

Швачка О. О., Барбарук В. М.

ДОСЛІДЖЕННЯ КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ ТА МЕТОДІВ РОЗПІЗНАВАННЯ ЖЕСТИВ ЛЮДИНИ ДЛЯ КЕРУВАННЯ КОМП'ЮТЕРОМ

У наведеній статті розглядається один з перспективних напрямків розвитку інформаційних технологій в області інтерфейсу людина-машина на основі використання природного для людини способу спілкування з комп'ютерами. Надано огляд основних етапів, процедур та технологій збору даних, що застосовуються у роботі систем розпізнавання жестів. Було проведено аналіз існуючих рішень в цій області. Було проаналізовано сенсори для отримання карти глибини. Використовуючи карту глибини було розроблено та реалізовано метод розпізнавання жестів руки оснований на дескрипторах та з використанням машинного навчання.

Ключові слова: розпізнавання жестів, сегментація, людина – комп'ютерний інтерфейс, датчик глибини, датчик руху, спосіб взаємодії.

Актуальність дослідження. Перспективним напрямком розвитку інформаційних технологій є розробка нових способів забезпечення інтерфейсу людина-машина. Перед розробниками подібних інтерфейсів ставиться задача використання природних для людини способів спілкування з комп'ютерами. Враховуючи усі можливі перешкоди та наявність шумів в оточуючому середовищі, перевага надається системам на основі комп'ютерного зору. Особливо перспективними для побудови інтерфейсів управління програмним та апаратним забезпеченням комп'ютерів є жести.

Постановка проблеми. Людино-комп'ютерна взаємодія - міждисциплінарний науковий напрям, що вивчає і вдосконалює методи розробки інтерактивних комп'ютерних систем, призначених для зручного використання людиною.

HCI (англ. human-computer interaction) отримало розвиток в контексті різних дисциплін: комп'ютерної графіки, когнітивної науки, інформатики і багато ін.

У зв'язку з тим що, HCI вивчається як з людського боку, так і з боку комп'ютера, знання які отримані в дослідженні, опираються як на людський фактор, так і на комп'ютерний. З комп'ютерного боку важливі технології: комп'ютерна графіка, мови програмування, операційні системи. З людського боку: теорія комунікації, когнітивна психологія, соціологія. Так само має значення інженерія та проектування.

Основне завдання HCI поліпшити взаємодію між людиною і комп'ютером. Кінцева мета HCI, створити якісний інтерфейс, який можна назвати точкою зв'язку між людиною і комп'ютером.

В останні роки широке поширення отримав один з методів HCI, заснований на розпізнаванні жестів людини. Жест - рух (дія) для вираження емоції чи інформації замість розмови, або під час розмови. Дія або рух людського тіла, або його частини, що має певне значення, або сенс, тобто є знаком або символом.

Рух пальців, рук, голови, плечей, міміка обличчя, все це - жести. У HCI, жестами передають інформацію в комп'ютер. Розпізнавання жестів можна застосовувати в управлінні комп'ютером і побутовими приладами, створення природних людино-машинних інтерфейсів для глухонімих, маніпуляція тривимірними об'єктами, додатки віртуальної реальності.

Існуючі методи для керування комп'ютером мають різні недоліки. У статті запропоновано новий підхід до реалізації завдання керування комп'ютером за допомогою жестів. Новий запропонований метод буде перевірено на певній кількості жестів, та отримані статистичні дані його застосування.

Мета статті. Аналіз роботи систем розпізнавання жестів, які розглядаються як системи машинного зору для виявлення та позиціонування жестів людини, та запропоновання нового методу розпізнавання жесту руки, який буде перевірено на певній кількості жестів, та отримано статистичні дані про його працездатність.

Основний текст. Першим етапом будь-якої системи розпізнавання є збір даних.

Інформація, отримана за допомогою зображення, обробляється для виділення області руки від фону. Цей етап називається локалізацією руки на зображенні.

Після збору даних стає можливим використання первинної інформації про руку для фільтрації даних та видалення шумів на зображенні, які можуть виникнути, наприклад, через перепади освітлення, а також видалення артефактів (наявність на руці каблучок). Дана процедура є надзвичайно важливою, так як набором жестів, які необхідно розпізнати, визначається кількість та розмір отриманих даних і для реалізації якісного розпізнавання необхідною умовою є точність (достовірність) цих даних.

На етапі розпізнавання рухів рук здійснюють виділення ознак. Вибір ознаки є важливою частиною процесу розпізнавання, тому що рухи рук відрізняються розмаїтістю форм, рухів і текстур. Для розпізнавання статичного положення руки застосовують геометричні ознаки, такі як кінчики пальців, напрямок пальців.

