

ІНФОРМАЦІЙНА СИСТЕМА ПРОВЕДЕННЯ ПРОЦЕСУ LIVENESS DETECTION ЗА ДОПОМОГОЮ AWS

Анотація. У роботі досліджено процес проведення відеоідентифікації особи за допомогою процесу Liveness Detection. Розроблено алгоритм проведення процесу Liveness Detection на основі виконання випадкових завдань особою, що ідентифікується. Проведено дослідження API Amazon DetectFaces, у процесі якого, емпіричним шляхом, були встановлені порогові значення характеристик обличчя на зображенні для проведення процесу Liveness Detection. На основі проведених досліджень розроблена програмна система біометричної відеоідентифікації людини, яка дозволяє автоматично встановлювати присутність особи перед відеокамерою і ідентифікувати її, запобігаючи використанню фальшивих артефактів з метою підробки особистості людини.

Ключові слова: Liveness Detection, відеоідентифікація, Amazon Web Services, розпізнавання обличчя.

Постановка проблеми. Дистанційне проведення процесу відеоідентифікації стає все більш актуальним у нашій країні. З одного боку, поширення у світі вірусу COVID-19 викликало різку зміну патернів споживчої поведінки. Зростає частка тих споживачів, які свідомо чи вимушено відмовляються від фізичного контакту з оточуючими. З іншого боку, велика кількість людей стикається з різними проблемами у зв'язку з переселенням та релокацією підприємств із територій, де ведуться активні бойові дії. Наприклад, людина, яка бажає відкрити особистий рахунок чи взяти кредит, не завжди може знайти відділення банку, де операціоніст зможе перевірити схожість людини з її документами. Або громадянин, який звертається для надання деяких електронних державних послуг, наразі, на певних етапах виконання його запиту, повинен бути особисто присутнім у відповідних державних органах, щоб підтвердити свою особистість.

Одним з шляхів вирішення проблеми дистанційної відеоідентифікації особи є процес Liveness detection [1], який дозволяє відрізнити реальну людину від штучної репродукції її зовнішності. При цьому підходить система, яка вико-

ристовує Liveness detection, повинна взаємодіяти із особою таким чином, щоб значно утруднити підробку її особистості хакером або шахраєм.

Робота має високу актуальність і практичну значимість, через що було прийнято рішення про розробку власного алгоритму проведення процесу Liveness detection, який дозволить дистанційно проаналізувати і оцінити присутність особи, що ідентифікується, та побудувати на його основі програмну систему відеоідентифікації співробітників підприємства, відкритість коду якої дозволить вільно, за потребою, повторно використовувати або змінювати її компоненти при реалізації процесу Liveness detection у будь-яких системах відеоідентифікації людини.

Аналіз останніх досліджень і публікацій. Загальний розвиток комп'ютерних технологій і вдосконалення алгоритмів роботи штучного інтелекту, протягом останніх кількох десятиріч, дозволили розробити безліч алгоритмів розпізнавання обличчя, які умовно можна поділити на три типи, залежно від того, як вони ідентифікують обличчя [2]: на основі зовнішнього вигляду (функції), який використовує повні текстурні функції; розпізнавання обличчя на основі підпростору; прийоми використання нейронних мереж. Конкретні рішення для проведення процесу Liveness detection є закритими для вільного використання та представлені каталогом комерційного API, які базуються на машинному навчанні та використовують нейронні мережі для виконання завдань, такі, як платформа ідентифікації Acuant [3], Kairos Face Recognition [4], API Anometrics Face Recognition [5], API Lambda Labs [6]. Проведений аналіз сучасного ринку програмних і алгоритмічних рішень для виконання процесу Liveness detection показав, що існуючи наразі підходи є цілком комерційними рішеннями із закритими алгоритмами своєї роботи. Окрім цього, алгоритм Liveness detection ще не є стандартизованим і має безліч варіантів реалізації, які іноді можуть приводити до неоднозначних результатів [7]. Саме тому, було прийнято рішення про розробку власного алгоритму та системи Liveness detection на основі отримання характеристик обличчя за допомогою AWS API [8], оскільки цей сервіс пропонує високу точність розпізнавання обличчя, яка складає 99.99% і щомісячно надає безкоштовні 10000 запитів для використання, чого достатньо для виконання мети нашої роботи.

