УДК 004.93′12; 004.932; 004.932.75
DOI: 10.15587/1729-4061.2018.148307

Розробка системи розпізнавання графічних Captcha-систем за допомогою конкуруючих клітинних автоматів

І. В. Миронів, В. В. Жебка, С. Е. Остапов, О. Д. Валь

Розглянуто особливості застосування конкуруючих клітинних автоматів до задач з розпізнавання складних captcha-систем. Для цього введено поняття конкуруючих клітинних автоматів та розроблено алгоритми їх функціонування та взаємодії. Описано математичну модель конкуруючих клітинних автоматів на основі теорії множин для опису рухомих клітинних автоматів, що пересуваються по сусідніх станах символів і в такий спосіб реалізують свої правила переходу.На основі математичної моделі розроблено систему розпізнавання captha-зображень, яка реалізована в програмному коді засобами технології JavaFX 2.0, що дозволила досягти кросплатформеності та правильного функціонування на різних операційних системах.

Бібліотеки клітинних автоматів розроблялися для англійської мови. Кожен символ алфавіту представляється у вигляді системи станів, котрій поставлено у відповідність клітинний автомат зі станами, що описують даний символ.

Для розробки було використано мову програмування Java та можливості обробки зображень бібліотекою OpenCV, яка дозволила досягти якісного результату розпізнавання. Розглянуто архітектуру розробленої системи розпізнавання складних captcha-зображень у вигляді діаграм класів головних блоків з детальним описом кожного класу. Проведено комп'ютерні експерименти з різними наборами спотворених символів, що використовуються у реальних captcha-системах та отримано показники якості розпізнавання розробленим програмним забезпеченням.

Показано, що ймовірність отримання правильного результату розпізнавання captcha-зображень перевищує 80 % при ступені деформації символів до 20 %. При ступені деформації символів понад 30 % існує велика ймовірність помилкового розпізнавання символів.

До перевагам методу розпізнавання символів тексту на основі конкуруючих клітинних автоматів слід віднести простоту правил взаємодії, можливість легкого розпаралелювання процесу розпізнавання, можливість розпізнавання спотворених та частково накладених символів, які складають основу сучасних captcha-систем

Ключові слова: конкуруючий клітинний автомат, рухомий клітинний автомат, captcha-системи

1. Вступ

Captcha найчастіше використовується при необхідності запобігти використовування інтернет-сервісів ботами, зокрема для запобігання автоматичних ре-

єстрацій поштових скриньок, відправок повідомлень, скачування файлів, масових розсилок тощо [1].

Актуальність застосування Captcha можна побачити, наприклад, зі статистики обсягів спаму, що розсилається. За оцінками глобальних поштових сервісів, обсяг спаму досягає 97 % від загальної кількості електронних повідомлень. Застосування Captcha-захисту дозволяє ускладнити задачу реєстрації поштових скриньок ботами і таким чином, зменшити об'єми спам-розсилок.

2. Аналіз літературних даних та постановка проблеми

Дуже привабливим з точки зору розробки нових систем розпізнавання є використання в них клітинних автоматів. У роботі [2] автором було показано безперечні переваги використання клітинних автоматів (КА) у задачах, де виникає необхідність розпаралелювання обчислень, що надає можливість простої реалізації складних алгоритмів обробки зображень та не вимагає значних обчислювальних ресурсів. Незважаючи на вказані переваги, клітинні автомати не так часто залучалися до задач розпізнавання. Чи не єдиними ретельними дослідженнями у цій області є робота [3], основна частина якої присвячена вивченню характеристик КА в процесах розпізнавання тексту. Автор використовує послідовності різних КА для виділення характерних ознак символів тексту: петель, перетинів, положень кінців. В роботі [4] досліджено КА, що дозволяють розпізнавати рукописні символи. Основними недоліками таких підходів вважається громіздкість та необхідність навчання системи.

Крім того, в роботі [5] автором було запропоновано новий алгоритм розпізнавання JPEG зображення водяних знаків на основі клітинних автоматів. В роботі [6] побудовані спеціалізовані клітинно-автоматні структури для аналізу зображення контуру [5, 6].

У роботах [7, 8] авторами було запропоновано підхід по сегментації зв'язаних символів у текстових CAPTCHA, та отримано результат розпізнавання символів, які не розділені між собою. А в роботі [9] авторами було проведено оцінку останніх досліджень по розпізнаванні Captcha-систем.