Наступним етапом буде визначення конкретних жестів на основі аналізу відфільтрованих даних, які несуть інформацію про рух руки. З цією метою проводиться процедура класифікації. Але перед цим систему необхідно «навчити» реагувати на жести, та адаптувати їх щодо конкретних рухів рук користувача. В результаті навчання система поступово здобуває здатність відповідати потрібними реакціями на певні

сукупності зовнішніх впливів, а в результаті адаптації відбувається підстроювання параметрів і структури системи з метою досягнення необхідної якості керування в умовах безперервних змін зовнішніх умов.

У даній статті проаналізовано існуючі методи та моделі взаємодії жестів людини з комп'ютером.

Kinect[1] — безконтактний сенсорний ігровий контролер, спочатку представлений для консолі Xbox 360, і значно пізніше для персональних комп'ютерів під керуванням ОС Windows. Kinect дозволяє користувачеві взаємодіяти з нею без допомоги контактної ігрової контролера через усні команди, пози тіла, об'єкти або малюнки.

Діапазон глибини та програмне забезпечення дозволяють автоматично калібрувати датчик з урахуванням умов гри і навколишніх умов, наприклад меблів, що знаходяться в кімнаті.

Leap Motion. Датчики руху зробили великий стрибок вперед завдяки Leap Motion[2]. Leap Motion створив новий пристрій, який використовує датчик руху. Все, що необхідно для управління комп'ютером є те, чим ви користуєтесь щодня - ваші руки.

Програмне забезпечення Leap Motion отримує дані з сенсорів і аналізує їх з врахуванням анатомії кистей, пальців, зап'ясть а також геометрії сторонніх інструментів. ПЗ містить модель людської руки і порівнює її з отриманими даними, визначаючи найкращу відповідність між ними. Сенсорні дані аналізуються покадрово і надсилаються драйвером у додатки, які підтримують Leap Motion. Отриманий додатком об'єкт Frame містить усі знайдені позиції, швидкості та сутності (кисті, пальці, предмети). Такі концепції, які охоплюють декілька кадрів (жести, рухи тощо), також оновлюються з кожним кадром.

Програмне забезпечення Leap Motion працює як сервіс (операційна система Windows) або демон (Mac і Linux) на клієнтському комп'ютері.

Нативні додатки з підтримкою Leap Motion можуть під'єднатися до драйвера з використанням динамічних бібліотек Leap Motion (які надаються як частина набору засобів розробки (SDK) Leap Motion). Веб-додатки можуть приєднатися до WebSocket-сервера підтримуваного драйвером. WebSocket передає відстеженні дані як повідомлення у форматі JSON, одне повідомлення на кадр. JavaScript-бібліотека LeapJS надає SDK, який інкапсулює ці дані.

Intel Perceptual Computing SDK. Intel Perceptual Computing SDK - це нова технологія, яка дозволить користувачеві взаємодіяти зі своїм мобільним пристроєм за допомогою голосу, рук, обличчя, тобто більш природних для людини способів взаємодії. В даний час компанія Intel веде роботу в цьому напрямку, на сайті компанії з'явилася перша версія perc sdk, яка підтримувала такі можливості:

- розпізнавання жестів, положення рук, пальців, рівень відкритості долоні;
- відстеження положення обличчя, визначення контрольних точок (губи, ніс, очі), ідентифікація («впізнання») особи;
- відстеження положення 2-х і 3-х мірних об'єктів.

Nintendo Wii Remote. Wii Remote - основний контролер для гральної консолі Wii компанії Nintendo. Головною особливістю Wii Remote в порівнянні з конкурентами є детектор руху, що дозволяє гравцям управляти персонажами гри і предметами на екрані рухом руки, або «вказуючи» на об'єкти, що відображаються на екрані. Це забезпечується роботою акселерометра і світлочутливої матриці. Крім цього, до контролера можна підключати різні пристрої, розширюючи таким чином його можливості.

Розпізнавання жестів по WiFi. Не секрет, що випромінювання WiFi дозволяє використовувати високоточні радары для вимірювання руху людей. На цьому принципі був заснований перший пасивний WiFi-радар, створений фахівцями з Лондонського університетського коледжу. Радар сканує приміщення крізь стіни і визначає точне місце розташування людей. Передбачалося, що пристрій стане в нагоді правоохоронним органам і армії.

Таким чином, достатньо всього лише точно реєструвати частоту відбитого сигналу і час його повернення, щоб знати швидкість і відстань до об'єкта, що рухається. Якщо навчитися робити це досить точно, то можна навіть реєструвати пульс і частоту дихання людини.

В даний момент система WiSee розпізнає дев'ять жестів з точністю 94%

В експериментальній установці приймач зібраний з декількох універсальних приймачів USRP-N210, кожен працює на своїй частоті, сигнал обробляється в програмі GNURadio.

