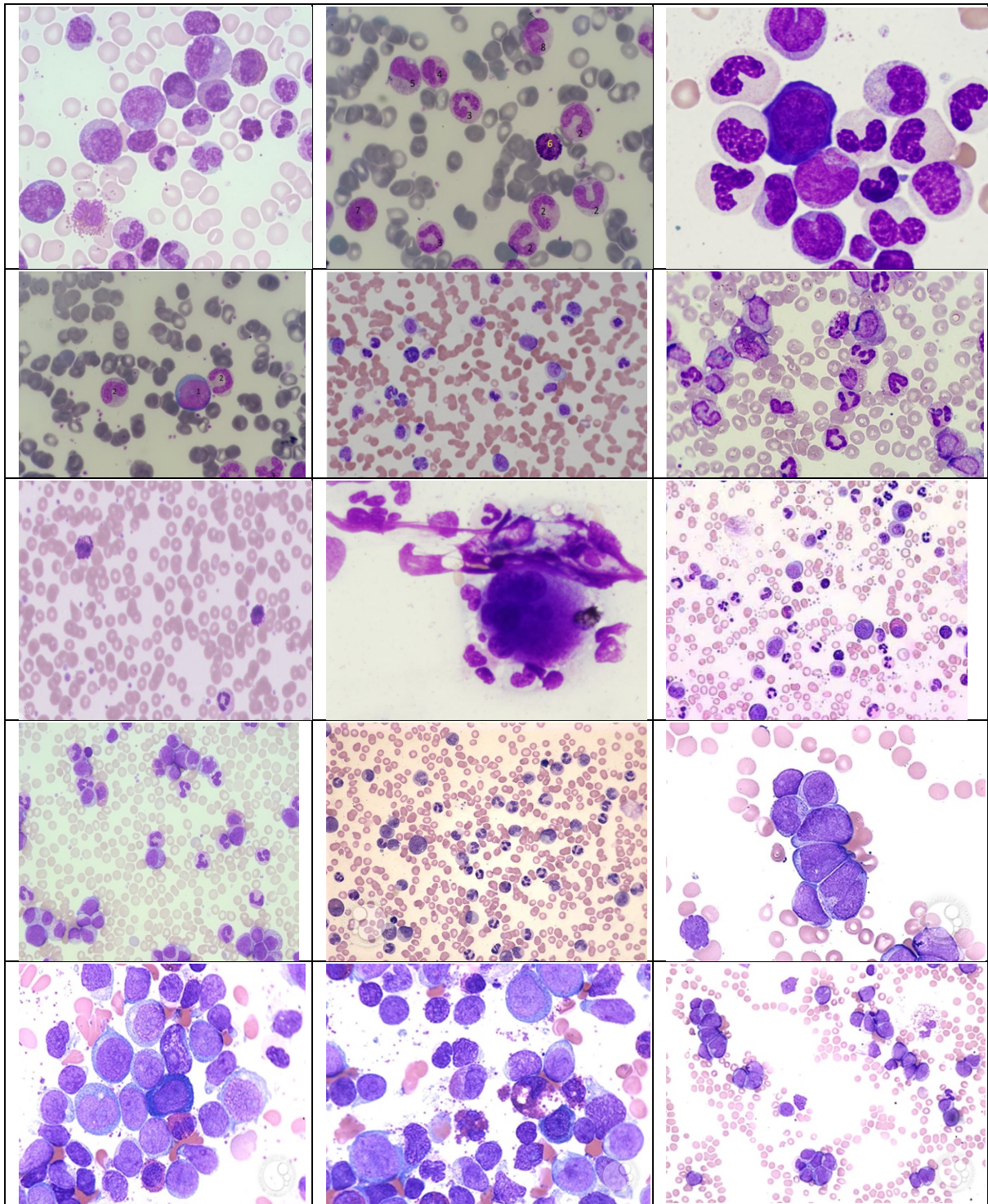
**Image Samples of Acute Myeloid Leukemia**