Найбільш відомий сервіс для онлайн розпізнавання [10] Captcha-систем, який представляє собою плагін під популярні браузери Chrome і Firefox, успішно розпізнає символи, структура яких не змінена деформацією, а символи, які накладаються, взагалі ігноруються. Середній час розпізнавання однієї капчи – 8 секунд.

Однак існує спосіб реального використання нового, запропонованого авторами [11], типу КА в процесі розпізнавання символів. Цей підхід ґрунтується на рухомих КА, які повинні реалізувати усі свої стани на відповідному символі тексту. Неоднозначність інтерпретації символів, яка виникає при цьому, компенсується розробленим механізмом конкуренції, коли "перемагає" КА з максимальним числом реалізованих станів. Такий КА і є найбільш правильним відображенням символу, що розпізнається.

Водночас усі розглянуті методи та системи розпізнавання неефективно працюють з рукописними та частково деформованими символами, а також сим-

волами, що частково накладаються, які складають основу сучасних Captcha – систем, як наведено на рис. 1.

Рис. 1. CAPTCHA-системи, які використовуються у Google

На рис. 1 наведено варіант Captcha-систем, що використовуються таким Інтернет гігантом як Google. Дані Captcha-системи характеризуються нелінійними спотвореннями тексту, зміщення символів один щодо одного, близьким розташування символів, різними шрифтами. Шуми не застосовуються, однак символи не завжди склеюються без проміжків, що ускладнює сам процес розпізнавання.

3. Мета та задачі дослідження

Мета роботи полягає у розробці на основі конкуруючих клітинних автоматів системи розпізнавання деформованих та таких, що частково накладаються, символів, аналогічних Captcha-системам Google, які не в змозі розпізнавати існуючі системи.

Для досягнення поставленої мети необхідно виконати такі завдання:

– розробити математичну модель рухомих клітинних автоматів, придатну для застосування в задачах розпізнавання;

– розробити механізм конкуренції клітинних автоматів для підвищення ефективності розпізнавання;

– розробити архітектуру та інтерфейс системи розпізнавання на основі конкуруючих КА;

– провести дослідження ефективності розробленого програмного забезпечення.

4. Математична модель рухомих конкуруючих клітинних автоматів

Множина клітинних автоматів, яка використовуються у цій роботі, може бути записана:

$$U = \{\sigma_i\}_{i=1}^{N}, \qquad (1)$$

де N – потужність алфавіту.

Кожен автомат має свою множину станів U, мітку ξ (колір, положення в алфавіті тощо), і залежить від дискретного часу:

$$\sigma_i = \sigma_i(U, t, \xi), \tag{2}$$

$U = \{u_k\}_{k=1}^{K}$; K – кількість станів поточного автомата. Перехід автомата з поточного стану k в наступний k+1 можна описати:

$$\sigma_i(u_k, t, \xi) \overset{t \to t+1}{\rightarrow} \sigma_i(u_{k+1}, t+1, \xi). \tag{3}$$

Переходом КА у новий стан управляє функція переходу φ тоді можна записати:

$$\sigma_i\left(u_k, t, \xi\right) = \varphi(\sigma_i\left(u_k, t, \xi\right)). \tag{4}$$

Рухомий КА рухається з поточного стану у наступний, використовуючи правила, що формуються функцією переходів.

Припустимо, що було подано зображення символів, які необхідно розпізнати, у вигляді сукупності станів, аналогічних станам КА. Тоді $\Omega = \{\omega_p\}_{p=1}^{P}$ – множина цих станів. Кількість станів P символу невідома.

Потрапивши на символ, автомат буде рухатися по його станах ω_p, кількість яких, P, взагалі кажучи, не дорівнює кількості станів поточного автомата K, $P \neq K$.

Переходи будуть виконуватися КА по станах символу, тобто:

$$\sigma_i(\omega_p, t, \xi) \overset{t \to t+1}{\rightarrow} \sigma_i(\omega_{p+1}, t+1, \xi), \tag{5}$$

тоді і тільки тоді, коли $\omega_p = u_k \in U$ та $\omega_{p+1} = u_{k+1} \in U$, тобто коли стани символу співпадають з аналогічними станами КА. Якщо ж $\omega_p, \omega_{p+1} \neq u_k, u_{k+1} \in U$, такий КА не має дозволених переходів і вилучається з клітинно -автоматного поля.

Якщо КА може реалізувати усі свої стани, задані функцією переходів на поточному символі, тобто якщо $\forall u_k \in U \exists \omega_p \in \Omega$, будемо вважати, що цей КА «вдало» описує поточний символ.

Якщо ж $\forall u_k \in U \nexists \omega_p \in \Omega$, тобто хоча б для одного стану КА не існує аналогічного стану символу, такий автомат вилучається з КА-поля.