Мета дослідження: розробка алгоритму проведення процесу Liveness Detection за допомогою AWS та побудова системи відеоідентифікації на його основі.

Основна частина: Найпершим кроком нашої роботи, що має бути виконаний для розробки системи відеоідентифікації особи, є розробка алгоритму процесу Liveness detection. Його головною метою є конкретизація шагів комунікації програмної системи із користувачем, з метою отримання висновку щодо фізичної присутності особи, яка проходить відеоідентифікацію, перед камерою. Розглянемо створений алгоритм, зображений на рис. 1. більш детально.

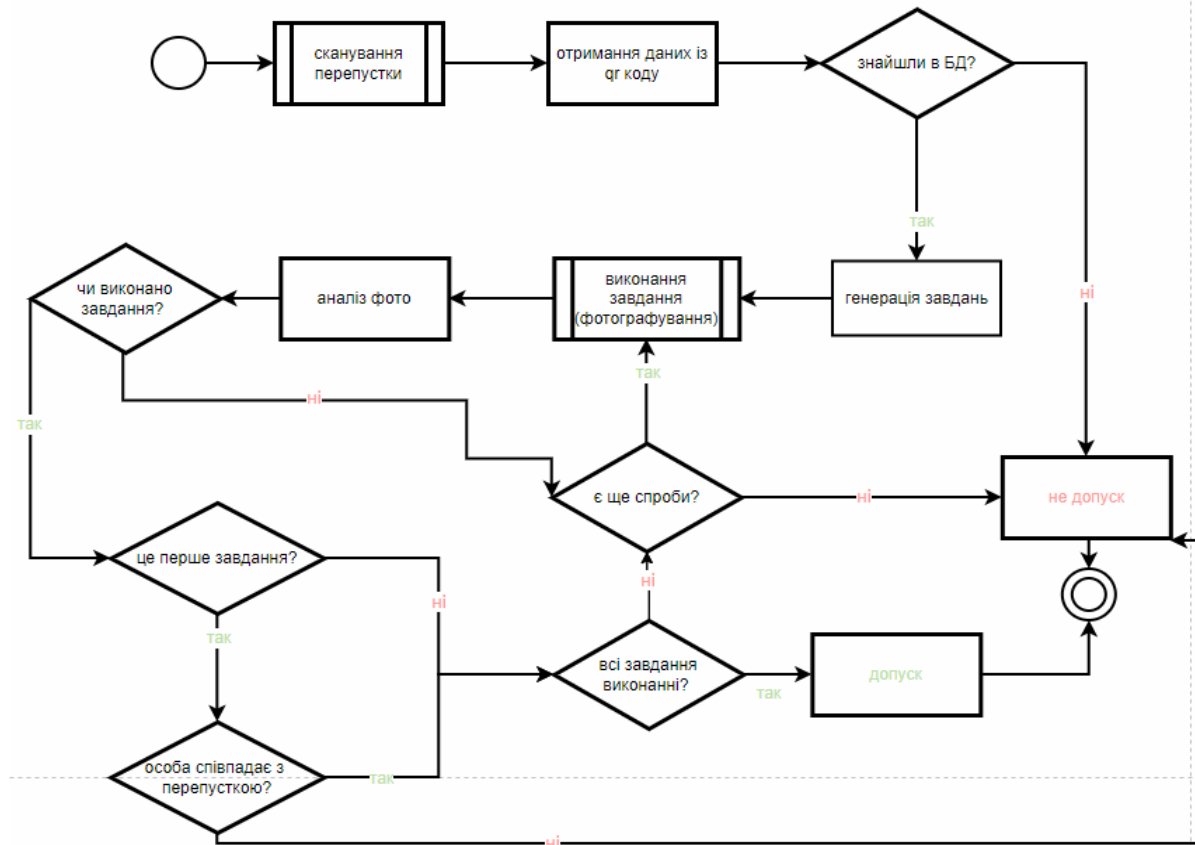


Рисунок 1 - Алгоритм виконання процесу Liveness Detection

Найпершим кроком розробленого алгоритму є отримання еталонного зображення для ідентифікації особи. Таким зображенням може бути фото з документів користувача, або, як зроблено у нашій системі - ідентифікатор перепустки, зашифрований у вигляді QR-коду, який генерується заздалегідь та містить в собі логін співробітника, до якого прив'язана його еталона центральна фотографія.

Відповідно до цього, другим та третім кроком є розшифровка QR-коду з перепустки та пошук розшифрованої інформації у базі даних системи. Якщо працівника за отриманим логіном не знайдено, це значить, що перепустка ви-

явилася підробленою та алгоритм не допускає такого співробітника, завершаючи таким чином процес відеоідентифікації.

Якщо дані перепустки були знайдені, на четвертому кроці алгоритму виконується генерація трьох завдань, націлених на отримання положень голови особи, яка ідентифікується. Першим завданням є отримання центральної фотографії особи, два інших обираються випадково з масиву заздалегідь доступних варіантів завдань.

П'ятим та шостим кроком є безпосереднє фотографування особи, з подальшим аналізом фотографії на положення голови та його відповідності поточному завданню.

На сьомому кроці відбувається перевірка, чи виконане поточне завдання. Якщо ні, то кількість доступних спроб зменшується на одну. Якщо у працівника ще є доступні спроби, то процес переходить до п'ятого кроку, тобто на процес фотографування. Якщо завдання було виконано, система перевіряє, чи співпадає особа на фотографії та на перепустці. Якщо відсоток співпадіння занадто малий, то процес буде завершено з не допуском до підприємства. Якщо особа пройшла перевірку, то надається змога виконати останні два завдання на положення голови, після чого система повертається на п'ятий крок.

