

УДК 004.93

*Володимир Казимир, Руслан Заровський, Андрій Радченко***МЕТОД НОРМАЛІЗАЦІЇ ЗОБРАЖЕННЯ АВТОМОБІЛЬНОГО НОМЕРА ПРИ ЙОГО РОЗПІЗНАВАННІ ЗАСОБАМИ ВІДЕОСПОСТЕРЕЖЕННЯ***Владимир Казимир, Руслан Заровский, Андрей Радченко***МЕТОД НОРМАЛИЗАЦИИ ИЗОБРАЖЕНИЯ АВТОМОБИЛЬНОГО НОМЕРА ПРИ ЕГО РАСПОЗНАВАНИИ СРЕДСТВАМИ ВИДЕОНАБЛЮДЕНИЯ***Volodymyr Kazymyr, Ruslan Zarovsky, Andrii Radchenko***NORMALIZATION METHOD OF LICENSE PLATE IMAGE IN CASE OF ITS RECOGNITION BY MEANS OF VIDEO SURVEILLANCE**

Одним з найбільш швидких і надійних детекторів об'єктів є детектор Хаара. Однак цей тип детектора в разі його застосування для детектування автомобільних номерів не виділяє точні границі номера. Точне виділення границь необхідно для подальшого розпізнавання символів. Існуючі алгоритми нормалізації в багатьох випадках не справляються із завданням точного виділення границь автомобільного номера, що зменшує ймовірність правильного розпізнавання номера.

У цій роботі описано метод нормалізації зображення автомобільного номера після його детектування детектором Хаара з метою подальшого розпізнавання символів номера.

**Ключові слова:** детектор Хаара, нормалізація, диференціальне зображення, кластеризація.

Рис.: 7. Бібл.: 15.

Одним из самых быстрых и надежных детекторов объектов является детектор Хаара. Однако данный тип детектора в случае его применимости для детектирования автомобильных номеров не выделяет точные границы номера. Точное выделение границ необходимо для дальнейшего распознавания символов. Существующие алгоритмы нормализации во многих случаях не справляются с задачей точного выделения границ автомобильного номера, что уменьшает вероятность правильного распознавания номера.

В данной работе описывается метод нормализации изображения автомобильного номера после его детектирования детектором Хаара с целью дальнейшего распознавания символов номера.

**Ключевые слова:** детектор Хаара, нормализация, дифференциальное изображение, кластеризация.

Рис.: 7. Библ.: 15.

One of the most rapid and reliable detectors of objects is the Haar detector. However this type of detector in the case of its application for the detection of license plates doesn't select exact boundaries of plate. The precise boundaries selection is necessary for further character recognition. The existing normalization algorithms in many cases can not precisely select the boundaries of license plate, which reduces the probability of correct recognition of plate.

This paper describes a method of normalization after the license plate detection by Haar detector for further plate's character recognition.

**Key words:** Haar detector, normalization, differential image, clustering.

Fig.: 7. Bibl.: 15.

**Вступ.** Розпізнавання автомобільних номерів представляє великий практичний інтерес, тому що дозволяє скоротити витрати на організацію стоянок для автомобілів, своєчасно виявляти дорожньо-транспортних порушників та оперативно реагувати на події, пов'язані з використанням автомобільного транспорту.

Сучасні програмні продукти, які в змозі розпізнавати номери на відеопотоці, коштують досить дорого. Наприклад, система розпізнавання автомобільних номерів «Номерок 2» коштує 500 євро [1]. Алгоритми, що використовуються в таких комерційних продуктах, є комерційною таємницею, а відкриті алгоритми, які з великою вірогідністю можуть розпізнавати автомобільні номери, відсутні. З цих причин завдання розроблення методів та програмних засобів для розпізнавання автомобільних номерів є актуальною.

У процесі розпізнавання одним із вирішальних етапів є нормалізація зображення автомобільного номера, яка полягає в точному визначенні границь номера та кута його нахилу. Саме від якості нормалізації залежить загальний результат усього процесу розпізнавання автомобільного номера. Крім того, правильне визначення границь номера дає можливість виділити символи на брудних та засвічених номерах за умови, якщо відомий формат номера. Це збільшує загальну результативність усього процесу розпізнавання.

**Аналіз останніх досліджень і публікацій.** Часто для нормалізації автомобільного номера застосовуються оператори Собеля або Лапласа [2] з подальшим пошуком ліній за до-

## TECHNICAL SCIENCES AND TECHNOLOGIES

помогою перетворення Хафа [3]. Окремі методи використовують такий підхід як перший етап розпізнавання номера [4; 5], тим самим об'єднуючи детектування номера та нормалізацію. Але головним недоліком цього підходу є висока чутливість до різного роду шумів на зображенні (низька якість камери, брудний номер та нечітко виділені границі) [5].

У [6] для пошуку границь та кута повороту номера використовують гистограму яскравостей по горизонтальним лініям. Для цього номер циклічно повертають на певний кут за годинниковою і проти годинникової стрілки і будують гистограма яскравості для кожного такого повороту по нижній області задетектованого зображення. При цьому верхня область не враховувалась, оскільки вона може перебувати в тіні від автомобіля. Для виявлення верхньої межі використовувався детектор Хаара, який був навчений на знаходження символів. Для пошуку лівої і правої межі використовувалася гистограма яскравості по вертикальних лініях. Недолік такого підходу – низька швидкодія і багато випадків недостовірною результату.