КА, що реалізують усі свої можливі стани на поточному символі, може бути декілька, Для того, щоб обрати з них той, який точно відповідає цьому символу, використовується механізм конкуренції, який полягає у наступному.

Нехай на одному символі рухається 3 КА:

$$\sigma_i \rightarrow U = \{u_k\}_{k=1}^{K},$$

$$\sigma_j \rightarrow V = \{v_l\}_{l=1}^{L}, \tag{6}$$

$$\sigma_h \to W = \{w_s\}_{s=1}^{S}.$$

Кожен з них реалізує на ньому усі свої стани. Будемо вважати, що конкуренцію «виграє» той КА, кількість станів якого найбільша, тобто необхідно знайти

$$\max(K, L, S). \qquad (7)$$

Нехай $\max(K, L, S)=S$. Тоді КА $\sigma_h = \sigma_h(S, t, \eta)$ буде вважатися таким, що найбільш «вдало» описує поточний символ. Зчитуючи його мітку η, дізнаємося, який символ розпізнали.

Сам алгоритм розпізнавання ґрунтується на побудові КА та його графа переходів, по станах якого рухається даний автомат.

5. Архітектура системи розпізнавання

Запропонована система розпізнавання була реалізована як програмний продукт засобами технології JavaFX 2.0. Бібліотеки клітинних автоматів розроблялися для англійської мови. Для розробки було використано мову програмування Java та можливості обробки зображень бібліотекою OpenCV [12], яка дозволила досягти хорошого результату.

Діаграма основних класів [13], яка відповідає за роботу з камерою пристрою, наведена на рис. 2.

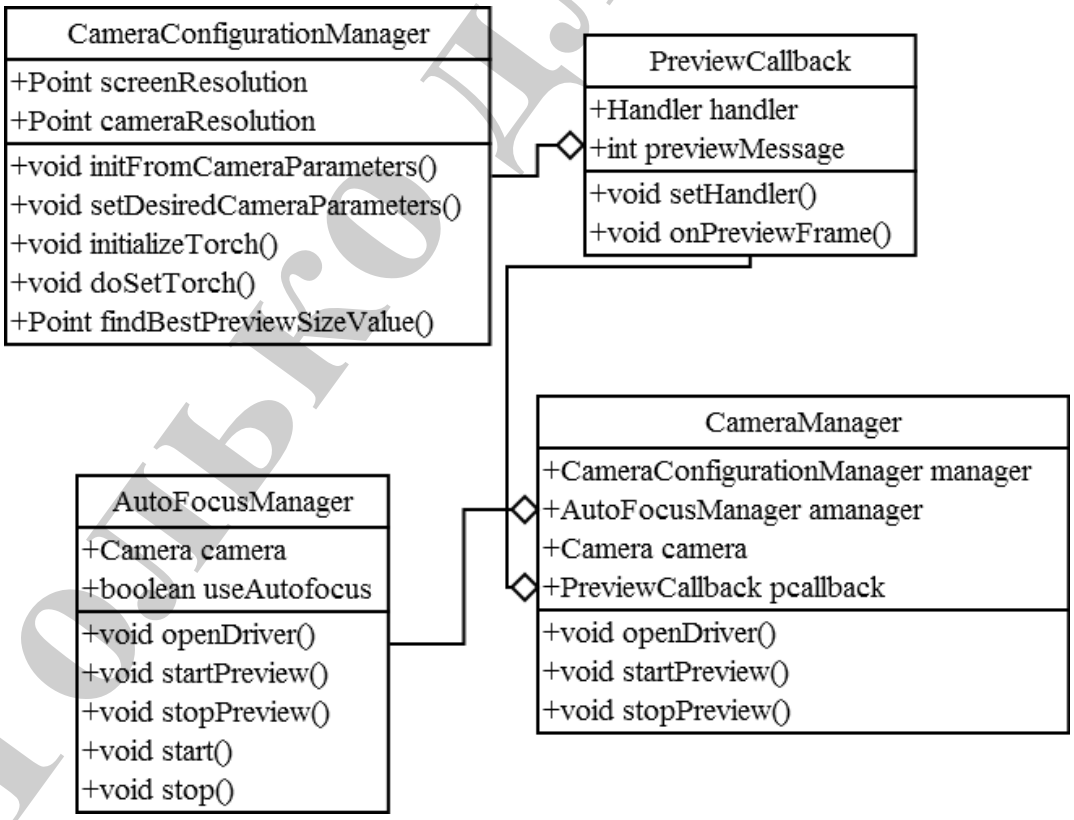

Рис. 2. Діаграма основних класів для роботи з камерою

Опис цих базових класів блоку роботи з камерою наведено в табл. 1.