Наступним кроком нашої роботи є дослідження можливості використання просторових характеристик обличчя для аналізу положення голови від API AWS, оскільки сам по собі API AWS не надає функціоналу для проведення Liveness detection. З безлічі можливих характеристик положення голови та обличчя на фотографії були обрані наступні: *isHead*, який відповідає за наявність та кількість осіб на фотографії, *brightness*, який відповідає за яскравість фотографії; *distanceHead*, який відповідає за відстань особи до веб-камери; *sunglasses*, який відповідає за наявність сонцезахисних окулярів на обличчі; *posX*, відповідальний за розташування особи щодо осі координат *OX*; *posY*, відповідальний за розташування особи щодо осі координат *OY*; *sharpness*, відповідальний за якість зображення, а саме за його чіткість; *posYaw*, відповідальний за поворот голови по сторонах; *posPitch*, який відповідає за нахил голови вгору і вниз; *posRoll*, відповідальний за нахил верхівки по сторонах; *eyesOpen*, який відповідає за стан очей, а саме, чи є вони відкритими; *mouthClosed*, який відповідає за стан рота, а саме, чи закритий він.

Для того, щоб зрозуміти, чи є дане положення голови користувача вірним, було виконане дослідження підібраних лімітів для кожного параметра окремо, емпіричним шляхом. Після опрацювання кожного з параметрів було виділено

два типи граничних значень: значення, які підходять для кожного з п'яти завдань і значення, які унікальні для певного типу завдання. У першу категорію відносяться параметри, що відповідають за кількість осіб на фотографії, освітлення і чіткість зображення. Значення параметра *isHead* розраховується за розміром масиву, що містить аналіз кожного розпізнаного обличчя сервісом AWS. Якщо розмір масиву дорівнює одиниці, це означає, що у кадрі виявлено лише одне обличчя, що саме і задовольняє систему. Параметр, що відповідає за яскравість, може набувати значення від 0 до 100 (включно), де більш високе значення вказує на яскравіше зображення обличчя. Проведені експерименти показали допустимий інтервал значень від 50 до 97. Для параметра, що відповідає за чіткість зображення, був підібраний інтервал від 40 до 100 включно. Інші параметри та їх допустимі значення є унікальними для кожного завдання.