**Метою статті** є опис запропонованого методу нормалізації автомобільного номера після його детектування детектором Хаара з метою подальшого розпізнавання символів номера, в основу якого покладено використання диференціальних зображень.

**Вимоги до вхідних та вихідних зображень.** Виходячи з проміжного положення етапу нормалізації при розробці методу нормалізації необхідно задати вимоги щодо вхідних та вихідних зображень. Для вхідних зображень ці вимоги повинні стосуватися значення відношення розміру номера до розміру задетектованої області та кута нахилу номера, а для вихідних – границь зображення щодо розташування символів та їх розміру.

Щодо значення відношення розміру номера до розміру задетектованої області необхідно врахувати, що детектор Хаара може задетектувати область вихідного зображення, на якій хоч і буде номер, але він буде займати лише дуже невелику частину задетектованої області. Наприклад, під час проведення експериментів при установці мінімального розміру об'єкта в  $100 \times 50$ , а максимального в  $500 \times 250$  пікселів на одному зображенні була задетектована область розміром  $450 \times 220$  пікселів при розмірі номера на цьому зображенні  $151 \times 35$  пікселів. У цьому прикладі детектор Хаара видав неправильний результат, оскільки він задетектував не тільки номер, а й всю передню частину автомобіля. До таких областей з номерами не можуть застосовуватися алгоритми нормалізації, адже в такому випадку, фактично, потрібна детекція номера, а не його нормалізація.

Стосовно куту нахилу було встановлено, що детектор Хаара, який ініціювався відповідним каскадом параметрів [7], не детектував номери з кутом нахилу більше  $5^\circ$ . Але зазвичай це не має практичного сенсу, бо при практичному розпізнаванні номерів детектування відбувається завжди у відеопотоці з деякої камери. При цьому більшість номерів мають кути нахилу в межах від  $-2^\circ$  до  $+2^\circ$ , а максимальний кут нахилу, який вдалось виявити на одному із таких відеопотоків становив  $+3,5^\circ$  (номер був неправильно закріплений на автомобілі).

Таким чином, вимоги до вхідних зображень методу полягають у такому:

- ширина номера не менше половини та не більше 0,9 ширини вхідного зображення;
- висота номера не менше третини та не більше 0,9 висоти вхідного зображення;
- максимальний кут нахилу номеру не більше  $5^\circ$ .

Зображення, які необхідні для подальшого розпізнавання номера, повинні задовольняти таким вимогам:

- границі результуючого зображення не обрізають символи номера;
- відстань між останньою літерою і правою границею результуючого зображення повинна становити не більше 0,1 його ширини;
- відстань між першою літерою і лівою границею зображення повинна становити не більше 0,15 його ширини;

- відстань між верхньою границею символів номера і верхньою границею результуючого зображення та нижньою границею букв і нижньою границею результуючого зображення повинна бути не більше третини висоти символів.

Саме зображення з такими вимогами будуть вважатися прийнятними для подальшого розпізнавання номера.

**Етапи методу нормалізації.** Запропонований метод нормалізації складається з таких етапів:

- 1) збільшення чіткості вихідного зображення;
- 2) побудова диференціальних зображень;
- 3) вирівнювання гістограми побудованих диференціальних зображень;
- 4) виявлення і групування прямих ліній по диференціальним зображенням;
- 5) вибірка горизонтальних ліній, що можуть вказувати на верхню та нижню границі номера;
- 6) визначення кута нахилу номера;
- 7) відсів вертикальних прямих, що не належать номеру;
- 8) визначення границь номера;
- 9) перевірка адекватності визначених границь.

**Збільшення чіткості вихідного зображення.** Збільшення чіткості вихідного зображення підвищує результативність усього методу нормалізації. Найпростіший варіант поліпшення чіткості вихідного зображення – використання операції згортки з таким оператором [8]:

$$\begin{bmatrix} -0.1 & -0.1 & -0.1 \\ -0.1 & 2 & -0.1 \\ -0.1 & -0.1 & -0.1 \end{bmatrix} \quad (1)$$

**Побудова диференціального зображення.** Принцип побудови спеціального зображення, яке в цій статті називається диференціальним, заснований на виконанні операції згортки з такими диференціальними операторами [9]:

$$\begin{bmatrix} 1 & -1 \end{bmatrix}, \begin{bmatrix} 1 \\ -1 \end{bmatrix} \quad (2)$$

Але на відміну від виконання згортки з такими операторами, кожен піксель диференціального зображення являє собою різницю або модуль різниці між пікселем вихідного зображення і його горизонтальним чи вертикальним сусідом, при виконанні деякої умови або без умови. Умовні позначання сусідів пікселя та осей координат зображення показані на рис. 1.

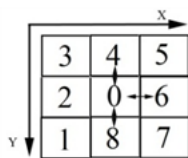


Рис. 1. Центральний піксель, його сусіди та осі координат зображення

Для запропонованого методу нормалізації необхідно побудувати три диференціальні зображення.

