МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЧОРНОМОРСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ПЕТРА МОГИЛИ

Іванова Катерина Андріївна

УДК 004.8

АВТОМАТИЗОВАНА СИСТЕМА ВІЗУАЛЬНОГО ТЕСТУВАННЯ WEB-ІНТЕРФЕЙСІВ НА ОСНОВІ МЕТОДІВ ШТУЧНОГО ІНТЕЛЕКТУ

122 – Комп’ютерні науки

Автореферат
магістерської наукової роботи на здобуття освітньої кваліфікації
«Магістр комп’ютерних наук»

Миколаїв – 2020

Магістерська наукова робота є рукопис.

Робота виконана в Чорноморському національному університеті імені Петра Могили Міністерства освіти і науки України на кафедрі інтелектуальних інформаційних систем

Науковий керівник: к.т.н., доцент, доцент кафедри інтелектуальних інформаційних систем Кондратенко Галина Володимирівна

Рецензент: к.т.н., доцент, доцент кафедри інтелектуальних інформаційних систем Калініна Ірина Олександрівна

Захист відбудеться «25» лютого 2020 р. о 9^{30} год. на засіданні екзаменаційної комісії (ауд. 2-403) у Чорноморському національному університеті імені Петра Могили за адресою: 54003, м. Миколаїв, вул. 68-ми Десантників, 10.

З магістерською науковою роботою можна ознайомитися в бібліотеці Чорноморського національного університету імені Петра Могили за адресою: 54003, м. Миколаїв, вул. 68-ми Десантників, 10.

Автореферат представлений «___» лютого 2020 р.

Секретар
екзаменаційної комісії,

к.пед.н., доцент Н. М. Болюбаш

ЗАГАЛЬНА ХАРАКТЕРИСТИКА РОБОТИ

Актуальність дослідження визначається складністю формування оцінки та вибору стартапу через неповноту та нечіткість поданої інформації про можливі альтернативи, складністю виділення найбільш важливих та впливових критеріїв для подальшого прийняття рішень.

Метою наукової роботи є дослідження засобів та технологій штучного інтелекту для аналізу, обробки та порівняння зображень графічного інтерфейсу, захоплених у процесі виконання автоматизованих регресійних функціональних тестів, та створення системи яка дозволить відслідковувати помилки інтерфейсу web-застосунку, генерувати звіт та редагувати базові (очікувані) зображення GUI.

Об'єктом дослідження є процес автоматизації візуальне тестування web-інтерфейсів.

Предметом дослідження є методи аналізу та порівняння очікуваних зображень інтерфейсу та фактичних.

Практичне значення даної магістерської наукової роботи полягає у можливості застосування методів штучного інтелекту для аналізу та порівняння зображень web-інтерфейсу.

Результати даної магістерської наукової роботи було надруковано у тезах XXI Всеукраїнської науково-методичної конференції «Могилянські читання – 2019» у секції Комп'ютерні науки.

Апробація результатів магістерської наукової роботи відбулася під час: «Всеукраїнська науково-практична конференція молодих вчених, аспірантів і студентів».

Магістерська наукова робота складається із вступу, 6 розділів, висновків, додатків. Загальна кількість сторінок – 102, таблиць - 4, рисунків - 27, додатків - 4 та використаних джерел - 61.

ОСНОВНИЙ ЗМІСТ РОБОТИ

У вступі магістерської наукової роботи обґрунтовано актуальність обраної теми, сформульовано мету і задачі дослідження, визначено предмет та об’єкт дослідження.

У першому розділі наведено огляд предметної області та теоретичних засад поняття автоматизованого тестування та саме візуального тестування в сфері інформаційних технологій. Аналіз існуючих методів і підходів показав, що перспективним напрямом у вирішенні задачі автоматизації візуального тестування є порівняння та аналіз зображень методами штучного інтелекту. З урахуванням проведеного аналізу сформовано постановку задачі.

У другому розділі здійснено опис поняття охарактеризовано два існуючі методи візуального тестуавання: метод верифікації стилів елементів, метод по-піксельного порівняння знімків екрану. Також поведено аналіз та наведено недоліки даних методів.