Параметр, що відповідає за нахил верхівки може приймати негативне значення, що означає, що нахил виконаний в ліву сторону. І навпаки, при нахилі праворуч значення буде позитивним. Таким чином, для виконання завдання на центральне положення голови допустимими вважаються значення, які знаходяться в діапазоні від -5 до 5. Для решти завдань діапазон був збільшений, оскільки, при повороті всієї голови, наприклад, вліво, сервіс AWS дає похибку у значенні нахилу верхівки. Таким чином, в діапазон включені числа від -20 до 20.

Для того, щоб визначити нахил голови вгору або вниз, необхідно проаналізувати значення *possPitch*. Рух голови для даного параметра показано на рис. 2 вісь *pitch*. Дані значення також можуть бути негативними, що означає, що голова опущена вниз, а якщо значення більше за нуль, то голова піднята вгору. Усі підібрані діапазони прийнятних значень для завдань вказані в таблиці 1.

Таблиця 1

Діапазон підібраних значень параметра *possPitch* для виконання завдань

Тип завдання	Мінімальне значення	Максимальне значення
центральне фото	0	18
подивіться праворуч	-5	18
подивіться ліворуч	-5	18
подивіться вгору	20	50
подивіться вниз	-50	-20

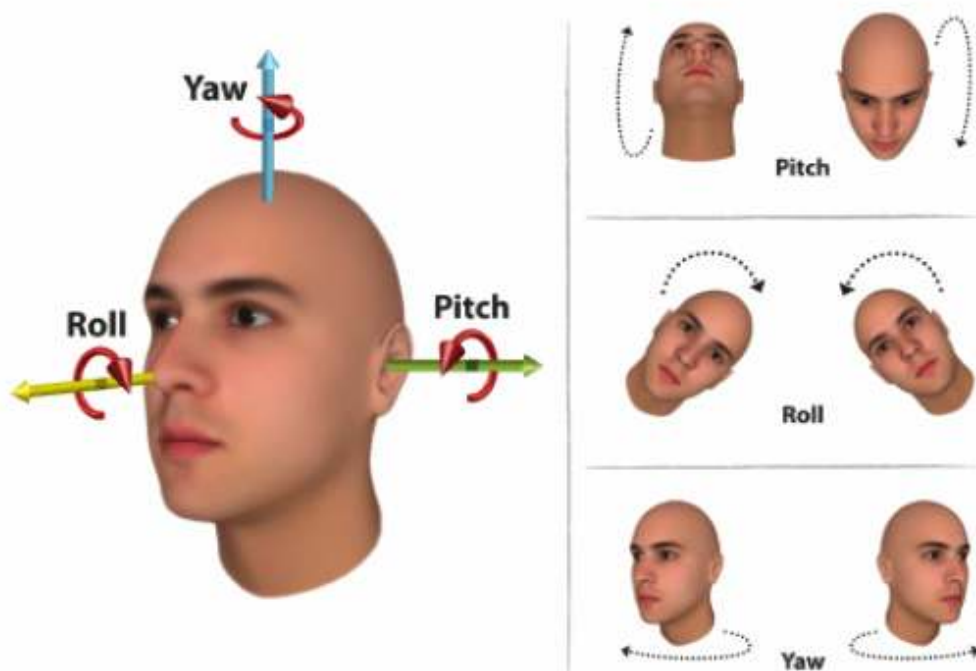


Рисунок 2 - Рухи голови по осях (pose) AWS API

Параметр `poseYaw` відповідає за поворот голови по сторонах. Рух голови для даного параметра показано на рис. 2, вісь Yaw. При повороті голови в ліворуч, значення будуть негативними, при повороті праворуч - позитивними. Усі підібрані діапазони прийнятних значень для завдань вказані в таблиці 2.