Перше диференціальне зображення являє собою зображення того ж розміру, що і вихідне, в якому яскравість кожного пікселя дорівнює модулю різниці між пікселем вихідного зображення і правим його сусідом (на рис. 1 0-й піксель – піксель, що розглядається, 6-й піксель – його правий сусід).

Друге диференціальне зображення являє собою зображення того ж розміру, що і вихідне, в якому яскравість кожного пікселя дорівнює різниці між пікселем вихідного зо-

## TECHNICAL SCIENCES AND TECHNOLOGIES

браження і нижнім його сусідом (0-й і 8-й пікселі рис. 1) за умови, що яскравість пікселя більше яскравості його нижнього сусіда.

Третє диференціальне зображення являє собою зображення того ж розміру, що і вихідне, в якому яскравість кожного пікселя дорівнює різниці між пікселем вихідного зображення і верхнім його сусідом (0-й і 4-й пікселі рис. 1) за умови, що яскравість пікселя більше яскравості його верхнього сусіда.

Яскравість пікселів диференціальних зображень, які не мають відповідних сусідів або для яких не виконується відповідна умова, встановлюється в 0.

**Вирівнювання гістограми диференціальних зображень.** Необхідність цього кроку пов'язана з тим, що діапазон яскравостей пікселів побудованих диференціальних зображень зазвичай менше половини можливого діапазону яскравостей пікселів (0-255). Збільшення діапазону здійснюється шляхом вирівнювання гістограми зображення [10], що є набагато простіше, ніж розрахунок порогового значення, за яким буде визначатись можливість належності пікселя лінії на четвертому кроці.

На рис. 2 показані вихідні зображення (а) та побудовані для них диференціальні зображення з вирівняними гістограмами (б, в, г).



Рис. 2. Вхідне (а) та диференціальні зображення з різницею 0-20 і 6-20 (б), 4-20 і 0-20 (в) та 0-20 і 8-20 (г) пікселів

**Вибірка і групування прямих ліній по диференціальному зображенню.** На цьому кроці методу на побудованих зображеннях необхідно знайти вертикальні та горизонтальні прямі лінії.

На вихідних зображеннях номер не може займати менше половини ширини зображення (одна з умов вихідних зображень), але через те, що номер може знаходитися під кутом або через те, що границі номера можуть бути погано виділені, необхідно шукати горизонтальні лінії, які займають менше половини ширини вихідного зображення. Крім того, необхідно вибрати поріг яскравості, значення якого буде вказувати на те, що певний піксель може належати до якоїсь лінії.

Емпіричним шляхом було встановлено, що поріг яскравості величиною 127 (при діапазоні 0-255), мінімальний розмір лінії по горизонталі рівний восьмій частині ширини вихідного зображення і мінімальний розмір лінії по вертикалі рівний шостій частині висоти вихідного зображення є оптимальними для цього завдання.

Координати знайдених прямих ліній необхідно зберегти для подальшої обробки. На рис. 3 показані знайдені лінії для відповідних диференціальних зображень з рис. 2 (б, в, г).

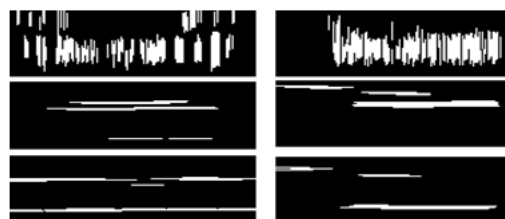


Рис. 3. Знайдені прямі лінії для відповідних диференціальних зображень

Як свідчать наведені зображення, область номера досить добре проглядається на зображенні з вертикальними лініями. Однак як вертикальні, так і горизонтальні лінії самі по собі часто не можуть свідчити про межі номера. Так, на першому номері вертикальні лінії захоплюють зліва і справа невелику область, що не відноситься до номера, а горизонтальні лінії дають відповідність по висоті, але не по ширині. Крім того, на деяких номерах без вертикальних ліній неможливо встановити, які ж горизонтальні лінії відносяться до номера через велику їх кількість. Тому пропонується використовувати комбінацію горизонтальних і вертикальних ліній для визначення меж номера.

Для визначення границь номера по горизонталі необхідно провести угруповання ліній з метою зменшення їх кількості та визначення більш точного положення границь номера по осі Y. Фактично необхідно провести кластеризацію [11] з об'єднанням сусідніх ліній. Причому об'єднувати потрібно тільки ті лінії, які були отримані з того ж самого диференціального зображення. Не слід змішувати і групувати лінії, отримані з різних диференціальних зображень. Оскільки вихідне число кластерів є невідомим, то слід використовувати ієрархічний метод кластеризації [12], а саме – агломеративний метод [12], який виконує угруповання вихідних кластерів (ліній). Кластеризацію горизонтальних ліній слід проводити тільки за значеннями осі Y, оскільки через розташування номера під кутом кластеризація по осі X може привести до неправильних результатів. Відстань між кластерами слід оцінювати по відстані між їх середніми значеннями, які являють собою середне по осі Y для прямих, що входять у кластер.