Наразі існує два основних напрямки у автоматизованому візуальному тестуванні, це верифікація назв CSS стилів елементів та порівняння знімків екрану зроблених під час виконання автоматизованих функціональних тестів.

Метод верифікації стилів елементів. Одним із традиційних методів автоматизованого візуального тестування є верифікація CSS стилів відповідного елементу HTML сторінки. Такі верифікації виконуються під час виконання функціональних тестів створених на основі різних фреймворків, таких як Selenium Webdriver, Cypress, WebdriverIO, або Appium. Драйвер управляє web-застосунком так, як це зробив би користувач, і перевіряє, чи програма працює належним чином: якщо з'являється очікуване повідомлення, елемент видаляється або клас CSS додається після відповідної дії користувача.

Для елементу наведеного на рисунку 2.1 можна виконати такі перевірки:

1. Елемент .todo-list повинен мати певний колір представлений у шістнадцятирічному виді #d6d6d6.

2. Елемент .todo-list повинен мати оформлення тексту закресленим – line-through.

Приклад таких перевірок з використанням мови програмування JavaScript та тестового фреймворку Cypress наведено нижче:

```
cy.get('.completed').should('have.css', 'text-decoration', 'line-through')
cy.get('.completed').should('have.css', 'color', 'rgb(217,217,217)')
```

При використанні традиційних контрольних точок в інструментах функціонального тестування, такому як Selenium Webdriver, Cypress, WebdriverIO або Appium, то необхідно буде перевірити наступне для кожного з візуальних елементів:

– видимий (true / false);
– зліва вгорі координати x, y;
– висота;
– ширина;
– колір фону.

Це означає, що знадобиться наступна кількість тверджень:

20 візуальний елемент x 5 тверджень на елемент = 105 рядків коду затвердження

Навіть з усім цим кодом твердження не можливо буде виявити всі візуальні помилки. Наприклад, не можна отримати доступ до візуального елементу, якщо він що він прихований під іншим.

Таким чином, такий метод перевірки має певні недоліки:

– значне збільшення коду тестових методів;
– складність підтримки тестового коду, оскільки зміна назв стилів може спричинити велику кількість невдалих тестів;
– такий спосіб не виявить дійсний візуальний дефект, такий як зміщення елементу на сторінці.

Метод по-піксельного порівняння знімків екрану. Даний метод реалізовується наступним чином. Автоматизоване візуальне тестування першого покоління використовує технологію під назвою snapshot testing. При тестуванні знімків

растрове зображення екрану захоплюється в різних точках тестового прогону, і його пікселі порівнюються з базовим растровим зображенням.

Алгоритми тестування знімків дуже прості: відбувається перебір пари пікселів, а потім перевіряється, чи збігається колірний шістнадцятирічний код. Якщо кольорові коди відрізняються, створюється звіт про візуальну помилку.

Оскільки вони можуть бути побудовані відносно легко, існує ряд відкритих і комерційних інструментів тестування моментальних знімків. На відміну від мануальних тестерів, інструменти тестування знімків можуть швидко і послідовно визначати відмінності пікселів. І це крок вперед, адже комп'ютер може виділити візуальні відмінності досить легко и точно. Деякі з цих інструментів позиціонують себе як так зване "піксель ідеальне тестування".