Таблиця 2

Діапазон значень параметра `poseYaw` для виконання завдань

Тип завдання	Мінімальне значення	Максимальне значення
центральне фото	-5	5
подивіться праворуч	10	50
подивіться ліворуч	-50	-10
подивіться вгору	-5	5
подивіться вниз	-5	5

Параметри, що відповідають за стан очей, рота та наявність сонцезахисних окулярів містять у собі, крім бульового значення, ще й коефіцієнт упевненості даної відповіді. Наприклад, при аналізі фотографії, де на обличчі знаходяться сонцезахисні окуляри, ми отримали значення параметра *sunglasses*,

який дорівнює *true*, але відсоток точності дорівнює 50%. У такому разі ми не можемо стверджувати, що окуляри дійсно присутні на фотографії. Однак, що стосується стану очей і рота, то нам не важливо в якому вони стані, якщо виконується будь-яке завдання на положення голови, крім центрального. Розглянемо варіанти значень кожного параметра докладніше. Якщо очі відкриті або рот закритий, що задовольняє умові виконання центрального завдання, то у відповіді від сервісу AWS ми отримаємо *true*, проте допустимий відсоток впевненості повинен бути не менше 60. Для інших типів завдань неважливо в якому стані очі і рот на фотографіях, тому що вони все одно не будуть брати участь у порівнянні з даними з бази даних і не зможуть вплинути на результат. Для параметра, що відповідає за наявність сонцезахисних окулярів, прийнятним значенням є *false* із впевненістю у 70 відсотків.

Для того, щоб перевірити правильність розташування обличчя в кадрі і відстань обличчя до веб-камери необхідно більш детально проаналізувати параметр *boundingBox*. Даний параметр відповідає за рамку, що обмежує обличчя навколо, де лівий верхній кут зображення - це початок координат (0,0), а верхнє (top) і ліве (left) значення є відношеннями до загального розміру зображення. Наприклад, якщо вхідне зображення має розмір 700x200 пікселів, а розмір рамки становить 350x50 пікселів, то AWS API повертає ліве значення (left) 0,5 (350/700) і верхнє значення (top) 0,25 (50/200). На рис. 3 зображено розташування рамки по відношенню до всього зображення.

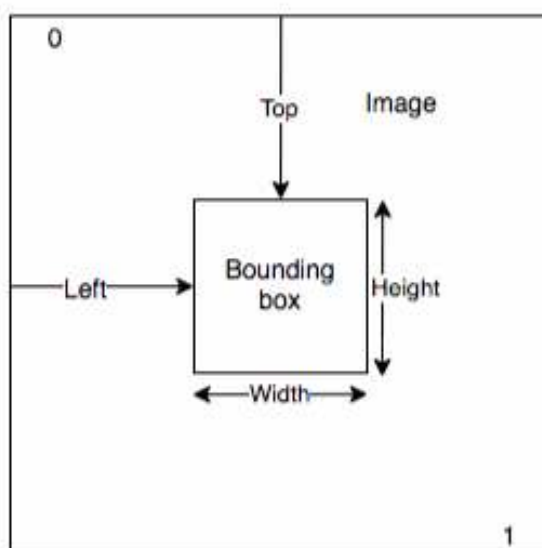


Рисунок 3 - Розташування рамки по відношенню до зображення

Таким чином, для розрахунку відстані від обличчя до веб-камери використовуються значення ширини та висоти рамки, що обмежує обличчя. Допустимі значення ширини лежать у діапазоні від 0.15 до 0.75, а висоти – від 0.2 до 0.65. Значення, які перевищують максимум, означають, що користувач підійшов надто близько до камери, а значення які є менше мінімуму, відповідно означають, що користувач знаходиться занадто далеко від камери. Для визначення розташування особи щодо центру кадру використовуються параметри left і top. Однак при виконанні завдань на повороти голови по сторонах вважається допустимим невеликий зсув від центру кадру убік. Для завдань, де потрібно відтворити нахил голови вгору або вниз, обличчя може бути зміщене щодо центру. У таблиці 3 представлені отримані діапазони значень для кожного із завдань.