Ознайомившись з системами які є на ринку, можна помітити те, що, взагалі не так багато рішень які вирішують проблеми взаємодії людини з комп'ютером. Навіть ті що є, важко порівняти, тому що, кожна з цих систем вирішує якусь одну проблему.

Kinect дуже якісна система, але на даному етапі свого розвитку вона має не так багато функцій. Основні це: skeletal tracking, seated skeletal tracking та face tracking.

Трекінг людини - це не просто розпізнавання людини на кадрі, а так-же спостереження за рухом. Kinect може відстежувати до шести осіб, але тільки для двох будувати скелет. Ми отримуємо ключові точки людини, тобто скелет.

Якщо вам потрібно специфічні функції їх можна реалізувати на базі системи Kinect, використовуючи відкриту бібліотеку комп'ютерного зору – OpenCV. Використовуючи офіційне SDK не можна буде розпізнати жести руки, можливо тільки отримати її координати. Алгоритми розпізнавання окремих жестів потрібно реалізувати окремо.

Leap Motion спрямована на відстеження мікро-жестів руки. Як на мене це дуже прогресуюча розробка для відстеження руки, але вона не може працювати на великих відстанях, руки потрібно тримати над контролером.

На основі методів які були досліджені можна зробити деякі висновки, а саме, що серед них не має універсального методу, кожен із них має плюси та мінуси. Універсальний метод який пропонується у даній статті можна описати як, комбінування переваг декількох з методів описаних раніше.

Розіб'ємо пропонуємий метод на основні складові:

- сегментація зображення та виділення руки на карті глибини;
- отримання дескрипторів які описують жест;
- зібрати статистику дескрипторів по кожному жесту;
- використовуючи один з методів машинного навчання обучити систему та отримати модель розпізнавання.

Сегментація - це процес розділення цифрового зображення на декілька сегментів.

Один з основних і простих способів - це побудова сегментації за допомогою порога. Поріг - це ознака (властивість), яка допомагає розділити шуканий сигнал на класи.

Карта глибини – це монохромне зображення в якому вихідне діапазонне зображення, містить в якості значень пікселів відстань до конкретних точок зображення.

Робота з картою глибини потребує другого підходу ніж робота з звичайним зображенням. Будемо використовувати дві стратегії для отримання руки на карті глибини. Перша стратегія – це використовувати діапазони, тобто розпізнавати жест будемо в заданому діапазоні l_1, l_2 . Де l_1 – ближня відстань до камери, а l_2 – дальня відстань. Все що буде в цьому діапазоні будемо вважати рукою. В результаті отримаємо карту глибини на якому лише одна рука

Після етапу сегментації ми отримаємо карту глибини на якій лише рука, тепер потрібно дістати дескриптор. Дескриптор – це вектор $X \in R^2$ в просторі, який однозначно описує жест, координати якого є ознаками (атрибутами) жесту.

Для знаходження контурів використовують ознаки зв'язності. Для побудови зв'язної моделі можна скористатися поняттями "сусідства". Відомо, що при квадратному растрі можлива чотиризв'язність (тобто сусідніми вважаються елементи, що прилягають до сторін); восьмизв'язність (тобто елементи, що дотикаються у кутах, також вважаються "сусідами").

Одиничний елемент $b(m, n)$ бінарної матриці $B(M, N) \in \mathbb{Z}^{M \times N}$ зв'язаним, як що хоча б один з сусідніх з ним елементів $b(m, n + 1), b(m + 1, n), b(m + 1, n + 1), b(m - 1, n), b(m, n - 1), b(m - 1, n - 1), b(m + 1, n - 1), b(m - 1, n + 1)$ є також одиничним, де m, n - координати елементу b . Потрібно знайти всі зв'язні точки в яких хоча б один сусід не належав області.

Для знаходження опуклої оболонки використаємо алгоритм швидкої оболонки, тому що його легко розпаралелити та узагальнити на довільну розмірність. В середньому складність алгоритму буде $O(n \cdot \log(n))$, якщо кількість елементів підзадач буде лінійно зменшуватись зі сталим коефіцієнтом $k < 1$. В найгіршому випадку швидкість буде $O(n^2)$.

Момент - це характеристика контуру, об'єднана з усіма пікселями контуру. Момент (p, q) визначається як:

$$m_{p,q} = \sum_{i=1}^n l(x, y) x^p y^q \quad (1)$$

Отже маючи ознаки які були перераховані вище можемо сформулювати вектор (дескриптор).

Використаємо «Дерево рішень» як апарат машинного навчання. Маємо деяку кількість жестів які необхідно розпізнати, кожен жест – це один клас зі своїми характеристиками.

