

ІНФОРМАЦІЙНИЙ АСПЕКТ: РОЗПІЗНАВАННЯ ОБРАЗІВ ІНДИВІДУУМА

Шеховцов А.В.

Розпізнавання образів є однією з найбільш фундаментальних проблем теорії інтелектуальних систем. З іншого боку, задача розпізнавання образів має величезне практичне значення. Замість терміну "розпізнавання" часто використовується інший термін - "класифікація". Ці два терміни у багатьох випадках розглядаються як синоніми, але не є повністю взаємозамінюваними. Кожний з цих термінів має свої сфери застосування, і інтерпретація обох термінів часто залежить від специфіки конкретної задачі.

Віднесення об'єкту до того чи іншого класу. Це може бути, наприклад, задача розпізнавання літер або прийняття рішення про наявність дефекту у деякій технічній деталі. Віднесення об'єкта до певного класу відображає найбільш типову проблему класифікації, і, коли говорять про розпізнавання образів, найчастіше мають на увазі саме цю проблему. Саме її буде розглянуто в першу чергу в даній роботі.

Дана стаття присвячена проблемі технічного зору і розпізнавання образів мобільним роботом. Упор робиться не на високу науковість, а на практичне застосування, тобто все описане нижче працює на реальному роботі.

Постановка задачі: Мобільний робот по ходу руху повинний розпізнавати образи, розташовані на столі, за допомогою відеокамери. З малим числом образів задача повинна вирішуватися в реальному часі. Алгоритм розпізнавання повинний бути інваріантний щодо розміру образів і їхнього положення (повороту). Усе працює під ОС Windows. [1]

Алгоритми розпізнавання образів. Теорія технічного зору існує не перший день, по цьому в літературі можна знайти досить підходів і рішень. Для початку перелічимо деякі з них:

- **Алгоритм скелетизації.** Коротенько, це якийсь метод розпізнавання одинарних бінарних образів, заснований на побудову кістяків цих образів і виділення з кістяків ребер і вузлів. Далі по співвідношенню ребер, їхньому числу і числу вузлів будується таблиця відповідності образам. Так, наприклад, кістяком кола буде один вузол, кістяком букви П - три ребра і два вузли, причому ребра відносяться як 2:2:1. У програмуванні даний метод має кілька можливих реалізацій.[5]

- **Нейросітєві структури.** Напрямок був дуже модним у 60-70 роки, у наслідку інтерес до них небагато зменшився, тому що солідне число нейронів вимагає солідні обчислювальні потужності, що звичайно відсутні на простеньких мобільних платформах. Однак треба мати на увазі, що нейросітки іноді дають досить цікаві результати, за рахунок своєї нелінійної структури, більш того деякі нейросітки здатні розпізнавати образи інваріантні щодо повороту без якої або зовнішньої пред обробки. Так, наприклад мережі на основі неокогнейтронів здатні виділяти деякі характерні риси образів, і розпізнавати їхній як би образи не були повернені.[6]

- **Інваріантні числа.** З геометрії образів можна виділити деякі числа, інваріантні щодо розміру і повороту образів, далі можна скласти таблицю відповідності цих чисел конкретному образу (майже як в алгоритмі скелетезації). Приклади інваріантних чисел: число Ейлера, ексцентриситет, орієнтація (розташування головної осі інерції щодо чогонбудь теж інваріантного).[7]

- **Поточечне процентне порівняння з еталоном.** Тут повинна бути деяка пред обробка, для одержання інваріантності щодо розміру і положення, потім здійснюється порівняння з заготовленою базою еталонів зображень - якщо збіг більше чим якась оцінка, то вважаємо образ розпізнаним.[7]

Практична частина розпізнавання образів. Проаналізувавши всі алгоритми розпізнавання, описані вище, прийшли до висновку, що: необхідно працювати з растровим зображенням (так швидше одержати реальні результати). При цьому вводяться наступні обмеження: наявність відблисків на зображенні зрушує центр мас і орієнтацію, так само не припустимі розриви в об'єкті. Однак запропонований метод володіє і плюсами (дрібні помилки повороту, неточності бінаризації і т.п. будуть природним образом згладжуватися на стадії стиску в матрицю-іконку). Запропонований метод виконується в наступній послідовності:

- зображення від камери;
- бінаризація;[5]
- сегментація;[5]
- вихоплювання, обчислення ознак;[6]
- поворот, повторне вихоплювання;[6]
- стиск матриці;
- розпізнавання матриці нейросіткою.[8]

Рішення проблеми. Одержання зображення з камери в ОС Windows. В ОС Windows усі джерела відео мають один шаблон. Робота з відео проводиться за допомогою DScap. Відеокамера, що може давати відео потік (послідовність Bitmap) далі відео потік потрібно настроїти, для цього запросити драйвер про всі можливі діалоги, що він (драйвер пристрою) може дати. І власне за події захоплення кадру (драйвером, якщо завгодно) буде викликатися функція користувача, на вхід йде хто неї викликав і ще параметр, власне захоплений Bitmap у зазначеній палітрі і з зазначеною яскравістю і т.п., за допомогою Direct. Скачати демо для роботи з відео в Windows під Delphi не представляє особливої праці.

Бінаризація. У розробленій програмі розпізнаються тільки бінарні образи, тому другим етапом після одержання картинки, формуються бінарні образи. При роботі з кольоровою камерою перетворення з кольору в чорно, білий колір йде по стандартній формулі $Y = 0.3 * R + 0.59 * G + 0.11 * B$. Далі алгоритм досить простий: є деяка планка, якщо колір відтінку сірого вище - він вважається білим, якщо нижче - вважається чорним. Як видно бінаризація дуже проста, однак для серйозного поліпшення якості роботи розпізнавання, і зменшення часу роботи наступних модулів, на цьому місці краще ввести якийсь фільтр, нехай навіть найпростіший. Для роботи з відео одним з найпростіших фільтрів є фільтр по контрастності. У програмі не використана така конструкція, однак місце де вона може бути включена, позначено, і на тім місці знаходиться програма, що відслідковує кількість пікселів (білих або чорних) ідучи підряд, і виключаючи виникнення послідовності в ряді: 01010101. У режимі DEBUG місця зміни кольору, зафіксовані програмою можна побачити по червоному обрамленню образів, що малюється безпосередньо в модулі бінаризації. Стандартні модулі DELPHI опитування кольору пікселів у bitmap – надзвичайно повільно працюють. Для прискорення процесу використані модулі Qpixels. Після однократного опитування кольорів і їх бінаризації, програма працює тільки зі звичайною бінарною матрицею (динамічним двовимірним масивом), тому далі процес йде швидко.

Сегментація. Всі описані вище алгоритми розпізнавання образів працюють з єдиним видимим образом, у реальній ситуації відеокамера може бачити відразу кілька об'єктів, спеціально розташованих поруч, або ж у полі зору може потрапити який-небудь сторонній об'єкт. Якщо не передбачати деяку розбивку загального зображення на частині, то жоден з описаних вище алгоритмів не зможе коректно працювати. Далі виробляється розбивка зображення на частині, кожна частина з яких містить свій унікальний об'єкт, названий сегментацією. І в розпізнаванні образів, і в сегментації існує досить багато алгоритмів, кожний з яких має свої достоїнства, у пропонованій статті алгоритм працює тільки з чорно-білим зображенням. У сегментації чітко розділяються чорно-білі зображення на бінарні кольори і з відтінками сірого. Тут працюють зовсім різні по швидкості і складності

000

У разцінці житті матриця має розмір дорівнює рядкові до 640 елементів 0 збо 1

І і М це мітки після проходження алгоритму по масиву (однократного) повинні

00

$$\begin{array}{ccc} 0 & 0 & 0 \\ 0 & 1 \longrightarrow & 0 \quad L \end{array} \quad \begin{array}{ccc} 0 & 0 & 0 \\ L & 1 \longrightarrow & L \quad L \end{array}$$

2. 11. 1988

Вихоплювання образу обчислення деяких інваріантних чисел. Після завершення

ра, ексцентриситет і орієнтація. Як уже було сказано, можна зібрати таблицю 8-ми інваріантних ознак і розпізнавати образи виходячи тільки з цих даних, однак ми підемо другим шляхом. З усієї безлічі інваріантних чисел обчислимо тільки одну орієнтацію, для цього використовуємо алгоритми, схожі на MATLAB (можливо вони там і використовуються).

