**Video Generation with Captions**

**ChatGPT Model Version: GPT-4O**

***My Prompt to ChatGPT:***

* Add captions to the video generated in Part 1 using MoviePy. I have the following images and captions:
* images = [ ("./image1.jpeg", 13), ("./image2.jpeg", 14), ("./image3.jpeg", 13)]
* Corresponding captions:

captions = [ "A sleek black cat sits on a window sill.", "A golden retriever leaps through a field of wildflowers.", "A lone cyclist pedals through a serene forest."]

How do I overlay these captions onto the images and generate the final video with voiceover?

***ChatGPT’s Response:***

Code for adding captions to an existing video using both **MoviePy** and **PIL** (Python Imaging Library). The steps are broken down as follows:

1. **Importing Libraries:**

The code starts by importing necessary libraries:

- `PIL (Pillow)` for creating images with text overlays.

- `MoviePy` for video processing and editing.

- `textwrap` to handle text wrapping for captions.

2. **Loading the Existing Video:**

- `mp.VideoFileClip` is used to load the video file (`video\_path`), which you will modify by overlaying captions.

3. **Captions Data:**

- A list of tuples contains the captions and the respective start and end times. For each caption, the tuple has:

- The caption text.

- The start time (in seconds).

- The end time (in seconds).

4. **Text Image Creation Function:**

- The `create\_text\_image` function generates an image with wrapped text using \*\*PIL\*\*.

- `textwrap.fill` wraps the text into multiple lines based on the `max\_width` parameter (which controls how many characters appear per line).

- The text is then centered and drawn onto the image using `ImageDraw`. A semi-transparent background rectangle is also drawn behind the text to enhance readability on the video.

- The image is returned as an RGBA (with transparency) image object.

5. **Creating Caption Clips:**

- For each caption, the code generates a corresponding image with the caption text using the `create\_text\_image` function. The image is saved to a temporary file.

- The saved image is then converted to a MoviePy `ImageClip`.

- Each `ImageClip` is positioned at the bottom of the video (`.set\_position('bottom')`) and has a set duration and start time (matching the specified start and end times for the caption).

- All these caption clips are added to a list (`caption\_clips`).

6. **Combining Video and Captions:**

- The `mp.CompositeVideoClip` function combines the original video and the list of caption clips to create a new video with captions overlaid.

7. **Saving the Final Video:**

- The video with captions is saved as an MP4 file using the `write\_videofile` method with specific settings (like `fps=24`, `libx264` for video encoding, and `aac` for audio codec).

### **Explanation of Key Parts:**

- **Image Creation:**

The text captions are turned into images using PIL. This allows for greater flexibility in terms of text styling, alignment, and background.

**- Text Wrapping:**

The `textwrap.fill` function ensures that long captions are automatically split into multiple lines.

- **Text Positioning:**

The captions are placed at the bottom of the video by setting the position of the `ImageClip` to `'bottom'`.

- **CompositeVideoClip:**

This is the key function for overlaying different video or image layers onto the base video.

This method is efficient because it uses PIL to handle the more complex task of text rendering (including wrapping and styling) and MoviePy to handle video generation and manipulation.

**Code:**

*from PIL import Image, ImageDraw, ImageFont*

*import moviepy.editor as mp*

*import textwrap*

*# Load the existing video*

*video\_path = "./Application\_of\_Deep\_Learning\_Course.mp4"*

*video = mp.VideoFileClip(video\_path)*

*# List of captions and their respective durations in seconds*

*captions = [*

*("A sleek black cat sits on a window sill, its eyes fixed on the vibrant garden below. Curiosity sparks in its emerald eyes as it watches butterflies flit between colorful blooms.", 0, 13),*

*("A golden retriever leaps through a field of wildflowers, its tail wagging with pure joy. The sun shines brightly, casting playful shadows on its golden fur as it chases a frisbee.", 13, 27),*

*("A lone cyclist pedals through a serene forest, surrounded by towering trees and lush greenery. The gentle rustling of leaves and the chirping of birds create a peaceful ambiance.", 27, 40)*

*]*

*# Function to create a text image using PIL and wrapping the text into multiple lines*

*def create\_text\_image(text, size=video.size, max\_width=40, fontsize=100):*

*font = ImageFont.load\_default() # Using the default PIL font*

*image = Image.new('RGBA', size, (0, 0, 0, 0))*

*draw = ImageDraw.Draw(image)*

*# Wrap text into multiple lines based on max\_width*

*wrapped\_text = textwrap.fill(text, width=max\_width)*

*# Calculate text size to center it*

*text\_size = draw.textsize(wrapped\_text, font=font)*

*text\_x = (size[0] - text\_size[0]) / 2*

*text\_y = size[1] - text\_size[1] - 100 # Padding from the bottom*

*# Draw background rectangle with padding*

*padding = 20*

*background\_size = (text\_size[0] + 2 \* padding, text\_size[1] + 2 \* padding)*

*background\_x = (size[0] - background\_size[0]) / 2*

*background\_y = text\_y - padding*

*draw.rectangle([background\_x, background\_y, background\_x + background\_size[0], background\_y + background\_size[1]], fill=(0, 0, 0, 180))*

*# Draw text*

*draw.text((text\_x, text\_y), wrapped\_text, font=font, fill=(255, 255, 255, 255))*

*return image*

*# Create text clips for captions*

*caption\_clips = []*

*for text, start\_time, end\_time in captions:*

*text\_image = create\_text\_image(text,(600, 200), max\_width=60) # Use max\_width to break lines*

*text\_image.save(f"/tmp/caption\_{start\_time}.png")*

*txt\_clip = mp.ImageClip(f"/tmp/caption\_{start\_time}.png")*

*txt\_clip = txt\_clip.set\_position('bottom').set\_duration(end\_time - start\_time).set\_start(start\_time)*

*caption\_clips.append(txt\_clip)*

*# Overlay text clips on the video*

*video\_with\_captions = mp.CompositeVideoClip([video] + caption\_clips)*

*# Write the final video with captions to a file*

*output\_file\_path\_with\_captions = "./Application\_of\_Deep\_Learning\_Course\_with\_Captions.mp4"*

*video\_with\_captions.write\_videofile(output\_file\_path\_with\_captions, fps=24, codec='libx264', audio\_codec='aac')*

*output\_file\_path\_with\_captions*