Таблиця 1
Опис класів блоку роботи з камерою

Клас	Опис
CameraManager	Базовий клас, що описує функціонування камери у режимі отримання зображення для розпізнавання
AutoFocusManager	Клас, що описує роботу камери в режимі автофокусу
CameraConfiguratonManager	Клас, що містить всю логіку по налаштуванні камери
PreviewCallback	Клас, який відповідає за отримання зображення

Блок роботи з камерою складається з одного основного класу: CameraManager, який описує роботу камери у режимі отримання зображення для розпізнавання, та трьох допоміжних класів, які описують логіку налаштування та функціонування камери.

Діаграма основних класів, яка відповідає за попередню обробку зображення, отриманого з камери, або зі скануючого пристрою, подана на рис. 3.

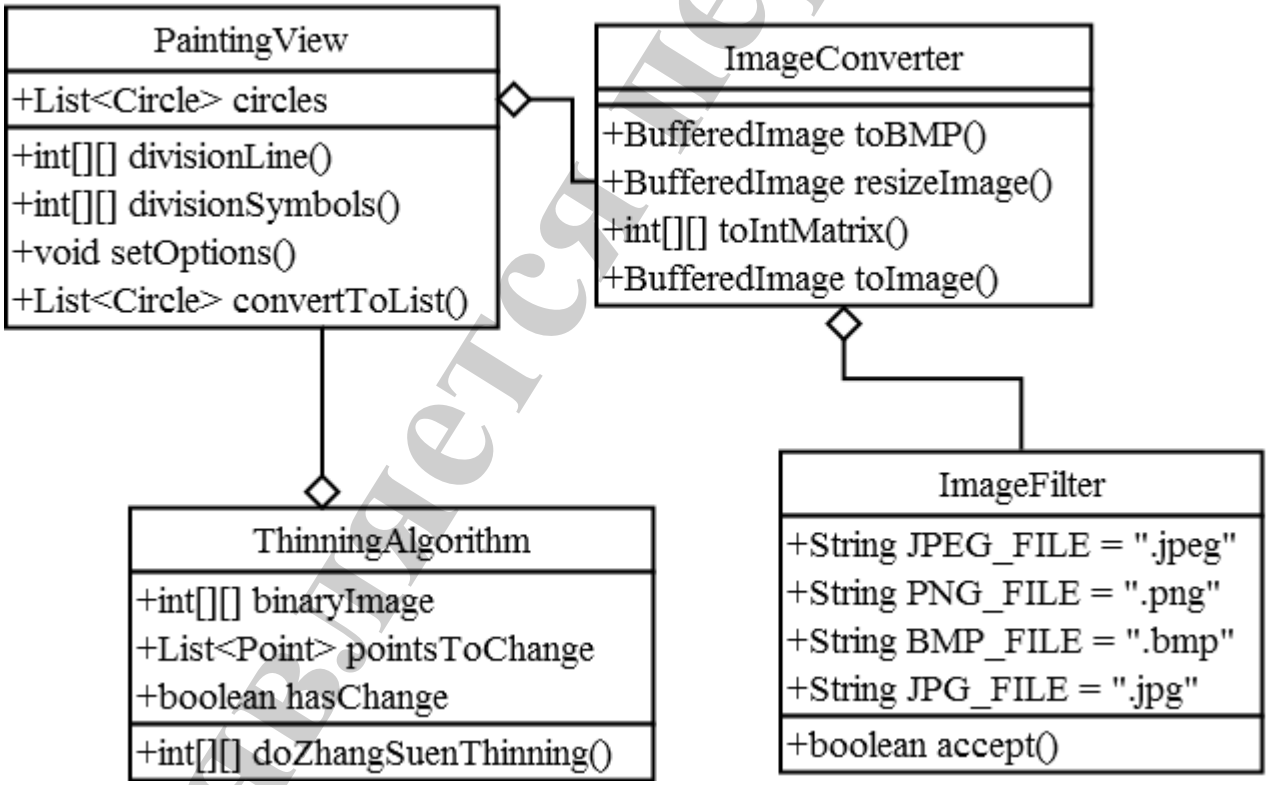

Рис. 3. Діаграма основних класів для роботи із зображенням

Опис базових класів блоку попередньої обробки зображення подано в табл. 2.