При обчисленні ряду морфометричних ознак використовуються поняття механіки твердого тіла. Зокрема, це відноситься до довжин осей інерції об'єкта. Напряму в тілі, що збігаються з півсями еліпсоїда інерції, називають головними осями інерції. Для перебування головних осей інерції, що лежать у площині об'єкта, у функції *imfeature* використовуються наступні співвідношення [1, 2, 5]. Нехай N - кількість пікселів, що відносяться до об'єкта. Уся безліч пікселів $p(x, y)$, що відносяться до об'єкта, позначимо Q . Тоді координати центра мас об'єкта обчислюються як:

$$x_c = \frac{1}{N} \sum_{p(x,y) \in \Omega} x, \quad y_c = \frac{1}{N} \sum_{p(x,y) \in \Omega} y$$

Обчислимо кілька допоміжних величин:

$$U_x = \frac{1}{12} + \frac{1}{N} \sum_{p(x,y) \in \Omega} (x - x_c)^2;$$

$$U_y = \frac{1}{12} + \frac{1}{N} \sum_{p(x,y) \in \Omega} (y - y_c)^2;$$

$$C = \sqrt{(U_x - U_y)^2 + 4 \cdot U_{xy}^2}$$

Тоді довжини максимальної A_{\max} і мінімальної A_{\min} осей інерції обчислюються як:

$$A_{\max} = 2\sqrt{2} \cdot \sqrt{U_x + U_y + C};$$

$$A_{\min} = 2\sqrt{2} \cdot \sqrt{U_x + U_y - C}$$

Довжини головних осей інерції використовуються для обчислення ексцентриситету й орієнтації об'єкта. Ексцентриситет визначається за допомогою співвідношення

$$E = \frac{2 \cdot \sqrt{(0.5 \cdot A_{\max})^2 - (0.5 \cdot A_{\min})^2}}{A_{\max}}$$

Орієнтація визначається як кут у градусах між максимальною віссю інерції і віссю X . Якщо $U_y > U_x$, то орієнтація O обчислюється за допомогою формули

$$O = \frac{180}{\pi} \cdot \arctg \left(\frac{U_y - U_x + C}{2 \cdot U_{xy}} \right),$$

у протилежному випадку O обчислюється як

$$O = \frac{180}{\pi} \cdot \arctg \left(\frac{2 \cdot U_{xy}}{U_x - U_y + C} \right)$$

Поворот образу, повторне вихоплювання. Знайдено орієнтацію зображення, що унікальна для кожного образу. Зроблено це для того, що б тепер образ можна було повернути щодо центра мас, так що б його орієнтація була рівнобіжна осі X . Власне цей прийом і дає інваріантність щодо початкового повороту. Ну і звичайно деяка демонстрація роботи алгоритму обчислення орієнтації і повороту бінарного образу. Орієнтація це ж не вектор, а тільки пряма, т.ч. образ можна повернути і ліворуч і праворуч. Насправді це навіть і не проблема:

- в перших можна навчати нейросітки на двох прикладах образу - покладеного поворотом праворуч і поворотом ліворуч. Це не цілком коректно, але на простих речах працює без утрати якості;
- по-друге, можна створити додатковий тип для кожного образу - там він буде ле-

