

ДОДАТОК А

Апробація результатів роботи



Co-funded by the
European Union

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки
кафедра комп'ютерно-інтегрованих технологій, автоматизації та робототехніки
Харківський національний університет міського господарства
імені О.М. Бекетова
Братиславський університет економіки та менеджменту
Громадська організація «Перспектива»
Угорський університет сільського господарства та природничих наук

МАТЕРІАЛИ

I International Conference

**«Sustainable smart cities and communities:
business and innovation solutions»**

**(Сталі розумні міста та спільноти:
бізнес та інноваційні рішення)**

SSC&C2025

21 квітня 2025

[електронне видання]

Харків 2025

Рисунок А.1 – Титульний аркуш електронного видання

РОЗПІЗНАВАННЯ ЖЕСТИВ І КОМП'ЮТЕРНИЙ ЗІР ДЛЯ БЕЗКОНТАКТНОГО КЕРУВАННЯ ПРИСТРОЯМИ

Д. Є. Заяць

Харківський національний університет радіоелектроніки

Україна, 61166, Харків, пр. Науки 14

E-mail: diana.zaiats@nure.ua

Анотація: У статті розглядається технологія комп'ютерного зору для дистанційного керування пристроями, що базується на розпізнаванні жестів і обробці зображень у реальному часі. Описано застосування цієї технології в промисловості для автоматизованого контролю якості, у медицині для безконтактного управління обладнанням та в побуті для взаємодії зі смарт-пристроями. Наведено приклад реалізації на основі бібліотеки MediaPipe від Google, яка дозволяє швидко та точно розпізнавати жести, перетворюючи їх у команди для комп'ютера. Відзначено гнучкість і перспективи розвитку цієї технології.

Ключові слова: Комп'ютерний зір, розпізнавання жестів, обробка зображень, машинне навчання, Алгоритми штучного інтелекту.

Комп'ютерний зір у системах дистанційного керування – технологія, що використовує камери та алгоритми обробки зображень для розпізнавання жестів, аналізу середовища та взаємодії з пристроями на відстані. Такий підхід усуває необхідність у фізичних контролерах та сенсорних екранах, забезпечуючи зручніше й інтуїтивне керування.

Ці системи базуються на різних технологіях. Для виявлення об'єктів застосовуються алгоритми машинного навчання з можливостями ідентифікації об'єктів і руху в реальному часі. Наприклад, сучасні камери можуть розпізнавати положення рук та інтерпретувати його як команду для пристрою. Серед найважливіших напрямків - системи управління жестами, що дозволяють відстежувати положення пальців і рухи користувача та перетворювати їх на певні команди.

Застосування комп'ютерного зору в дистанційному керуванні широко використовується в різних сферах. У промисловості він використовується для контролю якості продукції без участі оператора, в медицині - для безконтактного управління обладнанням в операційних, а в повсякденному житті - для управління смарт-телевізором жестами без пульта.

Одним із прикладів реалізації таких технологій є проєкт, який використовує бібліотеку MediaPipe від Google. Ця бібліотека дозволяє швидко визначати положення долоні та ключові точки пальців, перетворюючи їх у команди для пристроїв. Для розпізнавання жестів аналізуються координати ключових точок зап'ястя та кінчиків пальців (0, 4, 8, 12, 16, 20), враховується нахил долоні, розділеної на 8 секторів по 45 градусів.

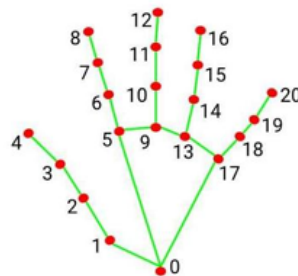


Рис. 1 – Ключові точки долоні за бібліотекою MediaPipe

Завдяки цьому можна отримати до 256 комбінацій жестів, що обробляються зі швидкістю близько 60 кадрів на секунду. Система, написана на Python із використанням OpenCV і MediaPipe, може взаємодіяти з комп'ютером через модуль керування на Golang із бібліотекою Robotgo, керуючи мишею, клавіатурою або виконуючи UNIX-команди. Найбільшою перевагою цієї моделі є її гнучкість, простота додавання функціоналу та пристосування до нових завдань.

