

ДОДАТОК А

Фрагмент коду розпізнавання та ідентифікацій розробленої системи ідентифікації знаходження людини в робочій зоні колаборативного робота

```
import cv2
import numpy as np
import tensorflow as tf

# Завантаження моделі MobileNetV2 для розпізнавання об'єктів
model =
tf.saved_model.load(r"C:\Users\Vitalii\.cache\kagglehub\models\tensorflow\ssd-
mobilenet-v2\tensorFlow2\fpnlite-320x320\1")

# Функція для розпізнавання об'єктів на зображенні
def detect_objects(image):
    input_tensor = tf.convert_to_tensor(image, dtype=tf.uint8)
    input_tensor = input_tensor[tf.newaxis, ...]
    detections = model.signatures['serving_default'](input_tensor)
    return detections

# Ініціалізація камери
cap = cv2.VideoCapture(0)
while True:
    ret, frame = cap.read()
    if not ret:
        break

    # Змінити розмір кадру до потрібного для моделі
    input_frame = cv2.resize(frame, (320, 320))
    # Виконати розпізнавання об'єктів
```

```

detections = detect_objects(input_frame)

# Обробка результатів розпізнавання
num_detections = int(detections['num_detections'][0])
detection_classes = detections['detection_classes'][0].numpy().astype(np.int64)
detection_boxes = detections['detection_boxes'][0].numpy()
detection_scores = detections['detection_scores'][0].numpy()

# Виведення результатів на кадр
for i in range(num_detections):
    if detection_scores[i] > 0.5 and detection_classes[i] == 1: # Клас 1
        відповідає людині
        box = detection_boxes[i]
        y1, x1, y2, x2 = box
        y1, x1, y2, x2 = int(y1 * frame.shape[0]), int(x1 * frame.shape[1]), int(y2 *
frame.shape[0]), int(x2 * frame.shape[1])
        cv2.rectangle(frame, (x1, y1), (x2, y2), (0, 255, 0), 2)
        cv2.putText(frame, 'Person', (x1, y1 - 10),
cv2.FONT_HERSHEY_SIMPLEX, 0.9, (0, 255, 0), 2)
        cv2.imshow('Video', frame)
        if cv2.waitKey(1) & 0xFF == ord('q'):
            break
cap.release()
cv2.destroyAllWindows()

```

ДОДАТОК Б

Фрагмент коду розпізнавання, ідентифікацій та трекінгу у розробленій системі ідентифікації знаходження людини в робочій зоні колаборативного робота

```
import cv2
import numpy as np
import tensorflow as tf

# Завантаження моделі MobileNetV2 для розпізнавання об'єктів
model = tf.saved_model.load(
    r"C:\Users\Vitalii\.cache\kagglehub\models\tensorflow\ssd-mobilenet-
v2\tensorFlow2\fpnlite-320x320\1")

# Функція для розпізнавання об'єктів на зображенні
def detect_objects(image):
    input_tensor = tf.convert_to_tensor(image, dtype=tf.uint8)
    input_tensor = input_tensor[tf.newaxis, ...]
    detections = model.signatures['serving_default'](input_tensor)
    return detections

# Ініціалізація камери
cap = cv2.VideoCapture(0)
# Змінна для зберігання трекінг-області
tracking_window = None
while True:
    ret, frame = cap.read()
    if not ret:
        break
    # Змінити розмір кадру до потрібного для моделі
```

```

input_frame = cv2.resize(frame, (320, 320))

# Виконати розпізнавання об'єктів
detections = detect_objects(input_frame)

# Обробка результатів розпізнавання
num_detections = int(detections['num_detections'][0])
detection_classes = detections['detection_classes'][0].numpy().astype(np.int64)
detection_boxes = detections['detection_boxes'][0].numpy()
detection_scores = detections['detection_scores'][0].numpy()

# Якщо трекінг-область не ініціалізована, знайти перший об'єкт "Person"
для трекінгу
if tracking_window is None:
    for i in range(num_detections):
        if detection_scores[i] > 0.5 and detection_classes[i] == 1: # Клас 1
            відповідає людині
                box = detection_boxes[i]
                y1, x1, y2, x2 = box
                y1, x1, y2, x2 = int(y1 * frame.shape[0]), int(x1 * frame.shape[1]),
int(y2 * frame.shape[0]), int(
                    x2 * frame.shape[1])
                tracking_window = (x1, y1, x2 - x1, y2 - y1)
                break

# Виконати трекінг з CAMShift, якщо трекінг-область ініціалізована
if tracking_window is not None:
    hsv = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)
    roi = frame[tracking_window[1]:tracking_window[1] + tracking_window[3],
tracking_window[0]:tracking_window[0] + tracking_window[2]]