Однак найбільша проблема ієрархічного кластерного аналізу, яка достовірно не вирішується жодним чином [13], – умова зупинки кластеризації. Емпірично було встановлено, що кластеризацію для горизонтальних ліній слід припинити, як тільки відстань між кластерами стане більшою, ніж десята частина висоти вихідного зображення. Крім того, у процесі кластеризації необхідно визначити мінімальне і максимальне значення по осі Y для прямих, які входять в отримані кластери.

При угрупованні горизонтальних прямих за їх значенням по осі Y групуються прямі, які можуть не з'єднуватися одна з одною. На рис. 4 показані розриви прямої одного з зображень рис. 3 (один з розривів непомітний через алгоритм відображення прямих).



Рис. 4. Розриви лінії

З вихідного зображення (рис. 2, а) видно, що як ліва, так і права частина прямої не відноситься до номера. Отже, у цьому випадку вони не повинні враховуватися. Що стосується інших номерів, які були задетектовані у процесі розроблення методу, то було встановлено: відсіювання прямих, які не з'єднуються з найдовшим ланцюжком прямих, які «торкаються» одна одній у кластері, позитивно позначається на кінцевому результаті нормалізації. Було проаналізовано 598 вихідних зображень номерів (одна з вибірок), і тільки на одному з них був присутній розрив, який поділяв нижню межу номера (нижній кластер) на дві частини й ускладнював подальшу нормалізацію. На інших номерах розривів або не було, або їх усунення покращувало нормалізацію. Тому для всіх кластерів необхідно знайти найдовший ланцюжок пов'язаних прямих і визначити мінімальне і максимальне значення цього ланцюжка по осі X. У цей ланцюжок повинні входити усі лінії, які з'єднуються один з одним.

Результати кластеризації горизонтальних прямих на рис. 3 після відсіву прямих, що не з'єднуються з найдовшим ланцюжком, показані на рис. 5.



Рис. 5. Результати кластеризації і відсіву прямих

Для вертикальних ліній також необхідно виконати кластеризацію по осі Y. Відмінність від попереднього випадку полягає тільки в тому, що для горизонтальних ліній значення по осі Y однакове для всіх точок прямої, а для вертикальних ліній воно змінюється. Як показали дослідження, слід проводити кластеризацію вертикальних ліній по двох точках – по верхній і по нижній точці вертикальної лінії. Тобто спочатку необхідно сформувати набір значень верхніх точок всіх вертикальних прямих по осі Y. Потім над цим набором необхідно провести кластеризацію. Далі необхідно сформувати набір значень за значеннями осі Y нижньої точки всіх вертикальних прямих і знову здійснити кластеризацію.

Проблема зупинки кластеризації залишається відкритою, проте для цієї задачі було встановлено, що кластеризацію слід обов'язково продовжувати до того моменту, поки відстань між кластерами не стане більшою, ніж десята частина висоти вихідного зображення. Після цього її слід продовжувати за умови, що відстань між кластерами менше третини висоти вихідного зображення або різниця відстаней між попередньою і поточною ітерацією становить менше десятої частини висоти вихідного зображення. Для зазначених на початку обмежень така кластеризація дозволяє відсіяти вертикальні прямі, які не належать номеру.

**Використання результатів угруповання прямих ліній для визначення меж і кута нахилу номера.** На цьому кроці перш за все необхідно вибрати два кластери (один – з кластерів верхньої точки, другий – з кластерів нижньої точки), які містять найбільшу кількість точок. На всіх задетектованих номерах саме такі кластери відносились до області номера.

Далі необхідно сформувати всі можливі пари кластерів, які були отримані з угруповання горизонтальних прямих (далі вони будуть називатись «горизонтальними кластерами») і які можуть показувати верхню і нижню межі номера. Пари повинні складатися з кластера, отриманого з диференціального зображення, що показує верхню межу номера (з різницею 8-го і 0-го пікселів) і кластера, що показує нижню межу номера (з різницею 0-го і 8-го пікселів). Пару горизонтальних кластерів, яка може представляти межі номера, слід визначати за умовою відстані по осі Y між цими кластерами. Якщо відстань між ними більше, ніж третина висоти вихідного зображення – кластери можуть являти собою пару, яка показує верхню і нижню межі номера. В іншому випадку – кластери не можуть вказувати на межі номера. Якщо не існує жодної пари кластерів, яка б відповідала вищезазначеній умові, то визначення границь номера за допомогою горизонтальних кластерів та кута його повороту виявляється неможливим. Тому необхідно одразу перейти на крок визначення границь номера за умови невідповідності горизонтальних кластерів області номера (8-й крок).

Після формування всіх можливих пар горизонтальних кластерів необхідно вибрати найбільш ймовірну пару, яка буде вказувати на верхню і нижню межу номера. Для такого визначення необхідно використовувати кластери, отримані з кластеризації вертикальних ліній. Для опису алгоритму вибору пари горизонтальних кластерів, введемо такі позначення:

- 1)  $\min X_1, \min X_2$  – початкові координати по осі X ліній, що входять у верхній і нижній горизонтальний кластер відповідно;
- 2)  $\max X_1, \max X_2$  – кінцеві координати по осі X ліній, що входять у верхній і нижній горизонтальний кластер відповідно;
- 3)  $\text{width}, \text{height}$  – ширина і висота вихідного зображення;
- 4)  $\text{value}H_1, \text{value}H_2$  – значення першого і другого горизонтального кластерів;
- 5)  $\text{value}V_1, \text{value}V_2$  – значення першого і другого вертикального кластерів;
- 6)  $\text{abs}(x)$  – абсолютне значення змінної  $x$ ;

7)  $\&\&$  – логічне «І».