ВИСНОВКИ. Комп'ютерний зір у системах дистанційного керування виконує ключову роль у трансформації взаємодії з технологіями, роблячи її інтуїтивнішою та безконтактною. Він дозволяє пристроям «бачити» й аналізувати рухи користувачів, переводячи їх у команди. Це не лише звільняє нас від використання фізичних контролерів, але й робить управління більш точним і швидким.

Практична цінність комп'ютерного зору очевидна в широкому спектрі застосувань. У промисловості він дозволяє автоматизувати контроль якості, у медицині — забезпечити стерильність і точність у роботі з обладнанням, а в побуті — полегшити управління смарт-пристроями. Він підвищує зручність використання технологій для людей з фізичними вадами та знижує поріг використання складних систем.

Подальші перспективи розвитку комп'ютерного зору пов'язані з удосконаленням алгоритмів машинного навчання, що дозволить підвищити точність розпізнавання жестів, враховувати контекст середовища та адаптувати системи до індивідуальних особливостей користувачів. Інтеграція комп'ютерного зору з іншими технологіями штучного інтелекту може створити більш «розумні» й адаптивні рішення, здатні працювати в найрізноманітніших умовах.

ЛІТЕРАТУРА

1. Attar, H., & et al.. (2022). Zoomorphic Mobile Robot Development for Vertical Movement Based on the Geometrical Family Caterpillar. *Computational Intelligence and Neuroscience*, 2022, Article ID 3046116, <https://doi.org/10.1155/2022/3046116>.
2. Stenger, B., Woodley, T., & Cipolla, R. (2010). A Vision-based Remote Control. *Studies in Computational Intelligence*. URL: https://www.researchgate.net/publication/226046175_A_Vision-based_Remote_Control.
3. Favorskaya, M. N., & Jain, L. C. *Computer Vision in Control Systems-3: Aerial and Satellite Image Processing* / M. N. Favorskaya, L. C. Jain ; Reshetnev Siberian State University of Science and Technology, University of Canberra, Bournemouth University, KES International. – Cham : Springer, 2018. – 263 p. – ISBN 978-3-319-67515-2.
4. Yevsieiev, V., Abu-Jassar, A., Maksymova, S., & Demaska, N. (2025). Development of a model for recognizing various objects and tools in a collaborative robot workspace. *ACUMEN: International journal of multidisciplinary research*, 2(1), 224-239
5. Nevliudov, I., Yevsieiev, V., Maksymova, S., Demaska, N., Kolesnyk, K., & Miliutina, O. (2022, September). Object Recognition for a Humanoid Robot Based on a Microcontroller. In *2022 IEEE XVIII International Conference on the Perspective Technologies and Methods in MEMS Design (MEMSTECH)* (pp. 61-64). IEEE.
6. Automated Monitoring and Visualization System in Production / V. Lyashenko, Abu-Jassar Amer Tahseen, V. Yevsieiev, S. Maksymova // *Int. Res. J. Multidiscip. Technovation*, 5(6), 09-18.
7. Gurin, D., Yevsieiev, V., Maksymova, S., & Alkhalaileh, A. (2024). MobileNetv2 Neural Network Model for Human Recognition and Identification in the Working Area of a Collaborative Robot. *Multidisciplinary Journal of Science and Technology*, 4(8), 5-12.
8. Nevliudov, I., Yevsieiev, V., Maksymova, S., Demaska, N., Kolesnyk, K., & Miliutina, O. (2022, September). Object Recognition for a Humanoid Robot Based on a Microcontroller. In *2022 IEEE XVIII International Conference on the Perspective Technologies and Methods in MEMS Design (MEMSTECH)* (pp. 61-64). IEEE.

