

УДК 004.78.1

## АВТОМАТИЗАЦІЯ КОНТРОЛЮ ТА ОБРОБКИ ЗОБРАЖЕНЬ В ІНТЕРНЕТ-ДОДАТКАХ

Є. М. Грабовський

*Харківський національний економічний університет імені Семена Кузнеця,  
просп. Науки, 9а, Харків, 61001, Україна*

*Роз'яснено суть сучасних теоретичних підходів та програмні розробки до процесу перевірки зображень. Проведено аналіз інтернет-додатків, які дають змогу виконати online-обробку зображень. Досліджено інструменти, які дають змогу здійснювати перевірку зображень. Для отримання даних про зображення пропонується використовувати клас ImageInfo простору імен System.Drawing.Imaging у С#. На основі наявних параметрів пропонуються властивості, які описують параметри зображень. Подано аналіз інтегрованих середовищ розробки, на основі якого обґрунтовано вибір технології зберігання даних. Як програми для аналізу використовувалися безкоштовні середовища розробки та напівбезкоштовні, в яких є можливість написання програм на мові програмування С#. Розроблено прототип програмного засобу для реалізації процесу автоматизації online-перевірки зображень. За допомогою технології ASP.NET розроблено Web-сервіс, який забезпечує взаємодію створеного прототипу з онлайн-мережею.*

**Ключові слова:** зображення, автоматизація, контроль, обробка, інтернет-додатки, інтегровані середовища розробки.

**Постановка проблеми.** Автоматизація технологічних процесів обробки зображень є однією з центральних і практично важливих завдань на додрукарському етапі створення поліграфічної продукції. Для якісного друку оригінальних зображень в журналах і рекламної продукції потрібно не тільки якісне сканування, а й професійна ретуш і навіть монтаж оригінального зображення. Основна складність роботи з цифровими зображеннями пов'язана з обмеженими можливостями передачі інформаційних даних в рамках локальних мереж, Інтернету і потребою у формуванні в системах додрукарських процесів поліграфії цифрового графічного матеріалу необхідної якості.

Обробка зображень в реальному масштабі часу є вельми ресурсомісткою операцією і для її прискорення можливі різні підходи.

**Аналіз останніх досліджень та публікацій.** Наукові дослідження з тематики статті присвячені аналізу механізмів візуалізації інформації [1–3], створенню веббазованих інструментів обробки інформації в інтернет-додатках [2–5], розробці математичних моделей створення інтерфейсів мультимедійних видань [6–9], підтримці роботи препрес-інженера з огляду на механізми обробки графічних зображень [10–14]. Проте в спеціалізованій літературі немає науково-обґрунтованих

рекомендації щодо автоматизації контролю та обробки зображень в інтернет-додатках.

**Мета статті.** Основну мету дослідження орієнтовано на розробку автоматизованих засобів контролю та обробки зображень в інтернет-додатках.

**Виклад основного матеріалу дослідження.** Успіхом будь-якого образотворчого видання є якісні, придатні для друку зображення. Якісні зображення відіграють значну роль у поліграфічній продукції. Для отримання зображень придатної якості необхідно контролювати процес прийому замовлення. На сьогодні під час прийому замовлення важливо враховувати той фактор, хто буде контролювати цей процес. Це дуже трудомістка робота перевіряти зображення на відповідність заданим параметрам, які залежать від обладнання та технологічних операцій, що використовують на підприємстві.

Друкарня, де використовують тільки різграфи, не може дозволити собі кольоровий друк, тому зникає необхідність у зображеннях, що мають кольорову модель. Але не зникає необхідність у перевірці отриманого матеріалу, від якості якого у подальшому залежатиме друковане видання. Тому необхідно посилити контроль прийому матеріалу для друку на вході. Цим питаннями займається дизайнер або препрес-інженер, які перевіряють отриманий матеріал та оброблюють, якщо той не відповідає вимогам. Самі вимоги може сформулювати замовник, а вже видавництво чи друкарня вирішують, чи можуть вони з відповідною технічною базою впоратися з цими вимогами. Тому виникає певна проблема. Потенційний замовник не має миттєвого зворотного зв'язку з підприємством, не має миттєвої реакції на запит, тому що перевірка зображень залежить як від часового, так і від людського факторів. Процес спілкування замовника і виконавця в режимі online не дає змоги своєчасно приступити до виконання замовлення без попереднього обговорення деталей.