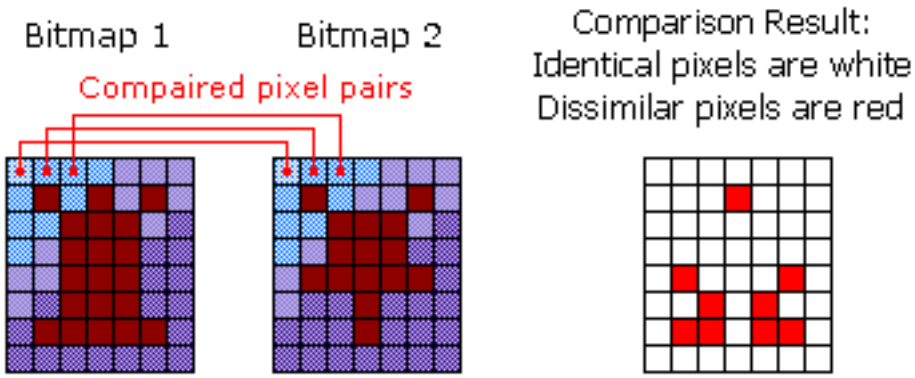

Рис. 2.1. Процес піксельного порівняння зображень

Найбільша проблема з автоматизованим візуальним тестуванням полягає в тому, що люди і машини сприймають пікселі по-різному. Два скріншота користувальницького інтерфейсу можуть здатися цілком ідентичними людині, але на відрізняютися від простого алгоритму 1:1. Алгоритми згладжування шрифтів, зміни розмірів зображень, відеокарти і навіть алгоритми рендеринга створюють відмінності в пікселях. І це просто статичний вміст. Фактичний зміст може варіюватися між будь-якими двома інтерфейсами. У результаті інструмент, який очікує точного збігу пікселів між двома зображеннями, може бути заповнений відмінностями пікселів, тобто великою кількістю помилкових дефектів.

Інша проблема полягає в тому, що ці тести часто повільні порівняно з більш легкими модульними тестами, які не вимагають повного браузера.

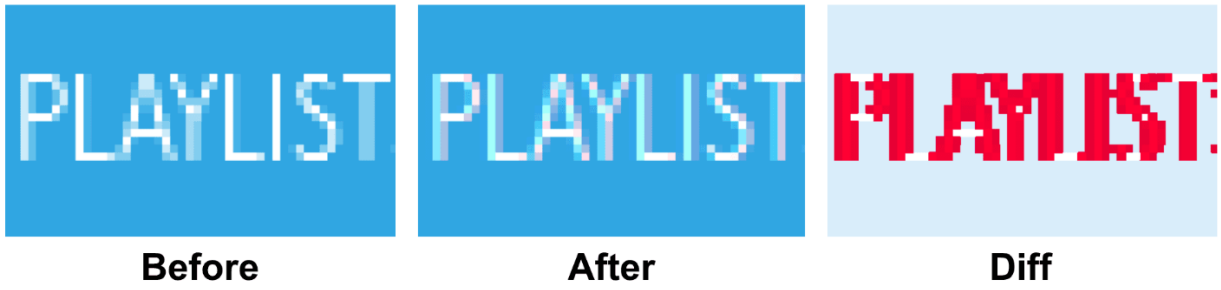

Рис. 2.2. Приклад помилкового спрацювання при порівнянні скріншотів

Підсумовуючи, порівняння зображень очікуваного результату (ОР) та фактичного результату (ФР) у такий спосіб має наступні недоліки:

– неможливість пошуку візуальних змін при різній роздільності зображень;

– значне сповільнення часу виконання автоматизованих тестових наборів;

– неможливість пошуку візуальних змін окремо для певного компоненту графічного інтерфейсу;

– висока ймовірність помилкового спрацювання.

В третьому розділі запропоновано використовувати алгоритми сегментації зображення та оглянуто деякі з них. Було розглянути згорткові нейронні мережі, як метод сегментації зобрадень та їх більш складні варіації які відносяться до технології Deep Learning – це U-Net та FCN. Також було оглянуто алгоритми кластеризації K-means та Mean Shift для задачі сегментації зображення.

Процес сегментації зображень відбувається по інформації наявній в самому зображенні. Це може бути інформації про колір пікселя, перепади кольорів та відтінків чи текстура зображення.

Техніки сегментації можна поділити на дві групи:

– методи на основі подібності;

– методи на основі границь (порогів).

Кожний підхід має різні варіанти реалізації. Алгоритм k – середніх

Алгоритм k-середніх (англ. k-means) був запропонований ще в 1979 році, проте він не втрачає своєї актуальності і сьогодні. Ідея, що лежить в його основі,

досить проста, і алгоритм працює досить ефективно. Однак, оптимальність рішень, отриманих за допомогою Алгоритму k-середніх, не гарантована. Крім того, недоліком алгоритму є необхідність знати число кластерів заздалегідь.