Міністерство освіти і науки України



NURE

Харківський національний університет
радіоелектроніки

ЗБІРНИК

студентських наукових статей

«Автоматизація та приладобудування»

«Automation and Development of Electronic Devices»

ADED-2025

(Випуск 1)

[електронне видання]



<http://nure.ua/department/kafedra-komp-yuterno-integrovanih-tehnologiy-avtomatizatsiyi-ta-mehatroniki-kitap>



<http://itez.zntu.edu.ua/>



<http://kafea.kdu.edu.ua>

Харків 2025

Рисунок А.4 – Титульний аркуш електронного видання

ШТУЧНИЙ ІНТЕЛЕКТ ТА ІНТЕЛЕКТУАЛЬНІ ПОМІЧНИКИ

Д.Є. Заяць

Харківський національний університет радіоелектроніки

Україна, 61166, Харків, пр. Науки 14

E-mail: diana.zaiats@nure.ua

Анотація: У статті розглядається застосування штучного інтелекту в поєднанні з комп'ютерним зором та розпізнаванням жестів для реалізації інтелектуального помічника. Описано програму, яка дозволяє користувачеві взаємодіяти з віртуальним полотном за допомогою жестів руки, створюючи зображення для подальшого аналізу моделлю ШІ. Представлено функціональну схему системи, що охоплює всі етапи: від захоплення відео та розпізнавання жестів до генерації відповіді штучним інтелектом. Зазначено переваги такого підходу у сфері цифрової взаємодії, зокрема інтуїтивність, гнучкість та можливості застосування у навчанні й креативних задачах. Відзначено перспективність розвитку таких систем завдяки інтеграції передових технологій.

Ключові слова: штучний інтелект, інтелектуальний помічник, розпізнавання жестів, комп'ютерний зір, взаємодія людина-комп'ютер.

ARTIFICIAL INTELLIGENCE AND INTELLIGENT ASSISTANTS

D. Zaiats

Kharkiv National University of Radio Electronics

Ukraine, 61166, Kharkiv, Nauky av.,14

E-mail: diana.zaiats@nure.ua

Annotation: This paper explores the integration of artificial intelligence with computer vision and gesture recognition technologies to implement an intelligent assistant. The study presents a software solution that enables users to interact with a virtual canvas through hand gestures, creating visual content for subsequent analysis by an AI model. A functional architecture of the system is outlined, encompassing all stages – from video capture and gesture interpretation to AI-driven response generation. The approach demonstrates clear advantages in the domain of digital interaction, particularly in terms of intuitiveness, adaptability, and its potential applications in education and creative tasks. The research highlights the promising future of such systems driven by the convergence of cutting-edge technologies.

Key words: artificial intelligence, intelligent assistant, gesture recognition, computer vision, human-computer interaction.

Штучний інтелект (ШІ) сьогодні є однією з ключових технологій, що активно розвивається та знаходить широке застосування в різних галузях діяльності людини. Завдяки зростанню обчислювальних ресурсів, удосконаленню алгоритмів машинного навчання та доступу до великих обсягів даних, системи на основі ШІ здатні виконувати дедалі складніші інтелектуальні завдання, які раніше були властиві лише людині. Однією з найбільш поширених форм використання ШІ стали інтелектуальні помічники (ІП).

Інтелектуальні помічники – це програмні засоби, що дозволяють користувачам взаємодіяти з комп'ютерними системами в зручній, природній мовній формі. Вони можуть здійснювати пошук інформації, планувати події, керувати різними пристроями та виступати посередниками між користувачем і цифровими сервісами. До найбільш відомих представників цієї категорії

належать Siri (Apple), Google Assistant, Alexa (Amazon) та Cortana (Microsoft). Висока ефективність таких рішень пояснюється поєднанням технологій обробки природної мови (Natural Language Processing), машинного навчання (Machine Learning) і хмарних обчислень.