Таблиця 2

Опис класів блоку попередньої обробки зображення

Клас	Опис
ThinningAlgorithm	Базовий клас, що описує роботу алгоритмів обробки зображення
ImageFilter	Клас, що описує вибір відповідного алгоритму обробки зображення у відповідності до його формату
ImageConverter	Клас, що містить логіку конвертації зображення у відповідну модель матриці пікселів
PaintingView	Клас який відповідає поділ зображення на рядки та символи

Блок попередньої обробки зображення складається з одного основного класу: ThinningAlgorithm, який описує основні алгоритми для обробки зображення. Три допоміжних класи, які відповідають за вибір відповідного алгоритму обробки зображення згідно його формату, за отримання матриці пікселів вхідного зображення та розбиття на рядки та символи.

Спрощена діаграма класів розробленого програмного забезпечення подана на рис. 4. На ньому відображено лише взаємодію класів, які описують клітинні автомати, тобто лише те, що важливе саме для процесу розпізнавання.

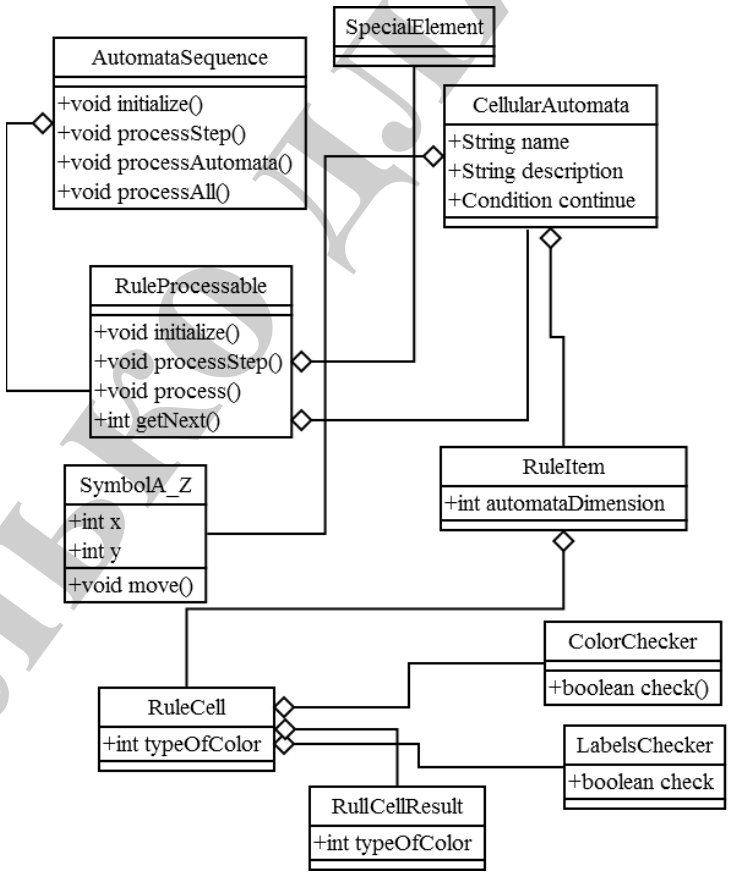

Рис. 4. Частина діаграми класів розробленого програмного забезпечення, яка реалізує роботу з клітинними автоматами

У табл. 3 подано опис класів блоку клітинних автоматів.

Таблиця 3
Опис класів блоку клітинних автоматів

Клас	Опис
CellularAutomata	Базовий клас, що описує роботу КА
RuleItem	Клас, що описує правила переходів КА
RuleCell	Клас, що описує умову, накладену на клітинку в рамках правила КА
ColorChecker	Абстрактний клас, що перевіряє колір клітинки
LabelsChecker	Абстрактний клас, що перевіряє мітки клітинки у графі переходів
RullCellResult	Клас, що описує події по визначенню кольору і міток клітинки в рамках правила КА
AutomataSequence	Базовий клас, що описує роботу послідовності КА та перевірку його конкуренції
RuleProcesable	Абстрактний клас, що описує роботу елемента послідовності
SpecialElement	Абстрактний клас, що описує роботу спеціального елемента послідовності
SymbolA_Z	Клас, що описує символи латинського алфавіту за допомогою КА

Система складається з двох базових класів: CellularAutomata та AutomataSequence, які описують роботу КА та їх послідовностей для запуску механізму конкуренції. У клітинно-автоматну взаємодію включені такі елементи як поділ зображення на окремі символи, перевірка умов у графі переходу та ін. Стани кожного автомату, який відповідає певній літері алфавіту та правила переходів по графу, реалізовано в класах RuleItem, RuleCell та SymbolA_Z. Для спрощення діаграми описи усіх літер показано в одному класі, хоча насправді їх реалізовано окремо. Ці класи відповідають за реалізацію руху КА та описують граф переходів. Індексом типу автомата, який дозволяє його однозначно ідентифікувати, служить кольорова мітка. Зчитуючи цю мітку ми можемо визначити розпізнану літеру. За цей процес відповідають класи RuleCellResult, LabelsChecker, ColorChecker. Результатом цієї роботи буде розпізнаний за допомогою конкуруючих клітинних автоматів текст.