Таблиця 3

Діапазон значень параметрів рамки для виконання завдань

Тип завдання	Poss_x min	Poss_x max	Poss_y min	Poss_y max
центральне фото	0.25	0.35	0.1	0.25
подивіться праворуч	0.35	0.65	0.1	0.25
подивіться ліворуч	0.05	0.2	0.1	0.25
подивіться вгору	0.25	0.35	0.2	0.6
подивіться вниз	0.25	0.35	0.02	0.25

Проведене тестування розробленої прикладної системи Liveness Detection на основі створеного алгоритму проходження Liveness та підібраних порогових значень AWS API складалося із 100 тестів різної спрямованості, від прямого проходження user-flow до тестувань із медичною маскою, окулярами і бородою, фотокарткою обличчя і відеозображенням людини з метою підробки її особистості. Завдяки початковій точності розпізнавання характеристик обличчя Amazon API у 99.99% та випадковому підбору завдань у розробленому алгоритмі Liveness Detection всі тести дали очікуваний результат – особа або успішно відеоідентифікувалася на підприємстві, або було встановлено підробку її особистості у ході виконання процесу Liveness Detection.

Висновки. Проведене дослідження порогових значень відповідей AWS API та розроблений у роботі алгоритм Liveness Detection дозволяє автоматично оцінювати, що у ході виконання відеоідентифікації система дійсно взаємодіє із фізично присутньою людиною, що дозволяє боротися із випадками шахрайства у багатьох сферах роботи інформаційних систем – від банального врахування часу роботи на підприємстві до запобігання злому систем безпеки банкінгу та інших фінтех рішень. Для розробки програмної імплементації створеного алгоритму була проведена інтеграція з сервісами отримання даних фотозображень AWS, під час якої, емпіричним шляхом були підібрані коефіцієнти якості вхідного зображення для проведення процесу Liveness Detection.

Врешті, на основі проведених досліджень і створених алгоритмів, була розроблена система відеоідентифікації співробітників підприємства та проведено її тестування, яке показало високі результати роботи створеного програмного продукту. Відкритість алгоритмів і коду системи, а також використання передових технологій при її розробці, дозволяють використовувати наробки, отримані у роботі, для інших програмних систем, де необхідно впроваджувати процес відеоідентифікації особи.

ЛИТЕРАТУРА

1. Apgar D. Survey of Face Liveness Detection for Unsupervised Locations / D. Apgar, M. R. Abid // 2021 IEEE 12th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON) – 2021 – P. 162-168.
2. Datta A. Face detection and recognition: Theory and practice / A. Datta, M. Datta – London: CRC Press, 2016 – 323 p.
3. Use-cases: Acuant [Електронний ресурс] – Режим доступу: <https://www.acuant.com/use-case/>.
4. Get to Know Kairos [Електронний ресурс] – Режим доступу: <https://www.kairos.com/docs/>.
5. Animetrics | Face Recognition [Електронний ресурс] – Режим доступу: <http://animetrics.com/>
6. API Documentation - Lambda Labs [Електронний ресурс] – Режим доступу: <https://lambdal.com/api-documentation.html>
7. Fan Y. Research on Liveness Detection Algorithms Based on Deep Learning / Y. Fan, Y. Shi, X. Wang, H. Yi // 2019 IEEE 10th International Conference on Software Engineering and Service Science (ICSESS) – 2019 – P. 1-6.
8. Amazon CloudFront Documentation [Електронний ресурс] – Режим доступу: https://docs.aws.amazon.com/cloudfront/index.html?nc2=h_ql_doc_cf.