Алгоритм вибору пари горизонтальних кластерів такий:

1) вибираємо поточну пару кластерів;

2) якщо пара відповідає умові:

$\text{abs}(\text{minX1} - \text{minX2}) < 0.2 * \text{width} \&\& \text{abs}(\text{maxX1} - \text{maxX2}) < 0.2 * \text{width} \&\& \text{valueH1} \leq \text{valueV1} + 0.05 * \text{height} \&\& \text{valueH2} \geq \text{valueV2} - 0.05 * \text{height}$ ,

то розраховуємо для такої пари відхилення значень горизонтальних кластерів від кластерів вертикальних (відхилення –  $\text{abs}(\text{valueH1} - \text{valueV1}) + \text{abs}(\text{valueH2} - \text{valueV2})$ ) і запам'ятовуємо таку пару;

3) повторюємо другий крок для всіх пар;

4) якщо кількість пар, які відповідають умові на кроці 2 не дорівнює 0, то вибираємо пару з мінімальним відхиленням;

5) в іншому випадку рахуємо відхилення для всіх пар і вибираємо пару з мінімальним відхиленням.

У цьому алгоритмі на кроці 2 до значень вертикальних кластерів додається і віднімається  $0.05 * \text{height}$  з тієї причини, що на деяких номерах значення вертикальних кластерів може бути трохи вище або нижче, ніж значення горизонтальних кластерів. Хоча здебільшого значення першого вертикального кластера більше (на зображеннях вісь Y збільшується вниз) значення горизонтального кластера, а значення другого вертикального кластера навпаки – менше.

Після вибору найбільш імовірної пари горизонтальних кластерів, які вказують на верхню і нижню границі номера, можна визначити кут нахилу номера за допомогою прямих, які входять у такі кластери. Для цього можна використовувати перетворення Хафа [3] або метод головних компонент [14]. У поточній реалізації методу для визначення кута нахилу номера використовується метод головних компонент, реалізація якого доступна в бібліотеці OpenCV [15]. Єдине перетворення даних, яке потрібно виконати до виклику стандартних методів бібліотеки, – створення масиву точок із множини прямих, що входять в горизонтальні кластери. Причому підрахунок кута за допомогою методу головних компонент запускається два рази – для набору точок прямих кластера, що вказує на верхню межу, і для набору точок прямих кластера, який вказує на нижню межу. Загальний кут визначається середнім значенням двох розрахованих кутів. Далі вихідне зображення повертається на вирахований кут. Приклади вихідних і результуючих зображень після повороту показані на рис. 6.

Після повороту для номерів, у яких кут нахилу виявився більше  $1^\circ$ , слід повторити кроки починаючи від визначення горизонтальних і вертикальних прямих ліній і закінчуючи кроком повороту. Причина необхідності повтору цих кроків полягає в тому, що при кластеризації вертикальних ліній деякі лінії, які відносяться до області номера, можуть через відносно великий кут повороту ( $>1^\circ$ ) прираховуватись до кластерів, які не відносяться до області номера. Через це при визначенні границь номера зображення зліва або справа може сильно обрізатись, що унеможливить подальше розпізнавання номера.



Рис. 6. Зображення до і після повороту на розрахований кут

**Відсів вертикальних прямих.** На цьому кроці відсіюються вертикальні прямі, які водночас не входять в обидва сформованих вертикальних кластера. При цьому також

## TECHNICAL SCIENCES AND TECHNOLOGIES

необхідно відсіяти прямі, які входять у ці кластери, але які знаходяться на зображенні на певній відстані ліворуч і праворуч від основної групи прямих (прямі, що знаходяться на віддалі від більшості інших прямих, не належать номеру). У ході досліджень було встановлено, що відсів прямих, що лежать на відстані один від одного більшому, ніж 0,12 ширини вихідного зображення, позитивно позначається на кінцевому результаті нормалізації.

**Визначення границь номера.** Спочатку вибираються межі по осі X. При цьому алгоритм вибору лівої границі включає такі кроки:

1) вибрати координату по осі X (назвемо її  $\min X_{ByY}$ ) найлівішої вертикальної прямої (після проведеного раніше відсіву);

2) порівняти  $\min X_{ByY}$  з мінімальною координатою по осі X верхнього горизонтального кластера (назвемо її  $\min X_{ByH1}$ ). Якщо різниця між ними не більше 0,02 ширини вихідного зображення, то ліву границю номера визначити як мінімальне значення  $\min X_{ByY}$  і  $\min X_{ByH1}$ ;

3) в іншому випадку порівняти різницю мінімальної координати по осі X нижнього горизонтального кластера ( $\min X_{ByH2}$ ) з  $\min X_{ByY}$ . Якщо різниця між ними не більше 0,02 ширини вихідного зображення, то ліву границю номера визначити як мінімальне значення  $\min X_{ByY}$  і  $\min X_{ByH2}$ ;

4) в іншому випадку порівняти різницю  $\min X_{ByH1}$  і  $\min X_{ByH2}$ . Якщо різниця між ними не більше десятої частини ширини номера, то ліву границю номера визначити як мінімальне значення між  $\min X_{ByH1}$  і  $\min X_{ByH2}$ . Але при цьому якщо різниця між  $\min X_{ByY}$  і визначеної межею більше десятої частини ширини вихідного зображення, то ліву границю слід визначити як  $\min X_{ByY}$ ;

5) в іншому випадку ліва межа –  $\min X_{ByY}$ .