Завершуючи роботу, часто дизайнери і верстальники не помічають помилки на відповідність технічним вимогам друкарень. У програмах верстки існують перевірки, але вони зазвичай слабо розвинені і мають багато обмежень, які не дають можливості перевірити саме зображення на відповідність технічним вимогам. На сьогодні існує безліч програм, які допомагають контролювати процес обробки зображень на додрукарському етапі.

Отже, для досягнення поставленої мети статті необхідно провести аналіз вже існуючих підходів для вирішення цієї проблеми. Серед наявних програмних продуктів, які здатні робити контроль та обробку зображень, набув популярності csXImage ActiveX. Цей програмний продукт підтримує формати BMP, GIF, JPG, PNG, PSD, PCX, TIFF і дає змогу експортувати зображення в PDF. До його функціональних можливостей належить робота з розміром, масштабуванням, обрізанням і перевертанням, функції покращення якості зображення передбачають яскравість, контрастність, різкість, розмиття і відтінки сірого. Online-керівництво надає повний список доступних властивостей і методів. Відредаговані зображення можуть бути розміщені на Web-сервер як HTTP-завантаження або FTP. Серед наявних недоліків цього програмного продукту можна вважати відсутність модуля контролю зображень, а тільки можливість редагувати їх.

Найвдалішим рішенням перевірки на відповідність технічним вимогам став Adobe Acrobat. Перевірка готового PDF-файлу попередить про наявність помилок, які необхідно виправити до початку виготовлення, дає змогу значно знизити час виконання замовлення і знизити зайві витрати. Adobe Acrobat містить Preflight — інструмент, який аналізує весь документ на відповідність заданим технічним вимогам і після завершення видає детальний звіт. Розробники Adobe Acrobat дуже вдало підійшли до вирішення проблеми, розробивши три види профілів залежно від виду друку. Завдяки розробленим профілям можна виявляти помилки згідно із заданими технічними вимогами для газетного друку, для листового офсетного та ротативного офсетного друку. Серед типових помилок, що стосуються невідповідності зображень заданим вимогам, які дає змогу знайти цей профіль, існують такі:

- роздільна здатність кольорових і напівтонових растрових зображень менше 250 ppi;
- роздільна здатність кольорових і напівтонових растрових зображень більше 450 ppi;
- роздільна здатність однобітних растрових зображень менше 550 ppi;
- роздільна здатність однобітних растрових зображень більше 2400 ppi;
- об'єкт RGB.

Детальніше про описані вимоги можна сказати, що профіль дає змогу відстежувати важливі параметри растрових зображень, але вже на підготовчому етапі, а не на етапі прийому матеріалів. Що стосується дозволу зображень, то в Adobe Acrobat відстежується мінімальне значення 150 ppi, оскільки значення нижче будуть помітні під час друку на будь-якому, навіть дешевому, папері. Що стосується максимального значення, то воно вибирається з погляду доцільності.

Єдиним недоліком Preflight є неспроможність завжди видавати повну інформацію про помилки. Тому часто виникає необхідність провести додатковий візуальний контроль.

Ще одним із вдалих рішень в Adobe Acrobat можна вважати формат файлів PDF/X-1a. PDF/X-1a — стандартний формат файлів, який розроблений для обміну готовими документами, які призначені для друку. Його використання дає змогу відправнику і отримувачу не обговорювати додаткову інформацію і отримання необхідних результатів у тиражі, тобто цей формат обмежує зміст PDF-документа до обсягу, необхідного безпосередньо для відтворення високоякісного друку. Застосування цього формату усуває найпоширеніші помилки, пов'язані із зображеннями, які можна знайти під час підготовки файлів:

- неправильний кольоровий простір;
- проблема Overprint / Trapping.