REFERENCES

1. Apgar D. Survey of Face Liveness Detection for Unsupervised Locations / D. Apgar, M. R. Abid // 2021 IEEE 12th Annual Information Technology, Elec-tronics and Mobile Communication Conference (IEMCON) – 2021 – P. 162-168.
2. Datta A. Face detection and recognition: Theory and practice / A. Datta, M. Datta – London: CRC Press, 2016 – 323 p.
3. Use-cases: Acuant [Electronic resource] – Access mode: <https://www.acuant.com/use-case/>.
4. Get to Know Kairos [Electronic resource] – Access mode: <https://www.kairos.com/docs/>.
5. Animetrics | Face [Electronic resource] – Access mode: <http://animetrics.com/>
6. API Documentation - Lambda Labs [Електронний ресурс] – [Electronic resource] – Access mode: <https://lambdal.com/api-documentation.html>
7. Fan Y. Research on Liveness Detection Algorithms Based on Deep Learning / Y. Fan, Y. Shi, X. Wang, H. Yi // 2019 IEEE 10th International Conference on Software Engineering and Service Science (ICSESS) – 2019 – P. 1-6.
8. Amazon CloudFront Documentation [Electronic resource] – Access mode: https://docs.aws.amazon.com/cloudfront/index.html?nc2=h_ql_doc_cf.

Received 29.08.2022.

Accepted 02.09.2022.

The information system for the liveness detection process using aws

Analysis of recent studies and publications. The analysis of the modern market of software and algorithmic solutions for performing the Liveness detection process showed that the currently existing approaches are completely commercial solutions with closed algorithms of their work. In addition, the Liveness detection algorithm is not yet standardized, and has many implementation options that can sometimes lead to ambiguous results [7]. That is why, it was decided to develop our own algorithm and liveness detection system based on obtaining face characteristics using the AWS API [8], because this service offers a high accuracy of face recognition, which is 99.99%, and provides 10,000 free requests to use every month, which enough to fulfill the purpose of our work.

Purpose of the study. Development of the algorithm for the Liveness Detection process using AWS and the construction of a video identification system based on it.

Main research material. This paper investigates the video identification of a person using the Liveness Detection process. The basis of the question, complexity of the Liveness Detection process and the implementation of biometric human video identification

have been studied. An algorithm for conducting the Liveness Detection process based on the execution of random tasks by an identified person has been developed.

Integration with the Amazon DetectFaces API was carried out in order to obtain the character of a digital image of the head, as a result of which it becomes possible to analyze a photo of a person for a wide range of key facial features. A manual test of the Amazon DetectFaces API was conducted, in the process of which, empirically, thresholds of facial characteristics in the image for the Liveness Detection process were set. Integration with the Amazon CompareFaces API was implemented to execute the biometric video identification.

The testing of the developed Liveness Detection application system based on the created Liveness passing algorithm and selected AWS API thresholds consisted of 100 tests of different orientations, from direct user-flow testing to tests with a medical mask, glasses and beard, a photo of a face and a video image of a person for the purpose of forgery his personality. Thanks to the initial accuracy of Amazon API facial recognition of 99.99% and the random selection of tasks in the developed Liveness Detection algorithm, all tests gave the expected result - the person was either successfully video-identified at the enterprise, or his identity was faked during the Liveness Detection process.

Conclusions. Based on the research, the software system of biometric human video identification has been developed, which allows to automatically establish the presence of a person in front of a video camera and identify it, preventing the use of fake artifacts to falsify a person's identity.

Яковлєва Марія Антонівна - студентка спеціальності «Ком'ютерні науки», кафедра інформаційних технологій. Університет імені Альфреда Нобеля.

Ковилін Єгор Романович - к.т.н., старший викладач кафедри інформаційних технологій Університет імені Альфреда Нобеля.

Yakovlieva Mariia Antonivna - student of the "Computer Science" specialty, Department of Information Technologies. Alfred Nobel University.

Kovylin Yehor Romanovych - Ph.D., Senior lecturer at the Department of Information Technologies, Alfred Nobel University.