Взаємодія з апаратним забезпеченням та виведення розпізнаного тексту виконується за допомогою стандартних функцій API Windows. Обробка зображень, отриманих зі скануючого обладнання, виконувалася за допомогою бібліотек з відкритим кодом OpenCV.

6. Опис інтерфейсу системи розпізнавання

Створене програмне забезпечення має дуже простий інтерфейс, оскільки призначене для апробації розроблених алгоритмів та методів розпізнавання, а

не для комерційного використання. Програмне забезпечення складається з одного вікна, яке розділене на два блоки. В верхньому блоці завантажується captcha зображення отримане шляхом генерування Captcha-генераторами, або простим збереженням зображення з браузера. У нижньому блоці виводиться отриманий результат.

Головне вікно програми в режимі розпізнавання наведено на рис. 5.

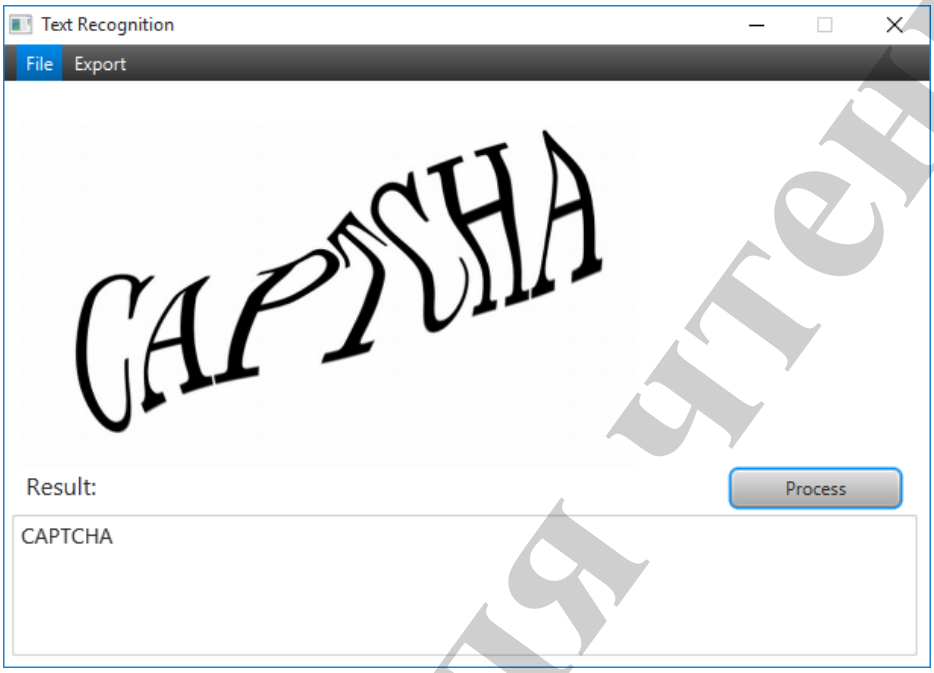

Рис. 5. Інтерфейс програмного забезпечення в режимі розпізнавання Captcha-систем

Система призначена для розпізнавання об’єктів зі зміщенням та близьким розташуванням символів, накреслених різними шрифтами та нечіткі зображення, що складаються з деформованих символів, об’єднаних у декілька груп.

7. Обговорення результатів дослідження розпізнавання captcha-зображень

Якість роботи розробленої системи була оцінена шляхом проведення розпізнавання captcha-зображень на персональному комп’ютері наступної конфігурації:

1. Процесор – Intel(R) Core(TM) i7-3612QM CPU @ 2.10 GHz.
2. ОЗП – 8 Гб.
3. Відеоадаптер – ATI AMD Radeon HD 7600M (1024 Мб).
4. Жорсткий диск – Seagate ST1000LM (931GB, SATA II).
5. DVD-RW – LG DVD+-RW.

Даний комп’ютер працює під ОС Windows 10 Professional.

Для генерування вхідних зображень використовувалися генератори Captcha [14]. Результати досліджень у вигляді графіка залежності якості розпізнавання від ступеня деформації наведено на рис. 6.