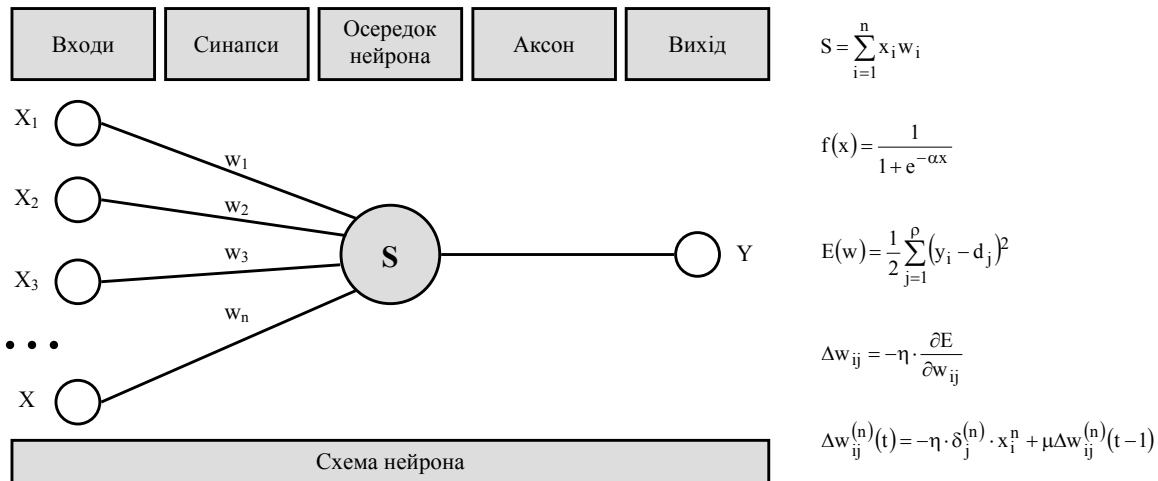
жати розгорнутий у зворотну сторону.

Стиск образу в матрицю заданого розміру. В оболонці програми знаходиться якась бінарна матриця, у яку ужимається кожен окремий сегментований образ. Розмір матриці задається по необхідності - тобто, якщо потрібна велика деталізація, то краще використувати матриці більшого розміру, ніж у поточній версії програми. Це дасть додаткові обчислення на стадії розпізнавання, що і підвищить якість процесу. Однак важливо розуміти, що малий дозвіл розпізнаваної матриці дозволяє виправити деякі можливі помилки при повороті зображення (обчислення орієнтації образу), так як незначне відхилення в 5 градусів, не буде помітно після стиску образу. На практиці такі помилки обов'язково будуть промайнати, коли якусь частину образу закрийє перешкода або відблиск - центр мас зміщається, орієнтація можливо теж - поворот буде не цілком коректний. У реальних умовах при розпізнаванні, образ ніколи не буде видний камерою як повинний бути на 100%, так як в русі хоча б одна точка образу буде гарантовано викривлена камерою, тому, чим менше розпізнавана матриця (тобто чим грубіше стиск до її розмірів), тим більше можливих помилок буде не замічено, але тем менше рівень можливої деталізації. Тут необхідно пояснити, що для кожної задачі розмір матриці треба вибирати відповідний і не завжди чим більше, тим краще.

Розпізнавання образу нейросіткою [7]. Для перевірки пред обробітки, і всіх описаних вище алгоритмів, як метод розпізнавання був втілений алгоритм процентного порівняння з еталонами. Спочатку програма фотографувала і провела предобробку тими ж алгоритмами тренувальні образи, після чого, коли було потрібно розпізнавати щось нове, вона це нове знову ж таки проводила предобробку до матриці заданого розміру, і потім цю матрицю порівнювала з усім запам'ятованими матрицями.

Структура нейронної мережі

Багатошарова мережа, зворотного поширення похибки. 256-6



Багатошарова нейросітка зворотного поширення помилки, займається тим же, що і процентний алгоритм порівняння матриці, однак за рахунок своєї нелінійної структури розпізнає на 10-30% краще порівняння двох матриць [6]. Для перевірки були створені дві сітки - одна 256-6, друга 256-40-6. Перший шар у 256 нейронів це вхід матриці 16x16, останній шар це вхід - розпізнаємо, приміром, 6 букв. На практиці, нейросітка із трьох шарів (з якоюсь внутрішньою обробкою - у 40 нейронів) навчалася з великим трудом, постійно ловлячи локальні мінімуми (висновок зроблений з поводження середньоквадратичної помилки, виведеної в реальному часі при навчанні). Якість же розпізнавання візуально не зросло. Було прийняте рішення зупинитися на простій структурі в 256-X (де X - число заучених образів), що навчається досить нетривалий час, поводить себе стабільно і показує гарні результати. За результат розпізнавання був узятий вихід нейрона, більший 1-K, причому жоден з інших виходів не може бути більше K. У задачі K береться рівне 0.2, тобто можна провести деяку