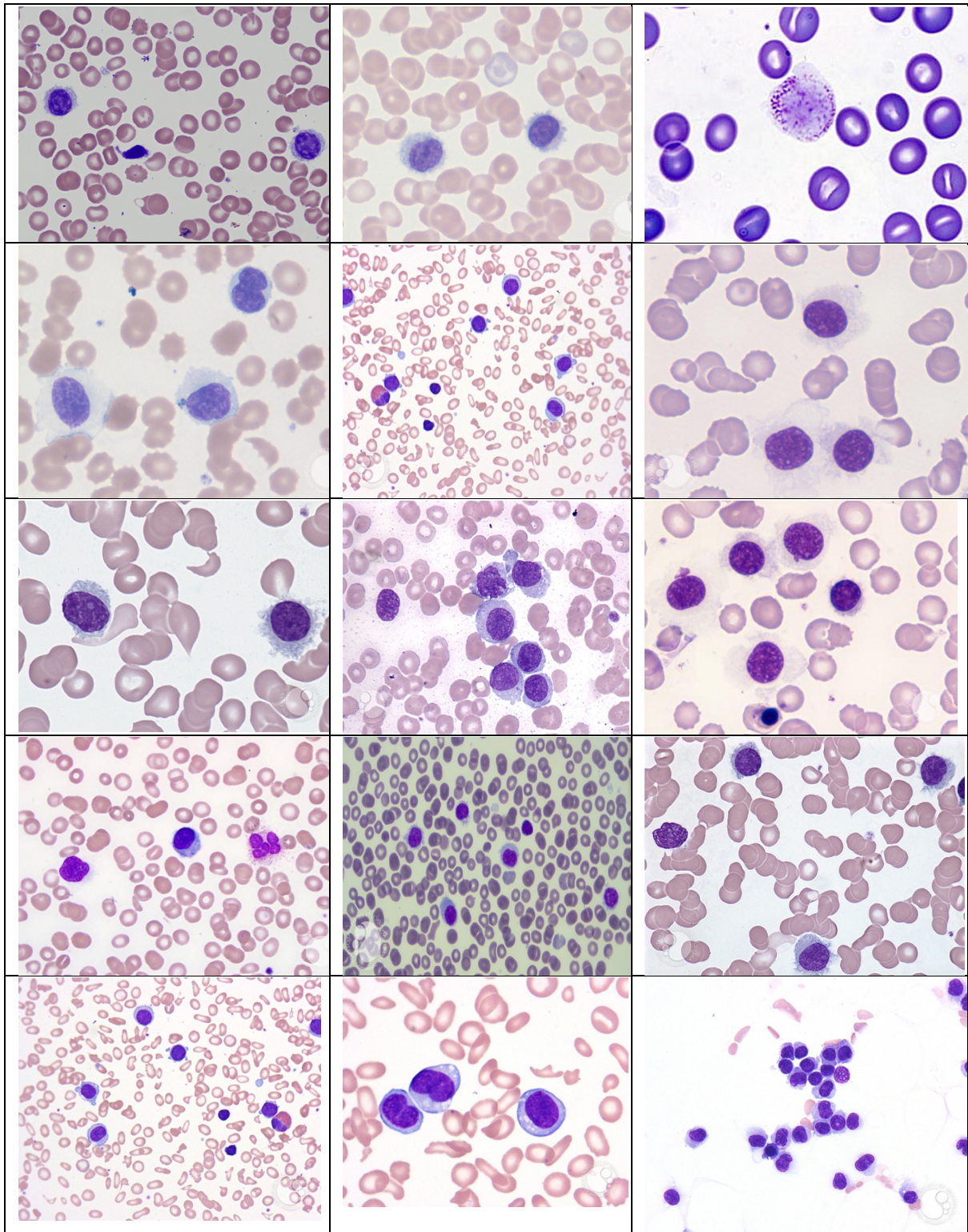
**Image Samples of Chronic Lymphoblastic Leukemia**



**Image Samples of Chronic Myeloid Leukemia**



**Image Samples of Hairy Cell Leukemia**



**Image Samples of Normal Blood Cells**