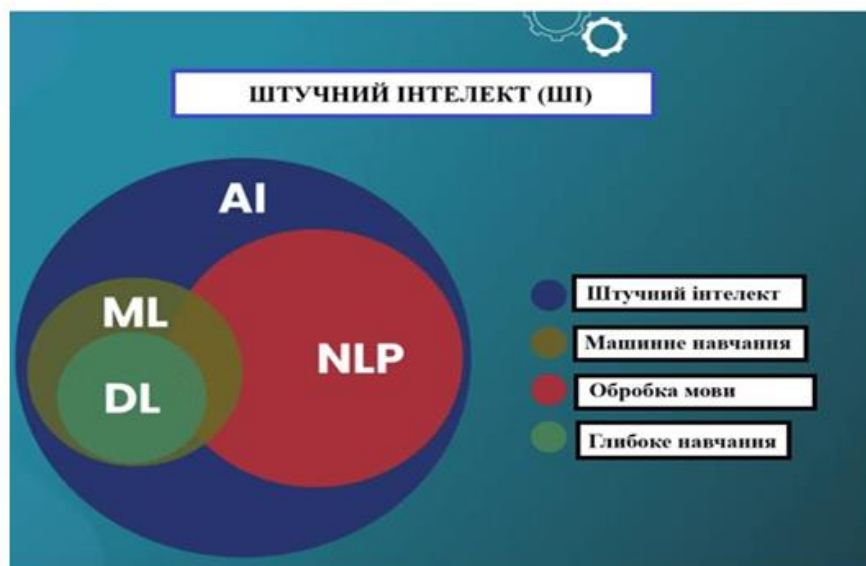


Рис. 1.1 – Компоненти штучного інтелекту

Ключовим елементом у функціонуванні ІІ є здатність обробляти природну мову, що дає змогу правильно інтерпретувати запити користувачів. Сучасні мовні моделі дозволяють цим системам розуміти навіть неформально сформульовані фрази, враховуючи мовні особливості. Під час взаємодії ІІ аналізують контекст, розпізнають наміри користувача й формують відповідні відповіді. Не менш важливою є їхня адаптивність – з кожною новою взаємодією система поступово навчається, покращуючи якість персоналізованих відповідей.

В освітній сфері інтелектуальні помічники відкривають нові можливості. Завдяки голосовій взаємодії та здатності надавати миттєві відповіді, вони можуть стати корисними інструментами для вивчення іноземних мов, самопідготовки або організації навчального процесу. Існують дослідження, які доводять, що використання ІІ сприяє підвищенню мотивації до навчання серед користувачів.

Разом з тим, варто враховувати і ризики, пов'язані з використанням таких систем. Перш за все, йдеться про безпеку персональних даних. Інтелектуальні помічники зазвичай зберігають дані про запити користувача, місцезнаходження, індивідуальні вподобання – і це викликає занепокоєння щодо конфіденційності. Тому надзвичайно важливим є дотримання стандартів кібербезпеки та прозорість у тому, як саме обробляються ці дані.

Ще одним викликом є зростаюча залежність користувачів від подібних технологій. Надмірне використання ІІ може знижувати здатність до самостійного аналізу інформації та прийняття рішень. До того ж, навіть найсучасніші помічники поки що не здатні повністю розпізнавати емоційні відтінки мови чи ефективно функціонувати в умовах невизначеності, що обмежує їхню ефективність у складних комунікативних або етичних ситуаціях.

Отже, інтелектуальні помічники – це перспективний напрямок розвитку технологій штучного інтелекту, який уже сьогодні змінює спосіб взаємодії людини з цифровим світом.

Подальший розвиток цих систем має орієнтуватися не лише на технічну досконалість, а й на етичні принципи та потреби користувачів, забезпечуючи безпечну й ефективну інтеграцію новітніх технологій у повсякденне життя.

В рамках моєї дипломної роботи було розроблено програму, що є прикладом інтелектуального помічника, який поєднує комп'ютерний зір, обробку жестів та інтеграцію зі штучним інтелектом. Завдяки жестам рук користувач може взаємодіяти з віртуальним полотном, створюючи зображення, яке потім аналізується моделлю ШІ. Отримана текстова відповідь виводиться у зручному інтерфейсі. Наведена функціональна схема ілюструє принцип роботи та структуру системи (рис. 1). Кожен з компонентів відповідає за окремий етап обробки даних – починаючи від обробки відеопотоку до формування відповіді за допомогою ШІ.