Якість розпізнавання – усереднення отриманих даних десяти незалежних проведених експериментів.

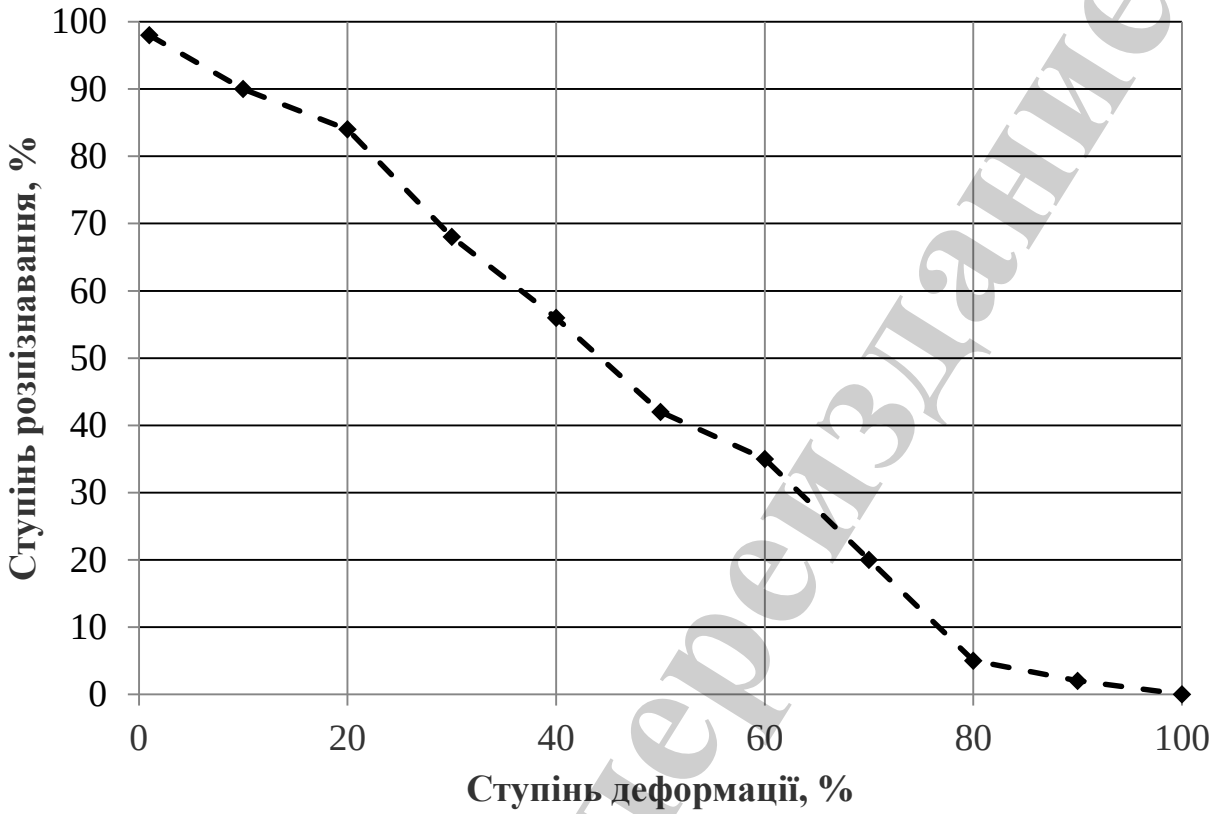

Рис. 6. Характеристика якості розпізнавання символів Captcha залежно від ступеня деформації розробленим програмним забезпеченням

Аналіз рис.6 показує, що залежність ступінь розпізнавання символів CAPTCHA розробленим програмним забезпеченням від ступеня деформації майже лінійна і складається з трьох ділянок. Перша ділянка – від 0 до 20 % деформації, де система розпізнає більше 80 % наданих символів, тобто показує досить високі результати. Друга ділянка, від 20 % до 80 % деформації, де нахил графіку збільшується, демонструє поступове спадання ймовірності розпізнавання, і при 45 % деформації розпізнає всього 50 % символів. Далі спадання продовжується, і система практично припиняє правильно розпізнавати Captcha при 80 % деформації (демонструє усього 5 % правильно розпізнаних символів). При подальшому збільшенні ступеня деформації ймовірність правильного розпізнавання спадає до нуля.