Формально рішення задачі кластеризації полягає в тому, щоб «помітити» кожен з наявних об'єктів – приписати йому номер певного класу. Алгоритм k-середніх передбачає таке розбиття об'єктів на класи, при якому мінімізуються відмінності («відстані») між об'єктами одного і того ж класу та максимізуються рас стояння між об'єктами різних класів.

Даний алгоритм являється прообразом практично всіх алгоритмів нечіткої кластеризації, і його розгляд допоможе нам краще зрозуміти принципи, що закладені в більш складні алгоритми.

В загальному алгоритм представляє собою ітераційну процедуру:

– Крок 1. Проініціалізувати початкову матрицю розбиття U випадковим чином і обрати точність δ, що буде використовуватися для завершення алгоритму, встановити номер ітерації $l = 0$;

– Крок 2. Визначити центри кластерів за наступною формулою:

$$c_l^{(i)} = \frac{\sum_{j=1}^{d} u_{ij} m_j}{\sum_{j=1}^{d} u_{ij}}, 1 \le i \le c \qquad (1.1)$$

де $c_l^{(i)}$– центри кластерів, d – вимір об’єкта, що досліджується, c – кількість кластерів, l – номер поточної ітерації, u – матриця розбиття, m – об’єкт, що досліджується;

– Крок 3. Обновити матрицю розбиття:

$$u_{ij}^{(l)} = \begin{cases} 1, & \text{при } d(m_j, c_i) = \min\limits_{l \le k \le c} d(m_j, c_k) \\ 0, & \text{в інших виппадках} \end{cases} \qquad (1.2)$$

де $u_{ij}^{(l)}$– елемент матриці розбиття, $d(x,y)$ – обрана метрика, c –кластер, m – об’єкт, що досліджується;

– Крок 4. Перевірити умову $\left\| U^{(l)} - U^{(l-1)} \right\| < \delta$, де U –матриця розбиття, і δ – обрана точність. Якщо умова виконується – завершити процес, якщо ні – перейти до кроку 2 з номером ітерації $l = l + 1$.

Також існує альтернативний варіант даного алгоритму:

– Крок 1. Випадковим чином обрати центри кластерів із елементів вхідних даних і обрати точність δ, що буде використовуватися для завершення алгоритму, встановити номер ітерації $l = 0$;

– Крок 2. Оновити матрицю розбиття(формула 1.1);

– Крок 3. Визначити центри кластерів(формула 1.2);

– Крок 4. Перевірити наскільки змістилися центри кластерів. Якщо вони змістилися менше ніж на точність δ – завершити процес, якщо ні – перейти до кроку 2 з номером ітерації $l = l + 1$.

Також існує набір обмежень: $u_{ij}^{(l)} \{0,1\}; \sum_{j=1}^{c} u_{ij} = 1; 0 < \sum_{j=1}^{d} u_{ij} < d$, який визначає, що кожен вектор даних може належати тільки одному кластеру і не належати іншим. В кожному кластері повинно бути не менше одного елемента даних і не більше загальної кількості елементів.

Основний недолік даного алгоритму через дискретність елементів матриці розбиття – великий розмір просторового розбиття. Одним із способів усунення даного недоліку є представлення елементів матриці розбиття числами із інтервалу від 0 до 1. Тобто належність елемента даних до кластера визначається функцією належності – елемент даних може належати декільком кластерам з різною степеню належності.

Метод середнього зсуву (Mean Shift)

На відміну від алгоритму K-Means, метод Mean Shift не вимагає вказівки кількості кластерів заздалегідь. Кількість кластерів визначається алгоритмом за вихідними даними. Напрямок до найближчого кластерного центроїду визначається тим, де знаходиться велика частина найближчих точок.