Алгоритм вибору правої границі аналогічний алгоритму вибору лівої границі (тільки використовуються не мінімальні координати по осі X, а максимальні).

Верхня і нижня межі номера визначаються за значеннями верхнього і нижнього горизонтальних кластерів відповідно.

На деяких номерах було виявлено, що обрані горизонтальні лінії можуть не відповідати номеру. Через це вибір границь (як по осі X, так і по осі Y) за допомогою горизонтальних кластерів може бути неадекватним. Тому необхідно перевіряти відхилення значень горизонтальних кластерів від значень кластерів вертикальних. Введемо наступні позначення для опису умов, за яких горизонтальні кластери найімовірніше не належать номеру:

1)  $value1$ ,  $value2$  – значення верхнього та нижнього горизонтальних кластерів;

2)  $maxPointMinus$ ,  $maxPointPlus$  – мінімальне і максимальне значення по осі Y прямих, що входять у верхній і нижній вертикальний кластер;

3)  $height$  – висота вихідного зображення;

4)  $\&\&$  – логічне «І».

Умова, за якої верхній горизонтальний кластер слід вважати таким, що не належить номеру, полягає в наступному:

$$value1 < maxPointMinus \&\& maxPointMinus - value1 > 0.08 * height.$$

Умова, за якої нижній горизонтальний кластер слід вважати таким, що не належить номеру:

$$value2 > maxPointPlus \&\& value2 - maxPointPlus > 0.08 * height.$$

Якщо горизонтальний кластер не належить номеру, то слід змінити ту границю по осі Y, яка була визначена за допомогою цього кластера. Тобто для верхнього кластера – це верхня границя, а для нижнього – нижня. Границю по осі Y слід визначити як  $maxPointMinus$  або  $maxPointPlus$  при невідповідності верхнього або нижнього класте-



рів. Також у цьому випадку слід визначити ліву і праву границю номера як  $\min X_{ByY}$  і  $\max X_{ByY}$  відповідно.

**Перевірка адекватності визначених границь.** На цьому кроці методу слід перевірити співвідношення висоти до ширини визначеної області, оскільки на одному номері було виявлено, що на нижній лінії номера був розрив, через який горизонтальні кластери неправильно визначали праву границю номера – вона була визначена приблизно по центру номера. Тому, якщо відношення ширини до висоти номера більше 0,32, то ліву і праву межу слід встановити у значення  $\min X_{ByY}$  і  $\max X_{ByY}$  відповідно.

**Результати.** Конкретна реалізація методу для вихідних зображень з вказаними на початку обмеженнями була протестована в режимі реального часу за допомогою камери, спрямованої на автомобільну трасу. Було зафіксовано близько 870 різних автомобільних номерів (всього було зафіксовано 1741 вихідне зображення, враховуючи, що деякі номери детектувались по 2 рази). Із цієї виборки алгоритм дав збій тільки на чотирьох автомобільних номерах. Приклади правильної і неправильної роботи алгоритму показані на рис. 7.

**Висновки.** Створений метод справляється з поставленим завданням нормалізації. Здебільшого він точно знаходить границі номера, що дозволяє визначити розташування символів для їх подальшого розпізнавання.

Правильність роботи методу залежить від зображень, які подаються на його вхід та від параметрів алгоритмів, які використовуються в ньому. Однак якщо говорити про ймовірності, які були отримані в результаті тестування описаного методу для конкретної вибірки, то відсоток правильних результатів становить  $(870 - 4) / 870 = 99,54\%$ , а неправильних –  $0,46\%$ .

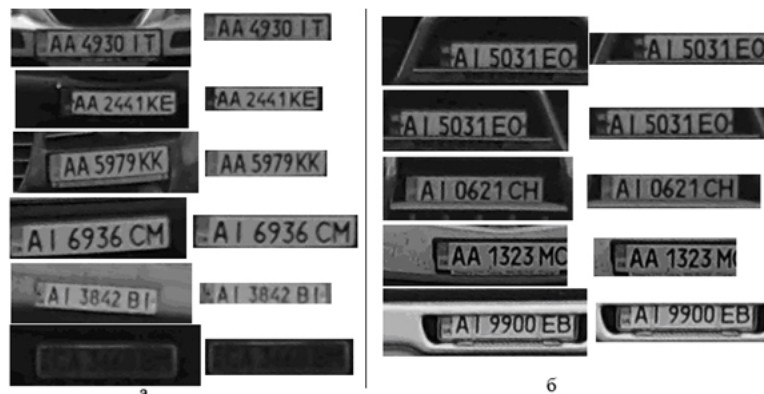


Рис. 7. Вхідні зображення і правильні результати роботи методу (а) та вхідні зображення і неправильні результати роботи методу (б)

Водночас, описаний метод має свої недоліки. По-перше, слабким місцем методу є відсіви вертикальних прямих, які не належать номеру. На поточний момент умова відсіву прямих задана статично і для поточної задачі становить 0,12 ширини вихідного зображення. Але для деяких номерів необхідно відкидати прямі, які знаходяться на відстані в 0,08 ширини вихідного зображення. Доопрацювання цієї частини методу з метою динамічного визначення відстані, за якої необхідно відкидати вертикальні прямі, дозволить правильно визначати границі номера для вихідних зображень з бамперами, заглибленнями і т. ін.