Таким чином, можна стверджувати, що розроблений метод можна успішно (до 70 % ймовірності правильного розпізнавання) використовувати для розпізнавання Captcha, символи яких деформовано не більше як на 30 %. Слід, однак, зауважити, що існуючи системи розпізнавання Captcha взагалі не можуть працювати з деформованими символами: саме тому, зокрема, й було уведено їх деформацію. Той факт, що розроблена в даній роботі система може впоратись з

частково деформованими символами та такими, що накладаються, можна вважати її суттєвою перевагою.

До недоліків слід віднести те, що ступінь впевненого розпізнавання обмежена 30 % деформаціями символів. Це, звичайно, мало. Подальші вдосконалення механізму конкуренції КА повинні збільшити цю ділянку, однак зрозуміло, що успішне розпізнавання повністю деформованих символів (або таких, що мають спільні лінії), сьогодні неможливе без залучення Machine Learning, що не є предметом цієї роботи. В подальшому поєднання теорії КА з засобами Machine Learning може призвести до суттєвого прориву у системах розпізнавання, що працюють в екстремальних умовах низької якості об'єктів розпізнавання.

7. Висновки

1. Запропоновано новий клас рухомих КА, рух яких описується правилами переходу: $\sigma_i(\omega_p, t, \xi) \overset{t \to t+1}{\to} \sigma_i(\omega_{p+1}, t+1, \xi)$, що дає змогу порівнювати траєкторію руху зі станами символу, який описує КА.

2. Розроблено механізм конкуренції КА, який полягає в тому, що автомат з максимальною кількістю реалізованих на конкретному символі станів "виграє" конкуренцію серед усіх, що одночасно рухаються по цьому символу автоматів, і визнається таким, що найправильніше описує поточний символ.

3. Розроблено архітектуру та спрощений одновіконний інтерфейс системи розпізнавання Captcha-символів на основі конкуруючих КА для дослідження адекватності моделі та якості розпізнавання.

4. Показано, що розроблена система демонструє високу ймовірність правильного розпізнавання символів Captcha (до 70 %) при невисоких ступенях деформації (до 30 %). При більших ступенях деформації ймовірність правильного розпізнавання суттєво зменшується.

Література

1. Тьюринг А. М. Вычислительные машины и разум. Самара: Бахрах-М, 2003. 128 с.
2. Wolfram S. A. New Kind of Science. Wolfram Media. Inc., 2002. 1197 p.
3. Oliveira C. C., de Oliveira P. P. B. An Approach to Searching for Two-Dimensional Cellular Automata for Recognition of Handwritten Digits // Lecture Notes in Computer Science. 2008. P. 462–471. doi: https://doi.org/10.1007/978-3-540-88636-5_44
4. Суясов Д. И. Выделение структурных признаков изображений символов на основе клеточных автоматов с метками // Информационно-управляющие системы. 2010. № 4. С. 39–45.
5. A new JPEG Image Watermarking Algorithm Based on Cellular Automata / Wu H., Zhou J., Gong X., Wen Y., Li B. // Journal of Information & Computational Science. 2011. Vol. 8, Issue 12. P. 2431–2439.
6. Belan S. N. Specialized cellular structures for image contour analysis // Cybernetics and Systems Analysis. 2011. Vol. 47, Issue 5. P. 695–704. doi: https://doi.org/10.1007/s10559-011-9349-8

7. Hussain R., Gao H., Shaikh R. A. Segmentation of connected characters in text-based CAPTCHAs for intelligent character recognition // Multimedia Tools and Applications. 2016. Vol. 76, Issue 24. P. 25547–25561. doi: https://doi.org/10.1007/s11042-016-4151-2

8. Recognition based segmentation of connected characters in text based CAPTCHAs / Hussain R., Gao H., Shaikh R. A., Soomro S. P. // 2016 8th IEEE International Conference on Communication Software and Networks (ICCSN). 2016. doi: https://doi.org/10.1109/iccsn.2016.7586608

9. Abdullah Hasan W. K. A Survey of Current Research on CAPTCHA // International Journal of Computer Science & Engineering Survey. 2016. Vol. 7, Issue 3. P. 1–21. doi: https://doi.org/10.5121/ijcses.2016.7301

10. Anti-captcha. URL: https://anti-captcha.com/mainpage/

11. Myroniv I. Development of the character recognition software on the base cellular authomata // VI-th International Conference of Students, PhD-Students and Young Scientists “Engineer of XXI Century”. 2016. P. 229–240.

12. OpenCV library. URL: https://opencv.org/

13. Леоненков А. В. Самоучитель UML. СПб.: БХВ Петербург, 2004. 576 с.

14. Fake Captcha is the #1 free fake captcha maker! URL: https://fakecaptcha.com/