Сформулюємо математичку постановку задачі: Нехай Y - простір жестів; y - жест, $y \in Y$; $m = \{1, 2, \dots\}$ - номери жестів; X - простір ознак, тобто векторний простір, точками якого є вектори ознак образів;

Існує невизначена залежність - відображення $y^*: X \rightarrow Y$, значення якої відомі тільки на об'єктах кінцевої навчальної вибірки. Потрібно побудувати алгоритм $a: X \rightarrow Y$, здатний класифікувати довільний жест $x \in X$. В якості даного алгоритму будемо використовувати дерева рішень.

Під терміном "Дерева рішень" мається на увазі сімейство алгоритмів, заснованих на створенні ієрархічної структури, яка базується на відповіді "Так" або "Ні" на набір питань.

Деякі жести мають різну кількість характеристик, тому було прийнято рішення розділити жести на групи, простір груп – G , де в кожній групі лише ті жести, для яких кількість характеристик однакова. Потрібно задати функцію $g: Y \rightarrow G$, яка буде відображати простір жестів на простір груп. Для кожної групи будемо окремо дерево, тобто тепер добавимо простір який складається з окремих дерев.

Для реалізації задачі будемо використовувати мову програмування C++ та бібліотеку OpenCV[6].

OpenCV - Open Source Computer Vision Library, бібліотека комп'ютерного зору з відкритим вихідним кодом. Бібліотека функцій та алгоритмів комп'ютерного зору, обробки зображень і чисельних алгоритмів загального призначення.

Реалізація складається з двох модулів:

- за допомогою першого модуля ми отримаємо дані для кожного жесту;
- за допомогою другого модуля використаємо дані с закономірностями та побудуємо модель закономірностей для використання її при класифікації жестів.

На рисунку 1 зображено класи використані у розробці:

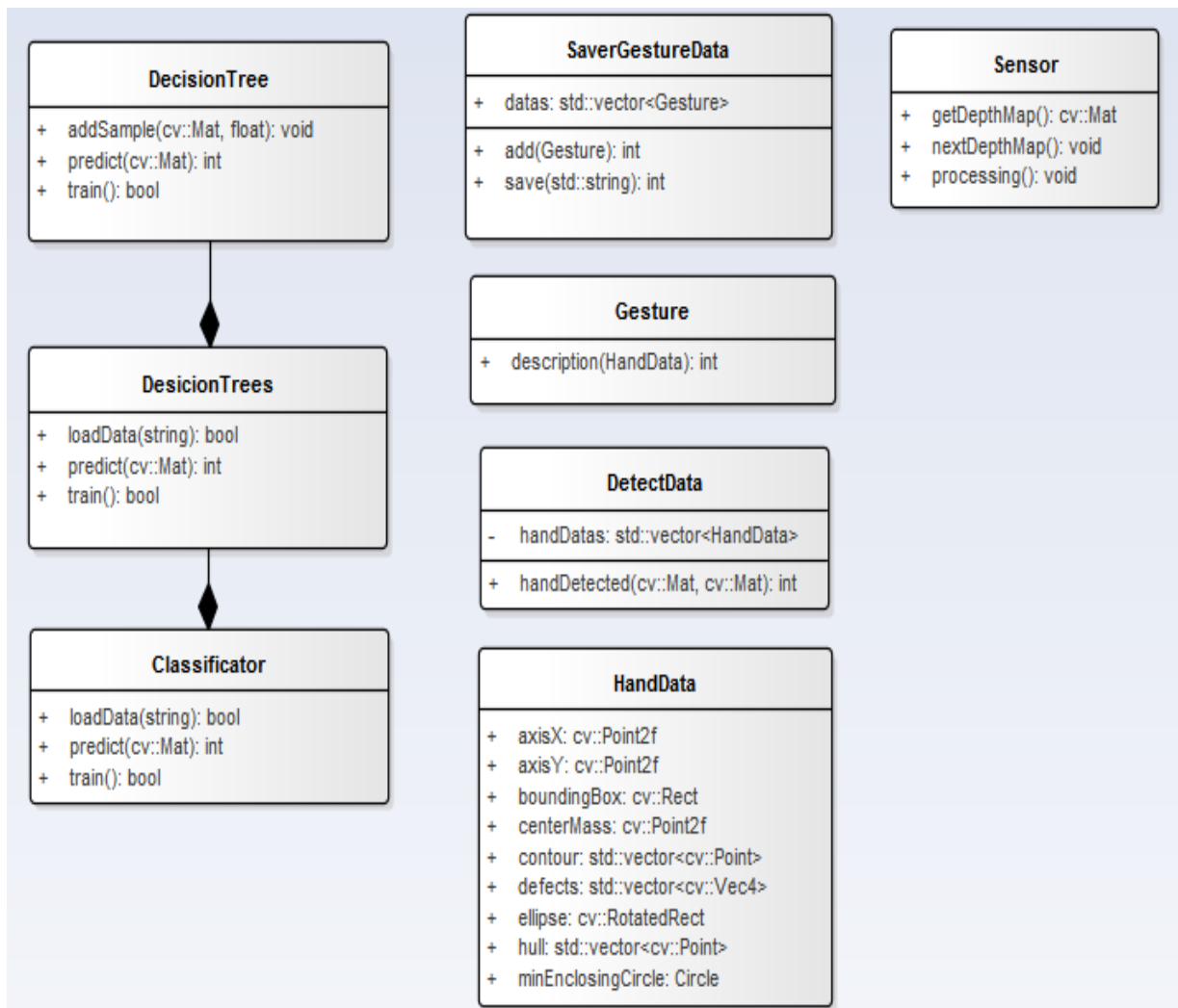


Рис. 1 – Класи використані у розробці

Деякі класи використаємо в обох модулях, для цього необхідно зробити один виконавчий файл, та за допомогою прапору (-train) який він приймає, може працювати в двох режимах – отримання даних за жестом та навчання з розпізнаванням.

Клас Gesture перетворює дані отримані за допомогою класу DetectData, та на виході дає дескриптор, який описує жест.

Після того як получили об'єкт класу Gesture, можемо використовувати ці дані для створення закономірностей чи використовувати класифікатор для розпізнавання жесту який описує цей дескриптор.

В якості камери глибини використаємо камеру Kinect.

Для роботи з камерою, потрібно завантажити SDK[7] для неї з офіційного сайту Microsoft. Після завантаження та установки SDK можна працювати з камерою. В SDK входить драйвер та приклади по роботі з камерою.

Щоб підключити камеру до свого проекту, потрібно прописати шлях до головних файлів с розширенням *.h, які знаходяться в папці inc. Та прописати шлях до бібліотек dll, які знаходяться в папці lib.

Для того, щоб отримувати зображення з Kinect необхідно ініціалізувати потік захоплення зображення. Для цих цілей використовується метод NuiImageStreamOpen, який у якості параметрів приймає:

- тип зображення (кольорове зображення, буфер глибини і т.д.);
- здатність (від 80x60 до 1280x1024);
- прапори обробки зображення (в beta-версії SDK не використовується);

- кількість кешованих кадрів (максимальне значення NUI_IMAGE_STREAM_FRAME_LIMIT_MAXIMUM на даний момент дорівнює 4);

- дескриптор події, яка буде виникати при отриманні нового кадру (необов'язковий параметр, але на ділі виявилось, що якщо передати NULL, то потік захоплення може і не запуститися);
- покажчик на змінну, в яку буде записаний дескриптор потоку захоплення зображення при успішному завершенні роботи функції.

Послідовність дій захоплення зображення які було виконано при розробці:

- для того, щоб почати захоплення карти, потрібно ініціалізувати пристрій методом NuiInitialize;
- стартувати потік захоплення за допомогою методу NuiImageStreamOpen;
- при отриманні нового зображення викликається подія, дескриптор якої ми передали в NuiImageStreamOpen;
- отримуємо фрейм за допомогою методу NuiImageStreamGetNextFrame;
- виконуємо захоплення буфера за допомогою методу LockRect;
- пройдемо по буферу і отримаємо колір кожного пікселя, нормалізував значення глибини.
- звільняємо буфер за допомогою методу NuiImageStreamReleaseFrame;
- для того, щоб зупинити захоплення зображень з пристрою, треба звільнити його методом NuiShutdown.

Після того, як за допомогою першого модуля нашої програми ми отримали інформацію по кожному жесту, необхідно зберегти цю вибірку в файл. В кожній строчці якого дескриптор, та номер жесту. Для кожного жесту маємо більше ніж 40 таких строчок. В результаті вибірка була з 500 строк.

На рисунку 2 показані жести, які були розпізнані за допомогою програми.

