**1. Yêu cầu cần thực hiện**

**Mục tiêu của dự án:**

Xây dựng một ứng dụng Python có giao diện người dùng để chuyển đổi văn bản chứa bài toán toán học thành video, bao gồm các bước giải và diễn giải dưới dạng hoạt hình.

**Yêu cầu chính:**

1. **Nhập văn bản chứa bài toán:** Người dùng nhập hoặc tải lên văn bản chứa bài toán toán học.
2. **Phân tích và giải bài toán:** Hệ thống tự động giải bài toán bằng cách sử dụng các công cụ toán học.
3. **Tạo giọng nói từ văn bản:** Chuyển đổi văn bản lời giải thành giọng nói.
4. **Tạo video từ giọng nói và hoạt hình:** Tạo video bằng cách kết hợp hoạt hình và giọng nói.
5. **Hiển thị và lưu trữ video:** Hiển thị kết quả video và cho phép người dùng lưu lại.

**2. Công nghệ và công cụ sử dụng**

**Công cụ và thư viện:**

1. **Python:** Ngôn ngữ lập trình chính.
2. **Tkinter:** Thư viện tạo giao diện người dùng (GUI) trong Python.
3. **SymPy:** Thư viện để giải các bài toán về toán học ký hiệu.
4. **gTTS (Google Text-to-Speech):** Thư viện để chuyển văn bản thành giọng nói.
5. **Manim:** Thư viện tạo hoạt hình toán học.
6. **MoviePy:** Thư viện để xử lý và kết hợp video và âm thanh.
7. **Pandas:** Để xử lý các file CSV chứa bài toán như file math\_dataset\_problem.csv

**3. Chi tiết các bước thực hiện**

**Huy Hoàng**

**Bước 1: Phân tích và xử lý văn bản**

* **Mô tả:** Đầu tiên, cần đọc và phân tích văn bản chứa bài toán. Văn bản này có thể là một chuỗi nhập trực tiếp từ người dùng hoặc được lấy từ một file CSV.
* **Thực hiện:**
  + Đọc dữ liệu từ file CSV (nếu có) và trích xuất bài toán từ văn bản.
  + Phân tích văn bản để nhận diện các thành phần toán học cần giải quyết.

python code

import pandas as pd

# Đọc file CSV

def load\_problems(file\_path):

df = pd.read\_csv(file\_path)

problems = df['Input Text'].tolist() # Giả sử cột chứa bài toán có tên "Input Text"

return problems

# Ví dụ sử dụng:

problems = load\_problems("math\_dataset\_problem.csv")

**Bước 2: Giải bài toán toán học**

* **Mô tả:** Sử dụng SymPy để giải bài toán toán học từ văn bản.
* **Thực hiện:** Tạo hàm nhận vào văn bản, phân tích nó để tìm các phương trình, sau đó sử dụng SymPy để giải bài toán đó.

python code

from sympy import symbols, Eq, solve

def solve\_math\_problem(text):

# Ví dụ đơn giản với phương trình bậc hai

x = symbols('x')

equation = Eq(x\*\*2 + 5\*x + 6, 0)

solutions = solve(equation, x)

return f"The solutions are: {solutions}"

# Ví dụ sử dụng:

result = solve\_math\_problem("Solve the equation x^2 + 5x + 6 = 0")

print(result)

**Bước 3: Chuyển đổi văn bản thành giọng nói**

* **Mô tả:** Dùng gTTS để chuyển đổi lời giải từ dạng văn bản thành giọng nói.
* **Thực hiện:** Tạo file âm thanh từ văn bản.

python code

from gtts import gTTS

def text\_to\_speech(text, output\_file):

tts = gTTS(text)

tts.save(output\_file)

return output\_file

# Ví dụ sử dụng:

audio\_file = text\_to\_speech("The solutions are x equals minus 2 and x equals minus 3", "solution.mp3")

**Minh Hùng**

**Bước 4: Tạo video hoạt hình với Manim**

* **Mô tả:** Dùng Manim để tạo video hoạt hình minh họa quá trình giải bài toán.
* **Thực hiện:** Viết mã để tạo các hoạt hình từ văn bản lời giải.