Рис. 1.2 – Функціональна схема роботи програми

Функціональна схема програми показує, як уся система працює покроково: від моменту розпізнавання жестів руки – до інтерактивного малювання та взаємодії зі штучним інтелектом. Все стартує з простого елемента керування – чекбоксу “Run”. Він відповідає за запуск або зупинку основного циклу роботи. Коли програма активується, починається захоплення відео з вебкамери користувача. Кожен кадр обробляється: модуль для відстеження рук аналізує

положення кисті та визначає, скільки пальців піднято – саме це і дозволяє зрозуміти, який жест було виконано.

Залежно від того, який жест розпізнала система, виконується відповідна дія. Наприклад, якщо піднято один палець – починається процес малювання лінії, якщо п'ять – полотно очищується. Далі малюнок, нанесений на віртуальне полотно, об'єднується з відео з камери. Ці два шари накладаються один на одного з певним рівнем прозорості, і результат у реальному часі виводиться на екран через інтерфейс, створений на платформі Streamlit, що забезпечує зручний зворотний зв'язок для користувача.

Окрім звичайних жестів, система вміє реагувати і на спеціальні – наприклад, чотири піднятих пальці. Це вважається командою для зв'язку з штучним інтелектом. У такому випадку програма автоматично надсилає зображення з полотна до моделі Gemini. Ця модель аналізує малюнок – наприклад, це може бути математичне завдання – і генерує відповідь у текстовому форматі. Отримана відповідь з'являється в окремому блоці вебінтерфейсу. Таким чином, програма забезпечує повноцінну інтерактивну взаємодію: від розпізнавання фізичних жестів до отримання інтелектуальної відповіді, що робить її ефективним інструментом у задачах візуального вводу та обробки інформації.

ВИСНОВКИ. Штучний інтелект та інтелектуальні помічники є важливим напрямом сучасних технологій, що активно трансформують спосіб взаємодії людини з цифровим середовищем. Їхня здатність до аналізу мовних запитів, адаптації та навчання дозволяє створювати ефективні інтерфейси, зокрема у сфері навчання, побуту та персоналізованої підтримки.

Розроблена в рамках роботи програма є прикладом інтеграції кількох таких технологій – комп'ютерного зору, розпізнавання жестів і генеративного ШІ. Завдяки цьому реалізовано інтуїтивну взаємодію: користувач може керувати віртуальним середовищем безконтактно, отримуючи відповідь від моделі ШІ на основі створеного зображення. Це демонструє практичну цінність інтелектуальних помічників у задачах обробки візуальної інформації.

Подальші перспективи розвитку таких систем пов'язані з удосконаленням мовних і візуальних моделей, що дозволить підвищити точність, гнучкість і інтелектуальність цифрових помічників у найрізноманітніших умовах використання.

ЛІТЕРАТУРА

1. How AI is Propelling the Intelligent Virtual Assistants into a New Epoch // AI Authority. URL: <https://aithority.com/botsintelligent-assistants/how-ai-is-propelling-the-intelligent-virtual-assistants-into-a-new-epoch/>.
2. Goksel-Canbek, N., & Mutlu, M. E. (2016). On the track of Artificial Intelligence: Learning with Intelligent Personal Assistants. *International Journal of Human Sciences*, 13(1), 592-601. doi:10.14687/ijhs.v13i1.3549
3. Balci, E. (2019). Overview of Intelligent Personal Assistants. *Acta Infologica*, 3(1), 22–33. <https://doi.org/10.26650/acin.454522>
4. Шаров, О. І. (2019). Становлення штучного інтелекту в Україні та світі. *Інформаційне суспільство*, 1500-Текст статті-4043-1-10-20191111.pdf
5. Що таке ШІ-асистент і чим він відрізняється від чат-бота // Блог Liga.net. URL: <https://blog.liga.net/user/dtsvok/article/39014>
6. AI Assistant Development Company // LeewayHertz. URL: <https://www.leewayhertz.com/ai-assistant/>
7. virtual assistant (AI assistant) // TechTarget. URL: <https://www.techtarget.com/searchcustomerexperience/definition/virtual-assistant-AI-assistant>

8. AI Agents vs. AI Assistants // IBM. URL: <https://www.ibm.com/think/topics/ai-agents-vs-ai-assistants>

9. Attar, H., & et al.. (2022). Zoomorphic Mobile Robot Development for Vertical Movement Based on the Geometrical Family Caterpillar. *Computational Intelligence and Neuroscience*, 2022, Article ID 3046116, <https://doi.org/10.1155/2022/3046116>

10. Chala, O., Yevsieiev, V., Maksymova, S., & Abu-Jassar, A. (2025). USING THE HUMAN FACE RECOGNITION METHOD BASED ON THE MOBILENETV2 NEURAL NETWORK IN AUTHENTICATION SYSTEMS. *Multidisciplinary Journal of Science and Technology*, 5(3), 882-895.