По-друге, умови зупинки кластеризації основані на висоті вихідного зображення. Через це знижується правильність кластеризації, що погіршує кінцевий результат роботи методу. Якщо умова зупинки кластеризації не буде залежати від висоти вихідного зображення, то, по-перше, зросте точність визначення границь номера, а по-друге, метод з деякими модифікаціями можна буде застосовувати для детектування номерів.

## TECHNICAL SCIENCES AND TECHNOLOGIES

По-третє, метод має досить багато кроків, що негативно позначається на його швидкодії. Програма, за допомогою якої відбувалось тестування конкретної реалізації методу, використовує каскад Хаара для детектування автомобільних номерів у відеопотоці. Роздільна здатність кадрів, які подаються на вхід програми, становить  $878 \times 552$ . Мінімальний розмір номера був встановлений у  $100 \times 50$ , а максимальний – у  $190 \times 90$ . Поточна реалізація методу завантажувала процесор приблизно на 60 % менше, ніж детектування номерів при таких параметрах. Однак для кроку нормалізації це досить великий результат.

Подальша робота над зазначеними проблемами дозволить поліпшити якість цього методу.

**Список використаних джерел**

1. Система распознавания автомобильных номеров «НомерОК» [Электронный ресурс]. – Режим доступа : <http://avtonomerok.su>.
2. Оператор Собеля и Лапласа [Электронный ресурс]. – Режим доступа : <http://robocraft.ru/blog/computervision/460.html>.
3. OpenCV шаг за шагом. Преобразование Хафа [Электронный ресурс]. – Режим доступа : <http://robocraft.ru/blog/computervision/502.html>.
4. Martinsky O. Algorithmic and mathematical principles of automatic number plate recognition systems [Электронный ресурс] / O. Martinsky. – Режим доступа : <http://javaanpr.sourceforge.net/anpr.pdf>.
5. Rasheed S., Naeem A., Ishaq O. Automated number plate recognition Using Hough Lines and template matching [Электронный ресурс] / S. Rasheed, A. Naeem, O. Ishaq. – Режим доступа : [http://www.iaeng.org/publication/WCECS2012/WCECS2012\\_pp199-203.pdf](http://www.iaeng.org/publication/WCECS2012/WCECS2012_pp199-203.pdf).
6. Распознавания автомобильных номеров в деталях [Электронный ресурс]. – Режим доступа <https://habrahabr.ru/company/recognitor/blog/225913/> (Дата доступа 28.09.2016)
7. haarcascade\_russian\_plate\_number.xml [Электронный ресурс]. – Режим доступа : [https://github.com/opencv/opencv/blob/master/data/haarcascades/haarcascade\\_russian\\_plate\\_number.xml](https://github.com/opencv/opencv/blob/master/data/haarcascades/haarcascade_russian_plate_number.xml).
8. OpenCV шаг за шагом. Обработка изображения – свёртка [Электронный ресурс]. – Режим доступа : <http://robocraft.ru/blog/computervision/427.html>.
9. Фисенко В. Т. Компьютерная обработка и распознавание изображений / В. Т. Фисенко, Т. Ю. Фисенко. – СПб. : СПбГУ ИТМО, 2008. – 192 с.
10. Histogram Equalization [Электронный ресурс]. – Режим доступа : [http://docs.opencv.org/2.4/doc/tutorials/imgproc/histograms/histogram\\_equalization/histogram\\_equalization.html](http://docs.opencv.org/2.4/doc/tutorials/imgproc/histograms/histogram_equalization/histogram_equalization.html).
11. Кластеризация [Электронный ресурс]. – Режим доступа : <http://www.machinelearning.ru/wiki/index.php?title=Кластеризация>.
12. Методы кластерного анализа. Иерархические методы [Электронный ресурс]. – Режим доступа : <http://www.intuit.ru/studies/courses/6/6/lecture/182?page=2>.
13. Бериков В. Б. Современные тенденции в кластерном анализе [Электронный ресурс] / В. Б. Бериков, Г. С. Лбов. – Режим доступа : <http://www.ict.edu.ru/ft/005638/62315e1-st02.pdf>.
14. OBJECT ORIENTATION, PRINCIPAL COMPONENT ANALYSIS & OPENCV [Электронный ресурс]. – Режим доступа : <https://robospace.wordpress.com/2013/10/09/object-orientation-principal-component-analysis-opencv/>.
15. OpenCV [Электронный ресурс]. – Режим доступа : <http://opencv.org/>.