python code

from manim import \*

class MathAnimation(Scene):

def \_\_init\_\_(self, equation\_text, solution\_text, \*\*kwargs):

self.equation\_text = equation\_text

self.solution\_text = solution\_text

super().\_\_init\_\_(\*\*kwargs)

def construct(self):

equation = MathTex(self.equation\_text)

equation.to\_edge(UP)

self.play(Write(equation))

solution = MathTex(self.solution\_text)

solution.next\_to(equation, DOWN)

self.play(Transform(equation, solution))

self.wait(2)

def create\_animation(math\_solution, output\_file):

animation = MathAnimation("x^2 + 5x + 6 = 0", math\_solution)

animation.render()

return output\_file

**Bước 5: Kết hợp video và âm thanh**

* **Mô tả:** Dùng MoviePy để kết hợp file âm thanh và file video đã tạo thành video hoàn chỉnh.
* **Thực hiện:**

python code

import moviepy.editor as mp

def combine\_audio\_video(video\_file, audio\_file, output\_file):

video = mp.VideoFileClip(video\_file)

audio = mp.AudioFileClip(audio\_file)

final\_video = video.set\_audio(audio)

final\_video.write\_videofile(output\_file, codec='libx264', audio\_codec='aac')

return output\_file

# Ví dụ sử dụng:

final\_video = combine\_audio\_video("quadratic\_equation.mp4", "solution.mp3", "final\_output.mp4")

**Thanh Hòa**

**4. Xây dựng giao diện người dùng với Tkinter**

**Thiết kế giao diện:**

Giao diện cơ bản sẽ bao gồm các thành phần chính:

* **Textbox:** Để nhập văn bản chứa bài toán toán học.
* **Button:** Để người dùng bắt đầu quá trình chuyển đổi.
* **Progress Bar (tuỳ chọn):** Để hiển thị tiến trình xử lý.

**Mã giao diện người dùng:**

python code

import tkinter as tk

from tkinter import filedialog, messagebox

from scripts.solve\_math import solve\_math\_problem

from scripts.text\_to\_speech import text\_to\_speech

from scripts.create\_animation import create\_animation

from scripts.combine\_audio\_video import combine\_audio\_video

import os

# Đường dẫn tới thư mục

audio\_dir = "audio\_files/"

video\_dir = "video\_files/"

final\_dir = "final\_videos/"

# Tạo thư mục nếu chưa tồn tại

for directory in [audio\_dir, video\_dir, final\_dir]:

if not os.path.exists(directory):

os.makedirs(directory)

def process\_text():

# Lấy văn bản từ Textbox

text = input\_textbox.get("1.0", tk.END).strip()

if not text:

messagebox.showerror("Error", "Please enter a math problem text.")

return

try:

# Giải bài toán và tạo nội dung video

math\_solution = solve\_math\_problem(text)

input\_file\_name = "problem"

audio\_file = text\_to\_speech(math\_solution, audio\_dir + input\_file\_name + '.mp3')

video\_file = create\_animation(math\_solution, video\_dir + input\_file\_name + '.mp4')

# Kết hợp video và âm thanh

final\_video = combine\_audio\_video(video\_file, audio\_file, final\_dir + input\_file\_name + '\_final.mp4')

messagebox.showinfo("Success", f"Video created: {final\_video}")

except Exception as e:

messagebox.showerror("Error", f"An error occurred: {str(e)}")

# Tạo cửa sổ chính

root = tk.Tk()

root.title("Math Problem to Video Converter")

# Tạo label và textbox cho input

tk.Label(root, text="Enter Math Problem:").pack(pady=5)

input\_textbox = tk.Text(root, height=10, width=50)

input\_textbox.pack(pady=5)

# Tạo nút để xử lý

process\_button = tk.Button(root, text="Convert to Video", command=process\_text)

process\_button.pack(pady=20)

# Chạy ứng dụng

root.mainloop()

**5. Kết luận và triển khai dự án**

* **Chạy ứng dụng:** Sau khi hoàn thành các bước trên, bạn có thể chạy ứng dụng bằng lệnh python main.py. Ứng dụng sẽ khởi chạy giao diện người dùng và cho phép bạn nhập bài toán và chuyển đổi nó thành video.
* **Triển khai:** Bạn có thể đóng gói ứng dụng thành file thực thi (.exe trên Windows) bằng các công cụ như pyinstaller, hoặc triển khai nó lên một nền tảng web bằng Flask hoặc Django nếu muốn.