```

```

# Переконайтеся, що roi має розмірність 3 канали
if roi.size > 0:
    roi_hsv = cv2.cvtColor(roi, cv2.COLOR_BGR2HSV)
    roi_hist = cv2.calcHist([roi_hsv], [0], None, [180], [0, 180])
    cv2.normalize(roi_hist, roi_hist, 0, 255, cv2.NORM_MINMAX)

# Трекінг за допомогою CAMShift
dst = cv2.calcBackProject([hsv], [0], roi_hist, [0, 180], 1)
ret, tracking_window = cv2.CamShift(dst, tracking_window, (255, 0, 0))

# Виведення результатів на кадр
pts = cv2.boxPoints(ret)
pts = np.intp(pts) # Зміна з np.int0 на np.intp
cv2.polylines(frame, [pts], True, (255, 0, 0), 2)
cv2.putText(frame, 'Tracking', (tracking_window[0], tracking_window[1] -
10), cv2.FONT_HERSHEY_SIMPLEX, 0.9,
            (255, 0, 0), 2)

# Відображення кадру
cv2.imshow('Video', frame)
if cv2.waitKey(1) & 0xFF == ord('q'):
    break

cap.release()
cv2.destroyAllWindows()

```

ДОДАТОК В

Апробація результатів наукових досліджень

УДК 681.5:004.89

РОЗРОБКА ЕЛЕМЕНТІВ СИСТЕМИ ІДЕНТИФІКАЦІЇ ЗНАХОДЖЕННЯ ЛЮДИНИ В РОБОЧІЙ ЗОНІ КОЛАБОРАТИВНОГО РОБОТА

Тетеря В.В.

Науковий керівник – к.н.т., доц. Максимова С.С.

Харківський національний університет радіоелектроніки, каф. КІТАР

м. Харків, Україна

e-mail: vitalii.teteria@nure.ua

This work is devoted to the use of modern computer vision technologies for the development of human presence identification systems in the working area of a collaborative robot. Visual data processing methods are presented, as well as their integration to improve the accuracy and reliability of the identification system. There is a very popular library for object identification such as OpenCV in the C++ programming language, with the help of this library it is possible to recognize faces, movements, specific details of the human body to protect a person from significant injuries that can be received at the workplace when interacting with a robot.

На даний момент майже всі великі підприємства мають виробничу ділянку, на якій розташовані роботи, які виконують завдання різного роду. Значна кількість роботів не мають досі системи захисту від травмувань людини, яка взаємодіє з тим чи іншим роботом. Тому виникає попит створення програмного забезпечення для колаборативних роботів для уникнення отримання травм людини при взаємодії з роботом. Для написання такого програмного забезпечення виникає необхідність використання бібліотек у сфері комп'ютерного зору та машинного навчання.

Машинне навчання - це галузь штучного інтелекту, яка вивчає методи, які дозволяють комп'ютерам вчитися з даних та робити прогнози або приймати рішення без явного програмування. Машинне навчання може бути застосоване до багатьох завдань у комп'ютерному зорі, таких як класифікація об'єктів на зображеннях, виявлення облич, визначення об'єктів тощо. Комп'ютерний зір та машинне навчання часто використовуються разом для розв'язання складних задач у сфері автоматизації, робототехніки, медицині, безпеці та інших.