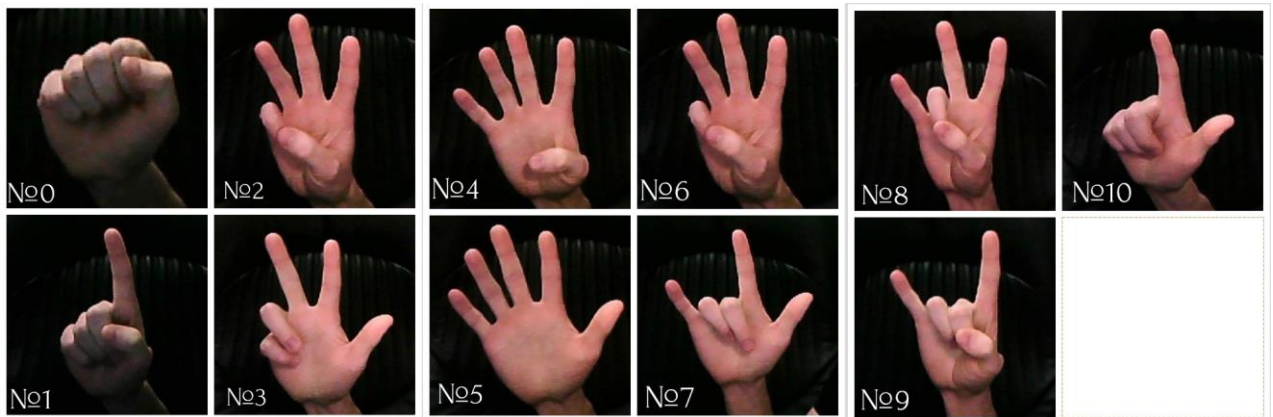


Рис. 2 – Жести які були розпізнані

Перевірка відбувалась таким чином: запускаємо програму та перед камерою показуємо жест, програма виводить той жест, який вона розпізнала, звіряємо правильність розпізнаного жесту. Потім показуємо довільний жест та знову показуємо перед камерою жест який необхідно розпізнати. Таких повторень було від 20 до 30, з різним нахилом жесту. Так як, це камера глибини то немає ніяких обмежень на умови використання навіть в темряві. Результати розпізнавання (таблиця 1.1).

Таблиця 1.1

Таблиця результатів

| Номер жесту | Точність розпізнавання |
|-------------|------------------------|
| 0 | 100% |
| 1 | 100% |
| 2 | 100% |
| 3 | 100% |
| 4 | 100% |
| 5 | 100% |
| 6 | 100% |
| 7 | 100% |
| 8 | 100% |
| 9 | 100% |
| 10 | 100% |

В результатах по перевірці точності програми показано номер жесту, та точність розпізнавання. Стовідсоткова ймовірність розпізнавання означає, що значення дескрипторів для різних жестів не пересікаються між собою.

Можна зробити одне зауваження, використовувалися жести тільки для правої руки і в програмі немає можливості розпізнати – права чи ліва це рука.

Наведемо опис дескрипторів, які було використано при класифікації для деяких із жестів (Таблиця 1.2, Таблиця 1.3).

Таблиця 1.2

Дескриптор для жесту №2

| Змінна дескриптора | Значення дескриптора |
|--------------------|----------------------|
| X_1 | 40.2854 |
| X_2 | 27.4925 |
| X_3 | 1 |
| X_4 | 0.896548 |
| X_5 | 12.8901 |

Таблиця 1.3

Дескриптор для жесту №9

| Змінна дескриптора | Значення дескриптора |
|--------------------|----------------------|
| X_1 | 71.7708 |
| X_2 | 42.2682 |
| X_3 | 1 |
| X_4 | 0.872412 |
| X_5 | 15.7692 |

Перед використання програми корегуємо вибірку так, щоб кожний дескриптор мав однакову дожину, записавши нулі в пропущених місцях. В вибірці 322 строки. Використовуючи програму Deductor будемо дерево рішень для вибірки.

Отримуємо 21 вузол та 11 правил (Рисунок 3). Після того як побудували дерево рішень, ми можемо проаналізувати на скільки добре можна використовувати дерево для класифікації та на скільки добре підходить наша вибірка.