Цей формат є стандартним для офсетного друку і його використання дає гарантії, що файл правильно підготовлений для друку на машині, на якій він буде друкуватися. Тобто завідомо у форматі передбачено, що всі кольори визначені як СМΥК та файл чітко ідентифікується як файл, що містить трепінг або не містить. Специфіка офсетного друку не дає змоги друкувати зображення із RGB. На виправлення помилок кожного документа, який було повернено у зв'язку з тим, що

його неправильно складено, необхідно витратити додатковий час та гроші. Використання PDF/X-1a дасть змогу зекономити витрати. Гірше, коли не виправлений документ піде до процесу виробництва, тоді є ризик втратити потенційного замовника або заплатити власні гроші.

З розвитком поліграфічної галузі розвиваються і способи друку, для яких розроблюються додаткові модулі, щоб якнайкраще задовольнити потреби найвибагливіших замовників. Альтернативою рішення формату для офсетного друку став формат PDF/X-3, призначений для цифрового друку. Цей формат, як і формат PDF/X-1a, здійснює перевірку і виявляє помилки, але вже дає змогу вирішити більш ширші проблеми, аніж PDF/X-1a, а саме використання системи управління кольором.

Наступним рішенням для контролю зображень є програма PitStop Professional, яка розроблена як модуль (плагін), що підключається до Adobe Acrobat. PitStop Professional може використовуватися для внесення виправлень в окремі сторінки і цілі документи. Функція Global Change дає змогу автоматизувати стандартні операції.

Виправлення у файл можуть бути внесені на будь-якому етапі підготовки документа. PitStop Professional містить набір інтерактивних інструментів для редагування та перевірки документів PDF — управління текстом, графікою, кольором та іншими атрибутами.

Інструменти редагування, що увійшли до складу пакета PitStop Professional, допомагають вносити виправлення і повторно використовувати готові документи з метою впорядкування обігу документів у форматі PDF. Вбудовані профілі PDF Profiles забезпечують автоматичне виправлення критичних помилок у файлах PDF. Можна тестувати файли на сумісність з новітніми специфікаціями PDF, відстежувати зміни, порівнювати версії, а також виконувати відкат до попередньої версії файлу та багато іншого. Під час створення профілю користувач заздалегідь може налаштувати виправлення можливих помилок. Якщо під час роботи виникає помилка, всі параметри користувача зберігаються без втрати даних. Якщо документ містить обрізані елементи, програма знайде їх і помістить в спеціальне місце в кеші для подальшого їх застосування, без пошкодження загального документа.

Проведемо аналіз інтернет-додатків, які дають змогу виконати online-обробку зображень. Так, додаток Cellsea надає різні можливості редагування, від зміни дозволу, повороту і обрізки до застосування різноманітних фільтрів. Програма підтримує роботу із зображеннями у форматах JPEG, GIF, PNG, BMP, TIFF розміром до 2 Мб. Ця online-система дає змогу застосовувати до зображень фільтри змашування або наведення різкості, спотворювати їх за допомогою фільтрів зсуву, калейдоскопа і ще двох десятків інших, додавати шум, мозаїку, рельєфність, а також керувати колірною гамою картинки.

Інтернет-додаток Pihoh також дає змогу редагувати зображення. Цей додаток відрізняється особливістю імпортувати зображення з різних Web-сайтів, виконувати основні інструменти редагування.

Іншим таким інтернет-додатком, у якому можна виконувати online-обробку, є FAN Studio. Цей додаток має низку функцій, які дають змогу редагувати зображен-

ня. Але у FAN Studio також нема змоги у повній мірі провести обробку зображень за потрібними параметрами.

Проведений аналіз наявних програмних продуктів та інтернет-додатків, які здатні робити контроль та обробку зображень, свідчить про відсутність на сьогодні достатньо функціонального додатку, який в змозі виконувати автоматизовану обробку графічних зображень. Розглянуті інструментальні засоби спроможні загалом на загальне редагування фотографій і не дають змоги виконати процес автоматизації обробки зображень.