Mean Shift – це потужний і універсальний непараметричний ітеративний метод. Середнє зміщення було введено в Fukunaga та Hostetler і було розширено для застосування в інших областях, таких як Computer Vision. Метод розглядає функціональний простір як емпіричну функцію густини імовірності. Якщо вхід являє собою набір точок, то mean shift розглядає їх як вибірку з базової функції густини імовірності. Якщо у функціональному просторі присутні густі регіони (або

кластери), то вони відповідають формі (або локальному максимуму) функції щільності ймовірності. Ми також можемо визначити кластери, пов'язані з даною формою, використовуючи метод mean shift.

Для кожної точки даних середній зсув асоціює його з найближчою вершиною функції густини імовірності набору даних. Для кожної точки даних середній зсув визначає вікно навколо нього та обчислює середнє значення точок даних. Потім він зміщує центр вікна на середнє і повторює алгоритм, поки не зійдеться. Після кожної ітерації ми можемо вважати, що вікно переходить у більш густу область набору даних.

Mean Shift групує об'єкти з близькими ознаками. Пікселі зі схожими ознаками об'єднуються в один сегмент, на виході отримуємо зображення з однорідними областями.

Наприклад, в якості координат в просторі ознак можна вибрати координати пікселя (x, y) і компоненти rgb пікселя. Зобразивши пікселі в просторі ознак, можна помітити згущення в певних місцях.

Щоб легше було описувати згущення точок, вводиться функція щільності:

$$f(\vec{x}) = \frac{1}{Nh^d} \sum_{i=1}^{N} K\left(\frac{\vec{x} - \vec{x_i}}{h}\right) \tag{1.5}$$

де $\vec{x}$ – вектор ознак і-ого пікселя,

d - кількість ознак,

N - число пікселів,

h - параметр, що відповідає за гладкість.

Максимуми функції розташовані в точках згущення пікселів зображення в просторі ознак. Пікселі, що належать одному локальному максимуму, об'єднуються в один сегмент. Виходить, щоб знайти до якого з центрів згущення відноситься піксель, треба крокувати по градієнту для знаходження найближчого локального максимуму.

При виборі в якості ознак координат пікселів і інтенсивностей за кольорами в один сегмент будуть об'єднуватися пікселі з близькими квітами і розташовані недалеко один від одного. Відповідно, якщо вибрати інший вектор ознак, то

об'єднання пікселів в сегменти вже буде йти по ньому. Наприклад, якщо прибрати з ознак координати, то небо і озеро будуть вважатися одним сегментом, так як пікселі цих об'єктів в просторі ознак потрапили б в один локальний максимум.

У четвертому розділі наводиться опис розробленого програмно-алгоритмічного забезпечення.

У спеціальній частині магістерської наукової роботи з «Охорони праці та безпеки життєдіяльності» розглянуто умови праці співробітників компанії «Global Logic Україна», заходи щодо їх покращення, а також заходи з метою запобігання та ліквідації наслідків надзвичайної ситуації, пов’язаної с пожежею. Розглянуті вимоги щодо забезпечення пожежної безпеки приміщень з робочими місцями, обладнаними комп’ютерною технікою. Наведені причини виникнення надзвичайної ситуації пов’язаною з пожежею, засоби запобігання її виникнення та заходи, щодо ліквідації наслідків надзвичайної ситуації.

У методичній частині розроблено практичні роботи на теми «Метод K-means» та «Метод Mean Shift».

ЗАГАЛЬНІ ВИСНОВКИ

Графічний інтерфейс являє собою центральний вузол в тестованому застосунку, звідки здійснюється доступ до всіх функцій. Важко ретельно протестувати програми через їх графічний інтерфейс, особливо тому, що графічні інтерфейси призначені для роботи з людьми, а не машинами. Не дивлячись на те, що багато автоматизованих засобів і методів тестування було розроблено, вони все ще не вирішують всі проблеми, такі як відслідковування візуальних змін інтерфейсу.

У зв’язку з стрімким розвитком автоматизованого візуального тестування, з’явилась необхідність розробки системи, яка дозволить виконувати більш точний аналіз інтерфейсу та зменшити кількість помилкових спрацювань.

Для досягнення поставленої мети було виконано наступні завдання:

- проаналізовано сучасний стан задачі автоматизації візуального тестування;
- проведено та вибір методів для аналізу зображень;
- реалізовано обраних методів для вирішення поставленої задачі.