11. Невлюдов, І. Ш., Євсєєв, В. В., & Гурін, Д. В. (2025). MODEL DEVELOPMENT OF DYNAMIC REPRESENTATION A MODEL DESCRIPTION PARAMETERS FOR THE ENVIRONMENT OF A COLLABORATIVE ROBOT MANIPULATOR WITHIN THE INDUSTRY 5.0 FRAMEWORK. *Системи управління, навігації та зв'язку. Збірник наукових праць*, 1(79), 42-48.

12. Abu-Jassar, A. T., Attar, H., Amer, A., Lyashenko, V., Yevsieiev, V., & Solyman, A. (2025). Development and Investigation of Vision System for a Small-Sized Mobile Humanoid Robot in a Smart Environment. *International Journal of Crowd Science*, 9(1), 29-43.

13. Yevsieiev, V., Maksymova, S., Alkhalaileh, A., & Gurin, D. (2025). Development of a program for processing 3d models of objects in a collaborative robot workspace using an HD camera. *ACUMEN: International journal of multidisciplinary research*, 2(1), 194-210.

14. Yevsieiev, V., Abu-Jassar, A., Maksymova, S., & Demska, N. (2025). Development of a model for recognizing various objects and tools in a collaborative robot workspace. *ACUMEN: International journal of multidisciplinary research*, 2(1), 224-239.

15. Yevsieiev, V., Abu-Jassar, A., & Maksymova, S. (2024). Calculation of the Distance to Objects in Collaborative Robots Workspace Using Computer Vision. *Journal of universal science research*, 2(11), 240-255.

16. Maksymova, S., Abu-Jassar, A., Gurin, D., & Yevsieiev, V. (2024). Comparative Analysis of methods for Predicting the Trajectory of Object Movement in a Collaborative Robot-Manipulator Working Area. *Multidisciplinary Journal of Science and Technology*, 4(10), 38-48.

17. Abu-Jassar, A. T., Attar, H., Amer, A., Lyashenko, V., Yevsieiev, V., & Solyman, A. (2024). Remote Monitoring System of Patient Status in Social IoT Environments Using Amazon Web Services (AWS) Technologies and Smart Health Care. *International Journal of Crowd Science*, 8.

18. Vladyslav, Y., & Bronnikov, A. (2020, October). ANALYSIS OF THE CMMI MODEL APPLICATION FOR SOLVING THE TASKS OF CPPS CONTROL PROCESSES AUTOMATION DEVELOPMENT. In The 4 th International scientific and practical conference "Actual trends of modern scientific research"(October 11-13, 2020) MDPC Publishing, Munich, Germany. 2020. 386 p. (p. 128).

19. Yevsieiev, V. V., & Bronnikov, A. I. (2020). Development of databases interconnection "essences" information model for cyber-physical production systems additive cyber design creation automation. *Збірник Наукових Праць НУК*, №3. С.56-62. DOI [https://doi.org/10.15589/znp2020.3\(481\).7](https://doi.org/10.15589/znp2020.3(481).7)

20. Yevsieiev V., Bronnikov A. Information systems development methodologies application analysis for cyber-physical production systems development. III International scientific-practical conference "Theory, science and practice" (Japan, Tokyo, 5–8 October 2020). P. 398–401. DOI: 10.46299/ISG.2020.II.III

Науковий керівник: Бронніков Артем, доцент кафедри КІТАР Харківського національного університету радіоелектроніки.