Якщо проаналізувати існуючі виробництва з ділянками, на яких працюють роботи, маємо попит в модифікації даних роботів до колаборативних для уникнення отримання травм людиною на виробництві. Таким чином набагато зменшиться кількість травм на виробництві.

Одна з найбільш популярних бібліотек комп'ютерного зору та машинного навчання є бібліотека OpenCV, яка була розроблена компанією

Intel. Надає велику кількість засобів для обробки, аналізу, розпізнавання об'єктів на зображенні, відстеження руху об'єктів, перетворення зображень та інші.

OpenCV – це спеціальна бібліотека відкритого програмного забезпечення, яка надає інструменти для розробки програмного забезпечення у сфері комп'ютерного зору та машинного навчання. Вона дозволяє виконувати різноманітні завдання, які пов'язані з обробкою зображень та відео, а також ідентифікацією об'єктів.

Вирішення проблеми створення програмного забезпечення для колаборативних роботів можливе за допомогою середовища розробки Microsoft Visual Studio та мови програмування C++. Бібліотека OpenCV має гарну сумісність з даною мовою, що надає зручності для написання програмного засобу.

Для створення безпосередньо самого програмного засобу необхідно створити MFC проєкт, налаштувати його та підключити всі необхідні бібліотеки для функціонування методів бібліотеки OpenCV.

Наступним кроком є обрання методу ідентифікації людини в зоні дії колаборативного робота, який найбільше підходить під той чи інший робот, в залежності від виду його роботи. Наприклад: ідентифікація по обличчю, частини тіла, руки або ноги.

Останнім кроком буде тестування створеного програмного забезпечення, відладка її роботи та після цього встановлення прошивки на роботів.

Можемо зробити висновок, що в теперішній час використання бібліотеки OpenCV в розробці програмного забезпечення для колаборативних роботів являє собою потужне рішення. При розробці ПЗ можуть виникати деякі складнощі, але всі вони мають вирішення.

Список використаних джерел:

1. Бібліотека OpenCV. URL: <https://opencv.org/> (дата звернення: 04.02.2024).
2. Використання бібліотеки opencv для розпізнавання людини в зоні дії робота. URL: <https://evergreens.com.ua/ru/articles/open-cv-face-recognition.html> (дата звернення: 07.02.2024).
3. Колаборативні роботи в галузі виробництва. URL: <https://www.fanuc.eu/ua/uk/> (дата звернення: 10.02.2024).

ДОДАТОК Г
Демонстраційний матеріал

Формат	Позиція	ПОЗНАЧЕННЯ	НАЙМЕНУВАННЯ	Кількість	Примітки		
			<u>Текстові документи</u>				
A4	1	ГЮІК 501210.017 ПЗ	Пояснювальна записка	88 с.			
			<u>Інші документи</u>				
	2		Додаток А Фрагмент коду розпізнавання та ідентифікацій розробленої системи	2 с.			
	3		Додаток Б Фрагмент коду розпізнавання, ідентифікацій та тренгу у розробленої системи	3 с.			
	4		Додаток В Апробація результатів наукових досліджень				
	5		Додаток Г Демонстраційний матеріал				
			ГЮІК 501210.017 ПЗ				
Змін	Арк.	№ докум.	Підпис	Дата			
Розробив		Тетеря В.В.			Розроблення системи автоматизації для ідентифікації оператора в робочій зоні колаборативного робота		
Перевірив		Максимова С.С.					
Норм. контроль		Демська Н.П.					
Затв.		Невлюдов І.Ш.					
					Літера	Аркуш	Аркуш
					Н	1	1
					Кафедра КІТАР ХНУРЕ		