Для розробки алгоритму автоматизації процесу контролю та обробки зображень було проведено дослідження, у процесі якого виявлено, що більшість середньостатистичних друкованих або електронних видань враховують деякі параметри. Такими параметрами для зображень є:

- кольоровий простір;
- реальне DPI;
- формат файлу;
- розмір (ширина і висота);
- розмір (пікселі об'єкта);
- масштабування зображення;
- наявність альфа-каналу;
- глибина кольору.

Якщо це електронне видання або просто зображення, що публікується в Інтернеті, то характеристики трохи відрізняються. Відмінності відображаються саме в застосуванні зображення. GIF-зображення зазвичай найкраще для графіки та малюнків, а JPEG — для фотографій. Для цього є дві причини: перша — GIF-зображення містять тільки до 256 кольорів, а JPEG — до мільйонів кольорів, і друге — формат GIF краще стискає прості зображення, ніж JPEG, який оптимізовано під більш складні зображення. Що вище стиснення, то менший файл зображення. Нижче наведені характеристики, які враховуються саме до цієї категорії.

Глибина кольору характеризує максимальне число кольорів, які використані у зображенні. Існує декілька типів зображень із різною глибиною кольору:

- чорно-білі;
- у відтінках сірого;
- з індексованими кольорами;
- повноколірні.

Тобто, враховуючи параметри, які використовуються в кожному конкретному випадку, чи то друк або підготовка електронного видання, будуть оброблятися описані вище характеристики зображень.

Для отримання даних про зображення варто використовувати клас `ImageInfo` простору імен `System.Drawing.Imaging` у `C#`. На основі наявних параметрів пропонуються властивості, які описують параметри зображень.

Першим параметром є формат зображення, що використовує властивість `Image.RawFormat`. Для отримання даних, що стосуються формату, необхідно оголосити перерахування окремого типу, що складається з набору іменованих констант, який називається списком перерахування у вигляді `public enum Format`.

Наступним параметром є кольоровий простір, що використовує властивість `ColorSpace` і перерахування окремого типу у вигляді `public enum ColorSpace`. У цьому перерахуванні вказуються всі моделі, які цікавлять замовника і які програма може опрацювати.

Дані про кольорову модель зображення можна отримати за допомогою бітового представлення інформації. Глибина кольору для цих моделей становить відповідно: чорно-біле, або штрихове зображення: 1 біт (2 кольори); відтінки сірого: 8 біт (256 відтінків сірого); обмежена кількість кольорів: 8 біт (256 кольорів); RGB — 16 біт (65538 кольорів). Зазвичай використовується глибина зображення у 8 біт, але це число може бути й меншим ніж 8. Повнокольорове зображення: 24 біт (16,7 мільйонів кольорів).

Отже, класи об'єктів виглядають так (рис. 1, рис. 2).

```
public class ImageInfo
{
    public ImageFormat Format { get; set; } //
    public ColorSpace ColorSpace { get; set; }
    public bool IsScalable { get; set; }
    public int Width { get; set; }
    public int Height { get; set; }
    public int Resolution { get; set; }
    public bool HasAlpha { get; set; }
    public int ColorDepth { get; set; }
    public int ImageSize { get { return Width * Height * ColorDepth; } }
}
```

Рис. 1. Клас об'єктів `ImageInfo`

```
public class ImageProfile
{
    public IList<ImageFormat> Formats { get; set; } //
    public IList<ColorSpace> ColorSpaces { get; set; }
    public bool Scalable { get; set; }
    public IDictionary<string, Size> Images { get; set; }
    public Range<int> Width { get; set; }
    public Range<int> Height { get; set; }
    public Range<int> Resolution { get; set; }
    public bool HasAlpha { get; set; }
    public Range<int> ColorDepth { get; set; }
    public Range<long> ImageSize { get; set; }
}
```

Рис. 2. Клас об'єктів `ImageService`

Наступним параметром є масштабованість зображення, яка використовує властивість `IsScalable` в створеному класі `ImageService`. Ця властивість допомагає

визначити чи має зображення здатність масштабуватися, тобто отримані дані про точки є масштабованими.

