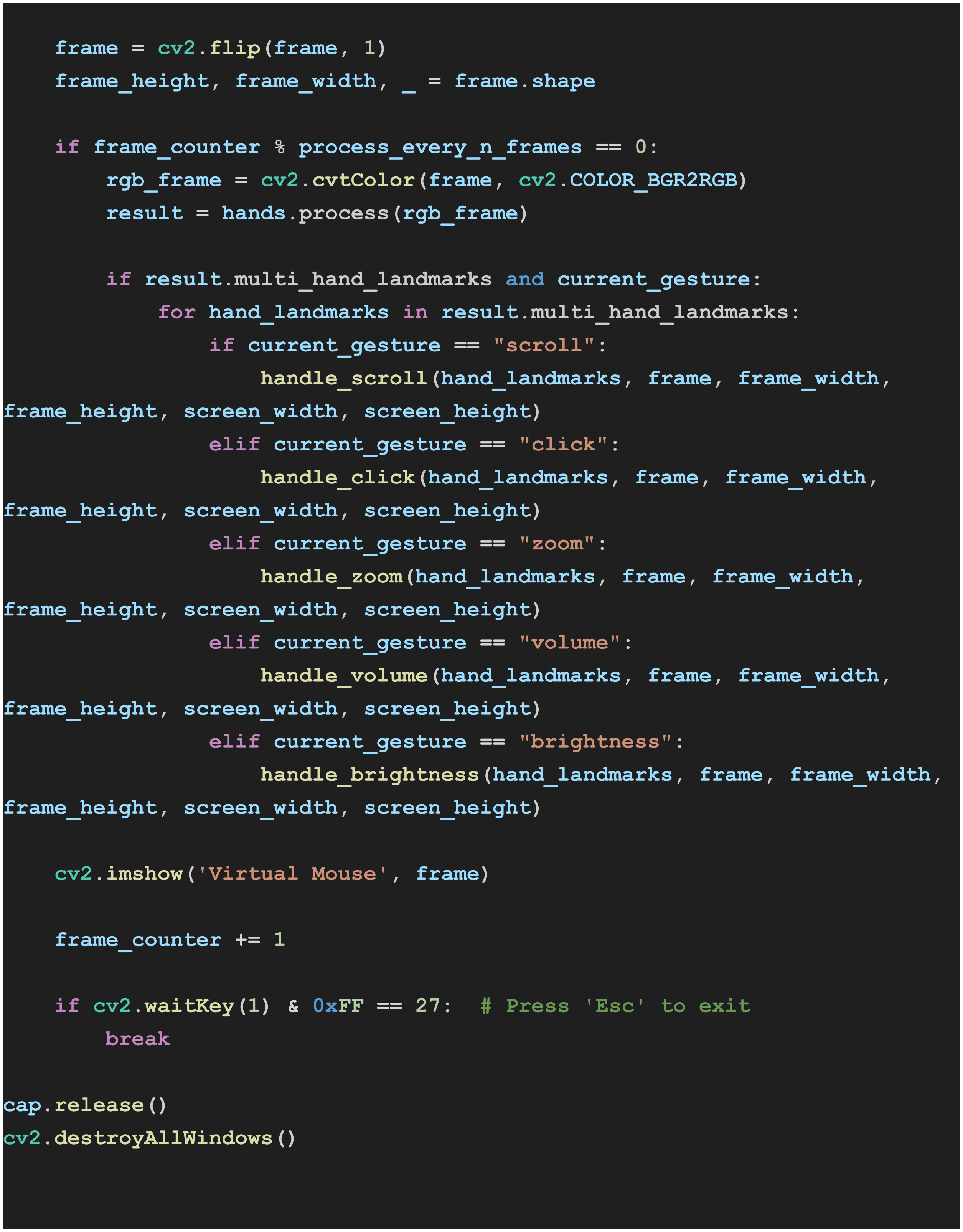
Suggested changes for main.py file

Dated: 23/05/2024

1) Although, this code isn’t tried by me on my system

Optimized code:

```
import cv2
import mediapipe as mp
import pyautogui
import speech_recognition as sr
import threading
import time

from scroll import handle_scroll
from click import handle_click
from zoom import handle_zoom
from volume import handle_volume
from brightness import handle_brightness

# Initialize Mediapipe and webcam
mp_hands = mp.solutions.hands
hands = mp_hands.Hands()
cap = cv2.VideoCapture(0)

screen_width, screen_height = pyautogui.size()

# Variable to store the recognized gesture
current_gesture = None

def detect_gesture():
    global current_gesture
    recognizer = sr.Recognizer()

    while True:
        with sr.Microphone() as source:
            recognizer.adjust_for_ambient_noise(source)
            audio = recognizer.listen(source)
        try:
            command = recognizer.recognize_google(audio).lower()
            print("You said:", command)

            if "scroll" in command:
                current_gesture = "scroll"
            elif "click" in command:
                current_gesture = "click"
            elif "zoom" in command:
                current_gesture = "zoom"
            elif "volume" in command:
                current_gesture = "volume"
            elif "brightness" in command:
                current_gesture = "brightness"
            else:
                print("Unrecognized command.")
                current_gesture = None
        except sr.UnknownValueError:
            print("Could not understand audio.")
        except sr.RequestError as e:
            print("Could not request results from Google Speech
Recognition service; {0}".format(e))

        # Sleep for a bit before listening again to avoid continuous
processing
        time.sleep(1)

# Start the speech recognition thread
gesture_thread = threading.Thread(target=detect_gesture)
gesture_thread.daemon = True
gesture_thread.start()

frame_counter = 0
process_every_n_frames = 5  # Process every 5th frame

while True:
    ret, frame = cap.read()
    if not ret:
        break
```



```
    frame = cv2.flip(frame, 1)
    frame_height, frame_width, _ = frame.shape

    if frame_counter % process_every_n_frames == 0:
        rgb_frame = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
        result = hands.process(rgb_frame)

        if result.multi_hand_landmarks and current_gesture:
            for hand_landmarks in result.multi_hand_landmarks:
                if current_gesture == "scroll":
                    handle_scroll(hand_landmarks, frame, frame_width,
frame_height, screen_width, screen_height)
                elif current_gesture == "click":
                    handle_click(hand_landmarks, frame, frame_width,
frame_height, screen_width, screen_height)
                elif current_gesture == "zoom":
                    handle_zoom(hand_landmarks, frame, frame_width,
frame_height, screen_width, screen_height)
                elif current_gesture == "volume":
                    handle_volume(hand_landmarks, frame, frame_width,
frame_height, screen_width, screen_height)
                elif current_gesture == "brightness":
                    handle_brightness(hand_landmarks, frame, frame_width,
frame_height, screen_width, screen_height)

    cv2.imshow('Virtual Mouse', frame)

    frame_counter += 1

    if cv2.waitKey(1) & 0xFF == 27:  # Press 'Esc' to exit
        break

cap.release()
cv2.destroyAllWindows()
```