ДОДАТОК Б

Код програми

```
import cvzone
import cv2
from cvzone.HandTrackingModule import HandDetector
import numpy as np
from google import genai
from PIL import Image, ImageDraw, ImageFont
import streamlit as st
import time
import csv
from datetime import datetime
import threading
import winsound
import requests
import io

BOT_TOKEN = "7840194256:AAGCh2qDRpdChvTzWcz2PdYnDwpGMII4gz8"
CHAT_ID = "-1002212573067"

DETECTION_CONFIDENCE = 0.8
MIN_TRACKING_CONFIDENCE = 0.7

def log_event(event):
    with open("logs.csv", "a", newline="", encoding="utf-8-sig") as file:
        writer = csv.writer(file)
        now = datetime.now().strftime("%Y-%m-%d %H:%M:%S")
        writer.writerow([now, f"Подія: {event}"])
```

```

def send_telegram_alert(img, text, gesture_type):
    url = f"https://api.telegram.org/bot{BOT_TOKEN}/sendPhoto"

    img_pil = Image.fromarray(cv2.cvtColor(img, cv2.COLOR_BGR2RGB))
    draw = ImageDraw.Draw(img_pil)

    font_path = "RobotoMono.ttf" # Наприклад: "arial.ttf" або "times.ttf"
    font_size = 30
    try:
        font = ImageFont.truetype(font_path, font_size)
    except:
        font = ImageFont.load_default()
        print("Увага: використовується стандартний шрифт замість українського")

    if gesture_type == "alert":
        draw.text((50, 50), " ТРИБОГА", font=font, fill=(255, 0, 0))
    elif gesture_type == "help":
        draw.text((50, 50), " ПОТРІБНА ДОПОМОГА", font=font, fill=(0, 165, 255))
    alert_img = cv2.cvtColor(np.array(img_pil), cv2.COLOR_RGB2BGR)

    _, img_encoded = cv2.imencode('.jpg', alert_img)
    img_bytes = img_encoded.tobytes()

    files = {'photo': ('alert.jpg', img_bytes)}
    data = {'chat_id': CHAT_ID, 'caption': text}

    requests.post(url, files=files, data=data)

coll, col2 = st.columns([2, 1])
with coll:

```

```

run = st.checkbox("Run", value=True)
FRAME_WINDOW = st.image([])

with col2:
    output_text_area = st.title("Answer")
    output_text_area = st.subheader("")

client = genai.Client(api_key="AIzaSyBbgm76F4zAxsftl2LpOpHfU0Tu6Wt6yM8")

cap = cv2.VideoCapture(0)
detector = HandDetector(
    staticMode=False,
    maxHands=1,
    modelComplexity=1,
    detectionCon=DETECTION_CONFIDENCE,
    minTrackCon=MIN_TRACKING_CONFIDENCE
)

GESTURE_HISTORY_LENGTH = 5
gesture_history = []
current_gesture = None
gesture_stable_time = 0

def get_stable_gesture(new_gesture):
    global gesture_history, current_gesture, gesture_stable_time

    gesture_history.append(new_gesture)
    if len(gesture_history) > GESTURE_HISTORY_LENGTH:
        gesture_history.pop(0)

```

```

    if len(gesture_history) == GESTURE_HISTORY_LENGTH and all(g == new_gesture for
g in gesture_history):
        if current_gesture == new_gesture:
            gesture_stable_time += 1
        else:
            current_gesture = new_gesture
            gesture_stable_time = 1

        if gesture_stable_time >= 3:
            return current_gesture
    return None

```

```

def getHandInfo(img):
    hands, img = detector.findHands(img, draw=True, flipType=True)
    if hands:
        hand = hands[0]
        lmList = hand["lmList"]
        fingers = detector.fingersUp(hand)
        return fingers, lmList, hand
    else:
        return None

```

```

def draw(info, prev_pos, canvas):
    fingers, lmList, hand = info
    current_pos = None

    if fingers == [0, 1, 0, 0, 0]:
        index_tip = lmList[8]
        thumb_tip = lmList[4]

```

```
distance = ((index_tip[0] - thumb_tip[0]) ** 2 + (index_tip[1] - thumb_tip[1]) ** 2) **
```