Важливою характеристикою з погляду друку є фізичний розмір зображення. Параметрами ширини та висоти зображення є властивості *Width* і *Height*. Визначення цих властивостей дає можливість замовнику контролювати один із суттєвих процесів при прийомі зображень, а саме його розмір в пікселях. У розробленій програмі вказані строгі значення кордону, що пропускатимуть допустимі значення ширини та висоти зображення. Значення параметрів ширини та висоти, які не вписалися в зазначений кордон, відсіваються як ті, що не пройшли перевірку.

Для визначення фізичного розміру зображення у сантиметрах чи міліметрах використовуються строгі фіксовані значення, якщо зображення їм не відповідає, то відправляється до папки із зображеннями, що не пройшли перевірку.

Для отримання даних про роздільність зображення використовується властивість *RealDPI*, що визначається в створеному класі *ImageService*, в якому виконується зчитування даних із зображень.

Отримання інформації про наявність альфа-каналу суттєво впливає на розмір зображення. Якщо альфа-канал використовується, то для збереження інформації в ньому необхідно додаткові 8 біт пам'яті, що впливає на збільшення розміру зображення. Альфа-канал — це порожній простір, або просто прозорість. У програмі цей параметр використовує властивість *HasAlpha*, що також визначається в створеному класі *ImageService*.

Значну роль відіграє розмір зображення, тобто об'єм, що зображення займає. Для його визначення використовуємо формулу (1):

$$\text{Розмір зображення} = W * H * ClDepth * Res, \quad (1)$$

де *W* — ширина зображення в пікселях;

*H* — висота зображення в пікселях;

*ClDepth* — глибина кольору в бітах;

*Res* — роздільна здатність зображення в dpi (дюймах на точку).

У програмі записуємо код у такому вигляді:

```
Public long ImageSize { get { return Width * Height * ColorDepth * Resolution; } }
```

Для вибору середовища розробки програмного модуля виконаємо аналіз інтегрованих середовищ розробки та на основі отриманих результатів зробимо висновки щодо доцільності використання того чи іншого середовища. Як програми для аналізу використовувалися безкоштовні середовища розробки та напівбезкоштовні, в яких є можливість написання програм на мові програмування C# (табл. 1, де «+» — наявність вказаного елементу, «-» — відсутність вказаного елементу).

З огляду на можливості кожного із розглянутих програмних середовищ розробки програмний модуль має бути розроблено у Microsoft Visual Studio, адже він має найбільшу функціональність. Оскільки прототип буде функціонувати online, то потрібно створити Web-сервіс, який має забезпечувати взаємодію розробленого

прототипу з онлайн-мережею. Web-сервіс був створений за допомогою технології ASP.NET в Visual Studio. Програмна платформа .NET Framework обробляє всю роботу нижніх рівнів, таку як створення, відсилання, прийняття та аналіз повідомлень SOAP.

Таблиця 1

### Аналіз інтегрованих систем розробки

ICP / Характеристики	Geany	Microsoft Visual Studio	MonoDevelop	SharpDevelop
Платформа	NIX/Windows	Windows	Cross-platform	GNU
Підсвічування синтаксису	+	+	+	+
Автозавершення	+	-	+	-
Можливість використання XML та CSS	-	+	-	-
Гнучкий інтерфейс	+	+	-	-
Наявність візуального конструктора	-	+	-	+
Можливість тестування вручну	-	+	-	-
Підтримка плагінів	-	+	+	+
Взаємодія з іншим програмним забезпеченням для створення багатого цифрового змісту	-	+	-	-

**Висновки.** Проаналізовано сучасні теоретичні підходи та програмні розробки до процесу перевірки зображень. Досліджено інструменти, які дають змогу здійснювати перевірку зображень. Обґрунтовано вибір технології зберігання даних. Розроблено програмний засіб для реалізації процесу автоматизації online-перевірки зображень. Створено Web-сервіс взаємодії розробленого прототипу з онлайн-мережею.

Отже, на своєчасний контроль та обробку зображень потрібні як трудові, так і матеріальні ресурси. Використання інтернет-додатків суттєво знизить витратні операції. Внаслідок підвищення популярності і широкого використання інтернет-технологій стало простіше спілкуватися із замовником. Завдяки використанню інтернет-технологій підприємство може значно скоротити процес виконання замовлення, трудові витрати та часові обмеження, що свідчить про актуальність запропонованої розробки в цьому дослідженні.



**СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ**

1. Hrabovskyi Y., Brynza N., Vilkhivska O. Development of information visualization methods for use in multimedia applications. *EUREKA: Physics and Engineering*. 2020. № 1. Pp. 3–17.
2. Khamula O. H., Soroka N. V., Vasiuta S. P. Factors of influence of interface use based on mobile applications. *Наукові записки [Української академії друкарства]*. 2016. № 2. С. 28–36.
3. Hryshchuk R. Synergetic control of social networking services actors' interactions. *Recent Advances in Systems, Control and Information Technology*. 2017. № 543. Pp. 34–42. DOI: [https://doi.org/10.1007/978-3-319-48923-0\\_5](https://doi.org/10.1007/978-3-319-48923-0_5).
4. Martins P. A Web-based Tool for Business Process Improvement. *International Journal of Web Portals*. 2017. № 9. Pp. 68 – 84. DOI: <https://doi.org/10.4018/IJWP.2017070104>.
5. Brambilla M. Large-scale Model-Driven Engineering of web user interaction: The WebML and WebRatio experience. *Science of Computer Programming*. 2014. № 89. Pp. 71–87. DOI: <https://doi.org/10.1016/j.scico.2013.03.010>.
6. Canessa E., Zennaro M. A Mobile Science Index for Development. *International Journal of Interactive Mobile Technologies*. 2012. № 6 (1). Pp. 4–6.
7. Norris D. Content Machine: Use Content Marketing to Build a 7-Figure Business With Zero Advertising. Kindle Edition, 2017. 164 p.
8. Khamula O. H., Soroka N. V., Vasiuta S. P. Optimization of mathematical model of the impact factors hierarchy of the interface use based on mobile. *Поліграфія і видавнича справа*. 2016. № 2 (72). С. 28–35.
9. Hrabovskyi Y., Fedorchenko V. Development of the optimization model of the interface of multimedia edition. *EUREKA: Physics and Engineering*. 2019. № 3. Pp. 3–12. DOI: 10.21303/2461-4262.2019.00902.
10. Mulisch M. Tissue-Printing. Springer. 2014. 40 p. DOI: 10.1007/978-3-658-03867-0.
11. Safonov I. Adaptive Image Processing Algorithms for Printing. Springer. 2018. 304 p. DOI: 10.1007/978-981-10-6931-4.
12. Hrabovskyi Y., Yevsyeyev O. Development of methodological principles of supportpreservation engineering work. *Technology audit and production reserves*. 2018. № 2 (2). Pp. 43–49. DOI: <https://doi.org/10.15587/2312-8372.2018.127776>.
13. Aralova N. I., Kyiashko O. Y. The Method of Technology Evaluation Based on Improved Cost Approach. *Science and Innovation*. 2017. № 13 (3). Pp 65–76. DOI:10.15407/scine13.03.065.
14. Kapela R., Guinness K., O'Connor N. Real-time field sports scene classification using colour and frequency space decompositions. *Journal of Real-Time Image Processing*. 2017. Vol. 13. Issue 4. Pp. 725–737. DOI: <https://doi.org/10.1007/s11554-014-0437-7>.

**REFERENCES**

1. Hrabovskyi, Y., Brynza, N., & Vilkhivska, O. (2020). Development of information visualization methods for use in multimedia applications: EUREKA: Physics and Engineering, 1, 3–17 (in English).
2. Khamula, O. H., Soroka, N. V., & Vasiuta, S. P. (2016). Factors of influence of interface use based on mobile applications: Naukovi zapysky [Ukrainskoi akademii drukarstva], 2, 28–36 (in English).