аналогію з процентним порівнянням (за істину береться більше 80% збігів), з тією лише різницею, що нейросітки "порівнює" нелінійно.

Висновок. Отриманий метод розпізнавання не є універсальним, але отримані результати охопили досить великий обсяг задач і відповідають поставленій задачі. Виконуючи досить вузьку проблему вдалось застосувати задачу по ідентифікації фото зображень в пошуках заданого наперед критерію.

The method of recognition of a photo of images is described. Carrying out rather narrow problem it was possible to apply a problem in identification of a photo of images in searches set on before criterion.

1. Головкин В.А. Нейроинтеллект: Теория и применения. Книга 1. Организация и обучение нейронных сетей с прямыми и обратными связями – Брест:БПИ, 1999, - 260с.
2. Головкин В.А. Нейроинтеллект: Теория и применения. Книга 2. Самоорганизация, отказоустойчивость и применение нейронных сетей – Брест:БПИ, 1999, - 228с.
3. Бардачов Ю.М., Ходаков В.Є., Шеховцов А.В., Бараненко Р.В. Аналіз створення та підходи реалізації автоматизованої системи «Реєстр виборців України» // Вісник Херсонського національного технічного університету, 2005 р. – С. 153-167.
4. Бардачов Ю.М., Ходаков В.Є., Шеховцов А.В., Шаганян С.М., Бараненко Р.В. «Вибори» - автоматизоване робоче місце працівників дільничної виборчої комісії // Вісник Херсонського національного технічного університету, 2006 р. – С. 10-15.
5. Уоссермен Ф. Нейрокомпьютерная техника: Теория и практика, 1992 – 184с.
6. Petrou M. Learning in Pattern Recognition. Lecture Notes in Artificial Intelligence – Machine Learning and Data Mining in Pattern Recognition, 1999, pp. 1-12.
7. Jacobsen X., Zscherpel U. and Perner P. A Comparison between Neural Networks and Decision Trees. Lecture Notes in Artificial Intelligence – Machine Learning and Data Mining in Pattern Recognition, 1999, pp. 144-158.
8. Aizenberg I. N., Aizenberg N. N. and Krivosheev G.A. Multi-valued and Universal Binary Neurons: Learning Algorithms, Applications to Image Processing and Recognition. Lecture Notes in Artificial Intelligence – Machine Learning and Data Mining in Pattern Recognition, 1999, pp. 21-35.
9. Yoon K. S., Ham Y. K. and Park R.-H. Hybrid approaches to frontal view face recognition using the Hidden Markov Model and Neural Network. Pattern Recognition 1998 Vol. 31, pp. 283-293.

УДК 677.05.059:621.3.078.08

КОНТРОЛЬ КАЧЕСТВА ТКАНЕЙ СПЕЦИАЛЬНОГО НАЗНАЧЕНИЯ С ПОМОЩЬЮ АВТОМАТИЧЕСКИХ СИСТЕМ

Терновая Т.И., Сумская О.П., Слободянюк И.И., Булка Т.И.

Введение. В настоящее время уровень исследований и разработок в области технологии текстильных материалов для чистых производственных помещений (ЧПП) различных классов справедливо считается свидетельством достижений науки и техники государства. Потребителями текстильных материалов для ЧПП является микроэлектронная, полупроводниковая, химико-фармацевтическая промышленности, медицинские учреждения и пищевые предприятия.

Основная функция технологической одежды состоит в защите технологической среды и производственного продукта от загрязнений, которые выделяет человек, и, исходя