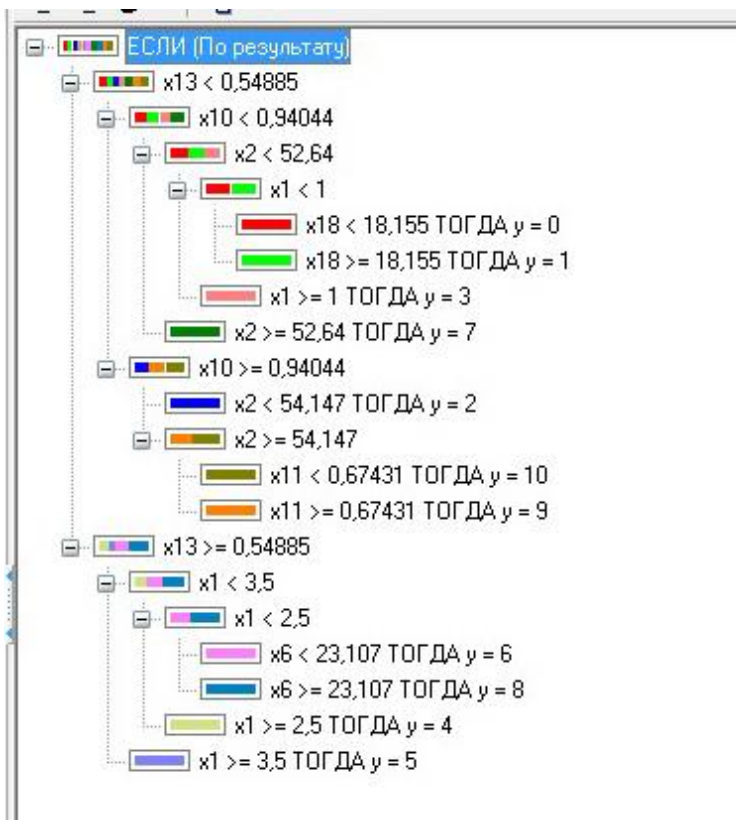


Рис. 3 – Вузли дерева

В результаті отримуємо 11 правил (Таблиця 1.4) за допомогою яких можемо однозначно класифікувати жести. Ймовірність збігається з той ймовірністю, яку ми одержали в своїй програмі. Це підтверджує коректність нашої роботи.

Правила

| № | Умови | Слідство(y) | Підтримка | | Достовірність | |
|----|--|-------------|-----------|-----------|---------------|-----------|
| | | | % | Кількість | % | Кількість |
| 1 | $x_{13} < 0,54885$ И $x_{10} < 0,94044$ И $x_2 < 52,64$ И $x_1 < 1$ И $x_{18} < 18,155$ | 0 | 9,12 | 28 | 100,00 | 28 |
| 2 | $x_{13} < 0,54885$ И $x_{10} < 0,94044$ И $x_2 < 52,64$ И $x_1 < 1$ И $x_{18} \geq 18,155$ | 1 | 9,12 | 28 | 100,00 | 28 |
| 3 | $x_{13} < 0,54885$ И $x_{10} < 0,94044$ И $x_2 < 52,64$ И $x_1 \geq 1$ | 3 | 7,82 | 24 | 100,00 | 24 |
| 4 | $x_{13} < 0,54885$ И $x_{10} < 0,94044$ И $x_2 \geq 52,64$ | 7 | 10,10 | 31 | 100,00 | 31 |
| 5 | $x_{13} < 0,54885$ И $x_{10} \geq 0,94044$ И $x_2 < 54,147$ | 2 | 8,47 | 26 | 100,00 | 26 |
| 6 | $x_{13} < 0,54885$ И $x_{10} \geq 0,94044$ И $x_2 \geq 54,147$ И $x_{11} < 0,67431$ | 10 | 10,47 | 33 | 100,00 | 33 |
| 7 | $x_{13} < 0,54885$ И $x_{10} \geq 0,94044$ И $x_2 \geq 54,147$ И $x_{11} \geq 0,67431$ | 9 | 9,12 | 28 | 100,00 | 28 |
| 8 | $x_{13} \geq 0,54885$ И $x_1 < 3,5$ И $x_1 < 2,5$ И $x_6 < 23,107$ | 6 | 10,10 | 31 | 100,00 | 31 |
| 9 | $x_{13} \geq 0,54885$ И $x_1 < 3,5$ И $x_1 < 2,5$ И $x_6 \geq 23,107$ | 8 | 14,01 | 43 | 100,00 | 43 |
| 10 | $x_{13} \geq 0,54885$ И $x_1 < 3,5$ И $x_1 \geq 2,5$ | 4 | 7,17 | 22 | 100,00 | 22 |
| 11 | $x_{13} \geq 0,54885$ И $x_1 \geq 3,5$ | 5 | 4,23 | 13 | 100,00 | 13 |

На рисунку 4 зображено які атрибути більш значимі а які менш.