0.5

```
if distance > 50:
    current_pos = index_tip[0:2]
    if prev_pos is not None:
        cv2.line(canvas, current_pos, prev_pos, color=(255, 0, 255), thickness=10)
    else:
        prev_pos = current_pos
```

```
elif fingers == [1, 1, 1, 1, 1]:
    canvas = np.zeros_like(img)
```

```
return current_pos, canvas
```

```
def sendToAI(canvas, fingers):
    if fingers == [1, 1, 1, 1, 0]:
        if np.any(canvas > 0):
            pil_image = Image.fromarray(cv2.cvtColor(canvas, cv2.COLOR_BGR2RGB))
            response = client.models.generate_content(
                model="gemini-2.0-flash",
                contents=["Розв'яжи математичне рівняння", pil_image]
            )
            return response.text
        return None
```

```
prev_pos = None
```

```
canvas = None
```

```
image_combined = None
```

```
output_text = ""
```

```
last_gesture_time = time.time()

while run:
    success, img = cap.read()
    if not success:
        continue

    img = cv2.flip(img, 1)

    if canvas is None:
        canvas = np.zeros_like(img)

    info = getHandInfo(img)
    if info:
        fingers, lmList, hand = info

        stable_gesture = get_stable_gesture(tuple(fingers))

        if stable_gesture:
            if stable_gesture == (0, 0, 0, 0, 0):
                if time.time() - last_gesture_time > 5:
                    blink_canvas = np.zeros_like(img)
                    blink_canvas[:] = (0, 0, 255)
                    for _ in range(3):
                        FRAME_WINDOW.image(blink_canvas, channels="BGR")
                        time.sleep(0.2)
                    FRAME_WINDOW.image(img, channels="BGR")
                    time.sleep(0.2)

            threading.Thread(target=lambda: winsound.Beep(1000, 500)).start()
            log_event("ТРИВОГА")
```

```

threading.Thread(
    target=send_telegram_alert,
    args=(img, "ТРИВОГА! ", "alert")
).start()

output_text = "ТРИВОГА!"
last_gesture_time = time.time()

elif stable_gesture == (0, 1, 1, 1, 0):
    if time.time() - last_gesture_time > 5:
        help_canvas = np.zeros_like(img)
        help_canvas[:] = (0, 165, 255)
        FRAME_WINDOW.image(help_canvas, channels="BGR")
        time.sleep(0.5)
        FRAME_WINDOW.image(img, channels="BGR")
        time.sleep(0.5)
        FRAME_WINDOW.image(help_canvas, channels="BGR")
        time.sleep(0.5)
        FRAME_WINDOW.image(img, channels="BGR")

    log_event("Потрібна допомога")

    threading.Thread(
        target=send_telegram_alert,
        args=(img, "ПОТРІБНА ДОПОМОГА! ", "help")
    ).start()

    output_text = " Запитано допомогу"
    last_gesture_time = time.time()

elif stable_gesture == (1, 0, 0, 0, 0):
    if time.time() - last_gesture_time > 5:

```

```

green_canvas = np.zeros_like(img)
green_canvas[:] = (0, 255, 0)
FRAME_WINDOW.image(green_canvas, channels="BGR")
time.sleep(1)
log_event("Стан: НОРМА")
threading.Thread(
    target=send_telegram_alert,
    args=(img, " Стан: НОРМА ", "normal")
).start()
output_text = " Стан: НОРМА"
last_gesture_time = time.time()

elif stable_gesture == (1, 0, 0, 0, 1):
    if time.time() - last_gesture_time > 5:
        log_event("Огляд завершено")
        threading.Thread(
            target=send_telegram_alert,
            args=(img, " Огляд завершено ", "normal")
        ).start()
        output_text = " Огляд завершено"
        last_gesture_time = time.time()

prev_pos, canvas = draw(info, prev_pos, canvas)
result = sendToAI(canvas, fingers)
if result:
    output_text = result

image_combined = cv2.addWeighted(img, 0.7, canvas, 0.3, 0)
FRAME_WINDOW.image(image_combined, channels="BGR")

if output_text:
    output_text_area.text(output_text)

```

```
cv2.waitKey(1)  
cap.release()
```

ДОДАТОК В
Демонстраційний матеріал