3. Hryshchuk, R. (2017). Synergetic control of social networking services actors' interactions: Recent Advances in Systems, Control and Information Technology, 543, 34–42. DOI: [https://doi.org/10.1007/978-3-319-48923-0\\_5](https://doi.org/10.1007/978-3-319-48923-0_5) (in English).
4. Martins, P. (2017). A Web-based Tool for Business Process Improvement: International Journal of Web Portals, 9, 68 – 84. DOI: <https://doi.org/10.4018/IJWP.2017070104> (in English).
5. Brambilla, M. (2014). Large-scale Model-Driven Engineering of web user interaction: The WebML and WebRatio experience: Science of Computer Programming, 89, 71–87. DOI: <https://doi.org/10.1016/j.scico.2013.03.010> (in English).
6. Canessa, E., & Zennaro, M. (2012). A Mobile Science Index for Development: International Journal of Interactive Mobile Technologies, 6 (1), 4–6 (in English).
7. Norris, D. (2017). Content Machine: Use Content Marketing to Build a 7-Figure Business With Zero Advertising. Kindle Edition (in English).
8. Khamula, O. H., Soroka, N. V., & Vasiuta, S. P. (2016). Optimization of mathematical model of the impact factors hierarchy of the interface use based on mobile: Polihrafiia i vydavnycha sprava, 2 (72), 28–35 (in English).
9. Hrabovskyi, Y., & Fedorchenko, V. (2019). Development of the optimization model of the interface of multimedia edition: EUREKA: Physics and Engineering, 3, 3–12. DOI: 10.21303/2461-4262.2019.00902 (in English).
10. Mulisch, M. (2014). Tissue-Printing. Springer. DOI: 10.1007/978-3-658-03867-0 (in English).
11. Safonov, I. (2018). Adaptive Image Processing Algorithms for Printing. Springer. DOI: 10.1007/978-981-10-6931-4 (in English).
12. Hrabovskyi, Y., & Yevsyeyev, O. (2018). Development of methodological principles of supportpreservation engineering work: Technology audit and production reserves, 2 (2), 43–49. DOI: <https://doi.org/10.15587/2312-8372.2018.127776> (in English).
13. Aralova, N. I., & Kyiashko, O. Y. (2017). The Method of Technology Evaluation Based on Improved Cost Approach: Science and Innovation, 13 (3), 65–76. DOI: 10.15407/scine13.03.065 (in English).
14. Kapela, R., Guinness, K., & O'Connor, N. (2017). Real-time field sports scene classification using colour and frequency space decompositions: Journal of Real-Time Image Processing, 13, 4, 725–737. DOI: <https://doi.org/10.1007/s11554-014-0437-7> (in English).

doi: 10.32403/1998-6912-2020-2-61-19-29

## AUTOMATION OF CONTROL AND PROCESSING OF IMAGES IN INTERNET APPLICATIONS

Ye. M. Hrabovskyi

*Semen Kuznets Kharkiv National University of Economics,  
9a, Nauky Avenue, Kharkiv, 61001, Ukraine  
maxmin903@gmail.com*

*The essence of modern theoretical approaches and software development to the process of image verification has been explained. The analysis of Internet applications*

*has been done that allows one to perform online image processing. Tools that allow image validation have been explored. It is suggested to use the ImageInfo class of the System.Drawing.Imaging namespace in C # to obtain image data. Based on existing parameters, properties are proposed that describe the parameters of the images. The study to develop an algorithm for automating the process of control and image processing has been conducted during which it was found that most of the average printed or electronic publications take into account some parameters. These parameters for images are color space, real DPI, file format, width and height, pixels of the object, image scaling, the presence of an alpha channel, color depth. Free development environments and semi-free ones, in which it is possible to write programs in the C # programming language, were used as analysis programs. To obtain the image resolution data, the RealDPI property is used, which is defined in the created ImageService class, in which the data is read from the images. A prototype of a software tool for the implementation of the process of automation of online image verification has been developed. Based on the capabilities of each of the considered software development environments, the software module should be developed in Microsoft Visual Studio, because it has the greatest functionality. With the help of ASP.NET technology, a Web service has been developed that provides the interaction of the created prototype with the online network. The .NET Framework handles all lower-level work, such as creating, sending, receiving, and analyzing SOAP messages.*

**Keywords:** *images, automation, control, processing, Internet applications, integrated development environments.*

*Стаття надійшла до редакції 09.11.2020.*

*Received 09.11.2020.*