**References**

1. Sistema raspoznavanija avtomobilnykh nomerov «NomerOK» [The car's plate recognition system «NomerOK»]. Retrieved from <http://avtonomerok.su/>.
2. Operator Sobelja i Laplasa [Sobel and Laplace operator]. Retrieved from <http://robocraft.ru/blog/computervision/460.html>.
3. OpenCV shag za shagom. Preobrazovanie Hafa [OpenCV step by step. Hough transformation]. Retrieved from: <http://robocraft.ru/blog/computervision/502.html>.
4. Martinsky O. Algorithmic and mathematical principles of automatic number plate recognition systems. Retrieved from <http://javaanpr.sourceforge.net/anpr.pdf>.

5. Rasheed S., Naeem A., Ishaq O. *Automated number plate recognition Using Hough Lines and template matching*. Retrieved from [http://www.iaeng.org/publication/WCECS2012/WCECS2012\\_pp199-203.pdf](http://www.iaeng.org/publication/WCECS2012/WCECS2012_pp199-203.pdf).
6. *Raspoznavaniia avtomobilnykh nomerov v detaliakh [License plate recognition details]*. Retrieved from <https://habrahabr.ru/company/recognitor/blog/225913/>.
7. *haarcascade\_russian\_plate\_number.xml*. Retrieved from: [https://github.com/opencv/opencv/blob/master/data/haarcascades/haarcascade\\_russian\\_plate\\_number.xml](https://github.com/opencv/opencv/blob/master/data/haarcascades/haarcascade_russian_plate_number.xml).
8. *OpenCV shag za shagom. Obrabotka izobrazheniia – sviortka [OpenCV step by step. Image processing – convolution]*. Retrieved from <http://robocraft.ru/blog/computervision/427.html>.
9. Fisenko, V.T. (2008). *Kompiuternaia obrabotka i raspoznavanie izobrazhenii [Computer image processing and recognition]*. Saint-Petersburg: SPbGUITMO Publisher (In Russian).
10. *Histogram Equalization*. Retrieved from [http://docs.opencv.org/2.4/doc/tutorials/imgproc/histograms/histogram\\_equalization/histogram\\_equalization.html](http://docs.opencv.org/2.4/doc/tutorials/imgproc/histograms/histogram_equalization/histogram_equalization.html).
11. *Klasterizatsiia [Clusterization]*. Retrieved from <http://www.machinelearning.ru/wiki/index.php?title=Кластеризация>.
12. *Metody klasterного analiza. Ierarkhicheskie metody [Methods of cluster analysis. Hierarchical methods]*. Retrieved from <http://www.intuit.ru/studies/courses/6/6/lecture/182?page=2>.
13. Berikov, V.B., Lbov, G.S. *Sovremennye tendentsii v klasterном analize [Current trends in the cluster analysis]*. Retrieved from <http://www.ict.edu.ru/ft/005638/62315e1-st02.pdf>.
14. *OBJECT ORIENTATION, PRINCIPAL COMPONENT ANALYSIS & OPENCV*. Retrieved from <https://robospace.wordpress.com/2013/10/09/object-orientation-principal-component-analysis-opencv/>.
15. *OpenCV*. Retrieved from <http://opencv.org/>.

**Казимир Володимир Вікторович** – доктор технічних наук, професор, Чернігівський національний технологічний університет (вул. Шевченка, 95, м. Чернігів, 14027, Україна).

**Казимир Владимир Викторович** – доктор технических наук, профессор, Черниговский национальный технологический университет (ул. Шевченко, 95, г. Чернигов, 14027, Украина).

**Kazymyr Volodymyr** – Doctor of Technical Sciences, Professor, Chernihiv National University of Technology (95 Shevchenka Str., 14027 Chernihiv, Ukraine).

**E-mail:** [vvkazymyr@gmail.com](mailto:vvkazymyr@gmail.com)

**ORCID:** <http://orcid.org/0000-0001-8163-1119>

**Scopus Author ID:** 56644727300

**Заровський Руслан Владиславович** – кандидат технічних наук, доцент, Чернігівський національний технологічний університет (вул. Шевченка, 95, м. Чернігів, 14027, Україна)

**Заровский Руслан Владиславович** – кандидат технических наук, доцент, Черниговский национальный технологический университет (ул. Шевченко, 95, г. Чернигов, 14027, Украина).

**Zarovsky Ruslan** – PhD in Technical Sciences, Associate Professor, Chernihiv National University of Technology (95 Shevchenka Str., 14027 Chernihiv, Ukraine).

**E-mail:** [rolandzar@ukr.net](mailto:rolandzar@ukr.net)

**Радченко Андрій Олексійович** – аспірант, Чернігівський національний технологічний університет (вул. Шевченка 95, м. Чернігів, 14027, Україна).

**Радченко Андрей Алексеевич** – аспирант, Черниговский национальный технологический университет (ул. Шевченка 95, г. Чернигов, 14027, Украина).

**Radchenko Andrii** – PhD student, Chernihiv National University of Technology (95 Shevchenka Str., 14027 Chernihiv, Ukraine).

**E-mail:** [teor292@gmail.com](mailto:teor292@gmail.com)