Результатом виконання даної дипломної роботи є розроблена автоматизована система для візуального тестування web-інтерфейсів з використанням методів штучного інтелекту для сегментації зображень та більш точного аналізу.

Поставлену мету дослідження засобів та технологій штучного інтелекту для аналізу, обробки та порівняння зображень графічного інтерфейсу, захоплених у процесі виконання автоматизованих регресійних функціональних тестів було досягнуто.

АНОТАЦІЯ

Іванова Катерина Андріївна. Автоматизована система візуального тестування web-інтерфейсів на основі методів штучного інтелекту. – На правах рукопису.

Магістерська наукова робота на здобуття освітньої кваліфікації «Магістр комп'ютерних наук». – Чорноморський національний університет імені Петра Могили, Миколаїв, 2020.

Дана магістерська наукова робота присвячена питанню здійснення автоматизованого візуального тестування за допомогою методів штучного інтелекту для аналізу та порівняння зображень графічного інтерфейсу.

Метою наукової роботи є дослідження засобів та технологій штучного інтелекту для аналізу, обробки та порівняння зображень графічного інтерфейсу, захоплених у процесі виконання автоматизованих регресійних функціональних тестів, та створення системи яка дозволить відслідковувати помилки інтерфейсу web-застосунку, генерувати звіт та редагувати базові (очікувані) зображення GUI.

Об'єктом дослідження є автоматизоване візуальне тестування web-інтерфейсів.

Предметом дослідження є методи аналізу та порівняння очікуваних зображень інтерфейсу та фактичних.

Основна частина складається з наступних розділів: аналіз та дослідження сучасного стану автоматизованого візуального тестування; аналіз методів, моделей та технологій для вирішення задачі візуального тестування; порівняння та вибір методів штучного інтелекту; розробка програмного забезпечення для аналізу та порівняння зображень графічного інтерфейсу.

У методичній частині розроблено практичні роботи на теми «Метод K-means» та «Метод Mean Shift»

В спеціальній частині дипломної роботи з «Охорони праці» розглянуто умови праці на робочих місцях у відділі розробки програмного забезпечення ТОВ «Глобал Лоджик». Результатом даного дослідження є інтегральна оцінка стану умов праці в приміщенні, а також рекомендації щодо їх покращення.

В цілому дипломна робота складається з вступу, шести розділів, висновків, розділу з ООП та методичної частини, загальна кількість сторінок – 102, таблиць - 4, рисунків - 27, додатків - 4 та використаних джерел - 61.

Ключові слова: *автоматизоване тестування, візуальне тестування, штучний інтелект, сегментація зображення, графічний інтерфейс.*

ABSTRACT

Kateryna Ivanova. AI based automated visual testing of web interfaces – On the rights of the manuscript.

This master's research paper is devoted to the implementation of automated visual testing using methods of artificial intelligence for analyzing and comparing images of the graphical user interface.

The purpose of the research is to study artificial intelligence tools and technologies for analyzing, processing and comparing graphical interface images captured during automated regression functional tests, and to create a system that will allow users to track errors in the web application interface, generate a report, and edit the basic (expected) GUI image.

The object of the research is automated visual testing of web interfaces.

The subject of the research is the methods of analysis and comparison of expected interface images and actual ones.

The main part consists of the following sections: analysis and research of the current state of automated visual testing; analysis of methods, models and technologies for solving the problem of visual testing; comparison and selection of artificial intelligence methods; development of software for analyzing and comparing images of the graphical interface.

In the special part of the diploma work regarding the "Occupational safety and health" the evaluating of working conditions at the workplaces in the department of software development at "Global Logic" Ltd. are considered. The result of this study is an integrated assessment of the condition of working conditions in the premises, as well as recommendations for their improvement.

In the methodical part practical works on the course of the topics "K-means Method" and "Mean Shift Method" are developed.

The work consists of 102 pages, 27 figures, 4 tables and 61 references to literary sources.

Keywords: *automated testing, visual testing, artificial intelligence, image segmentation, graphical interface.*