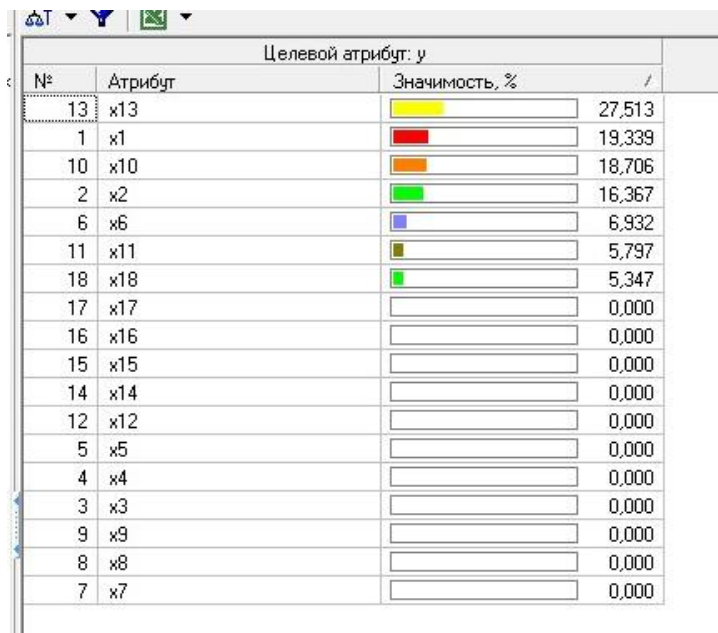


Рис. 4 – Значимість атрибутів

Висновки. Досліджено роботу систем управління жестами як перспективного інтерфейсу людина-машина. Такі системи дозволяють реалізувати дистанційне управління автоматизованими пристроями не тільки звичайними користувачами, а й людьми з вадами слуху і мови.

Аналіз показав, що це бурно розвиваюча технологія. Можливо в недалекому майбутньому, це новий крок в еволюції інтерфейсу для взаємодії з комп'ютером. Було проведено аналіз існуючих рішень в цій області. Kinect та Leap Motion одні з таких систем, які показали, що не тільки теоретично можливо створити такий інтерфейс, а що можливо і на готових робочих приладах.

Було проаналізовано сенсори для отримання карти глибини. Використовуючи карту глибини було розроблено та реалізовано метод розпізнавання жестів руки оснований на дескрипторах та з використанням машинного навчання. Не зважаючи на простоту метода, він дає гарний результат для заданих жестів.

Література

1. *Опис сенсора Kinect*[Електронний ресурс] –Режим доступу: [www/ URL : http://uk.wikipedia.org/wiki/Kinect/](http://uk.wikipedia.org/wiki/Kinect/) - 15.09.2018
2. *Офіційний сайт Leap Motion*[Електронний ресурс] – Режим доступу: [www/ URL : https://www.leapmotion.com/](https://www.leapmotion.com/) - 15.09.2018
3. *Опис технології WiSee*[Електронний ресурс] – Режим доступу:[www/ URL : http://wisee.cs.washington.edu/wisee_paper.pdf](http://wisee.cs.washington.edu/wisee_paper.pdf) - 15.09.2018
4. *Oikonomidis I. Efficient Model-based 3D Tracking of Hand Articulations using Kinect. [Text] / I.Oikonomidis, N.Kyriazis, A.A.Argyros In BMVC 2011. BMVA, 2011.*
5. *Шапиро, Л. Компьютерное зрение [Текст] / Л. Шапиро, Дж. Стокман; Пер. с англ. – М.: Бином. Лаборатория знаний, 2006. – 752 с.*
6. *Офіційний сайт документації бібліотеки OpenCV*[Електронний ресурс] – Режим доступу: [www/ URL : http://docs.opencv.org/](http://docs.opencv.org/) - 15.09.2018
7. *Сайт загрузки SDK Kinect*[Електронний ресурс] – Режим доступу: [www/ URL : https://www.microsoft.com/en-us/kinectforwindows/](https://www.microsoft.com/en-us/kinectforwindows/) - 15.09.2018

References

1. *Description of Kinect Sensor [Electronic resource] -Access mode: www / URL: http://uk.wikipedia.org/wiki/Kinect/ - 09/15/2018*
2. *Official site Leap Motion [Electronic resource] - Access mode: www / URL: https://www.leapmotion.com/ - 09/15/2018*
3. *Description of the WiSee technology [Electronic resource] - Access mode: www / URL: http://wisee.cs.washington.edu/wisee_paper.pdf - 09/15/2018*
4. *On-line I-Effective Model-Based 3D Tracking of Hand Articulations using Kinect. [Text] / I.Oikonomidis, N.Kyriazis, A.A.Argyros In BMVC 2011 BMVA, 2011.*
5. *Shapiro, L. Computer vision [Text] / L. Shapiro, J. Stokman; Per. from english - M .: Bean. Laboratory of Knowledge, 2006. - 752 p.*
6. *Official OpenCV Library Documentation Website [Electronic Resource] - Access Mode: www / URL: http://docs.opencv.org/ - 09/15/2018*
7. *Kinect SDK Download Page [Electronic resource] - Access Mode: www / URL: https://www.microsoft.com/en-us/kinectforwindows/ - 09/15/2018*

Барбарук Віктор Миколайович – к.т.н., доц., директор Центру удосконалення освіти СЧУ ім. В. Даля barbaruk.viktor@gmail.com

Швачка Олександр Олександрович – студент групи КІ -17дм, Східноукраїнський національний університет ім. В. Даля. ptichkd@mail.ru