Реклами, реклами, … – намиране без взиране

Мирослав Иванов[1], Красимира Иванова[1], Илия Митов[1], Евгения Великова[2]

1. Институт по математика и информатика при БАН, София
mivanov@math.bas.bg, kivanova@math.bas.bg, imitov@math.bas.bg
2. ФМИ при Софийски университет „Св. Климент Охридски“, София
velikova@fmi.uni-sofia.bg

Резюме: Обект на изследването са възможностите за облекчаване на търсенето на повтарящи се реклами в пресата при осъществяването на медия мониторинг. Анализира се възможността за бързо намиране на повторенията чрез използване на дескриптор на изображенията Average Hash (aHash). Предлага се оригинален подход за организация на етикетираните екземпляри и намерените повторения в една обща база от данни.

Ключови думи: image authentication, aHash, multimedia data base, ArM32.

ACM 1998 Classification Keywords: H.5.1 Multimedia Information Systems

Въведение

В наши дни информационните и клипинг агенции добиват особена значимост при анализите на социалното присъствие на институциите и фирмите. Чрез обработване на постъпващата информация от централните и регионални електронни и печатни медии, те предоставят на своите клиенти задълбочени анализи от вида [1]:

- оценка на медийния отзвук на кампания или събитие. Този вид анализ показва коя информация е предизвикала интерес в медиите, кои комуникационни и PR дейности са били успешни. Анализът служи за оценка, оптимизиране и оформяне на бъдещите отношения между медиите и PR дейността;
- Benchmark анализ. Служи за сравняване на медийното отразяване на две или повече компании и е подходящ за проучване на медийното присъствие на конкуренти и партньори. Той показва медийното позициониране на една компания и може да служи за определяне на степента на постигане на комуникационните цели;
- изчисляване на рекламната стойност на реклами и PR публикации. Той може да се използва за остойностяване на публикувани PR материали на базата на рекламните тарифи на медиите, както и да се изчисли стойността на дадена рекламна кампания. Рекламният анализ е приложим за изчисляване на собствени материали, както и за материалите на конкуренти и партньори.

Един от основните изходни материали, предоставян на клиентите на такива компании, е бюлетинът, който съдържа както обобщени количествени данни,

оценки и анализи, така и преки референции към материалите, съдържащи името на клиента или следените от него конкуренти.

Голяма част от материалите, които са във фокуса на разглежданията, са отпечатаните реклами. Характерно е, че обикновено в рамките на период от време компанията или институцията използва една и съща реклама. В момента търсенето на срещанията на рекламите и класифицирането им по видове се осъществява ръчно от оператори, което е времеемка и уморителна работа. Цялостният процес започва със сканирането на страниците на печатните материали, след което се извършва „нарязване“ и съхраняване на отделните части в различни контейнери. Текстовите участъци се подлагат на OCR обработка, ръчно почистване и последващо категоризиране по набор от ключови думи, а изображенията се насочват към специализиран преглед за наличието на реклами на следените институции, брендове или кампании. На пазара съществуват софтуерни продукти, които се занимават с управлението на колекции от изображения, но само като пример ще посочим, че продуктът MatchEngine на канадската фирма TinEye започва от $200 месечно за ограничен обем от изображения и стига до $1500 (и нагоре) за корпоративни клиенти [2].

В настоящата работа се предлага един подход за организиране на процеса на преглеждане на изображенията за наличие на търсени реклами, който е полуавтоматизиран. Постъпващите изображения се сравняват с налични в базата етикетирани първообрази и при достатъчна близост те директно се категоризират в същата група. При срещане на изображение, което системата не може да причисли към някое от етикетираните, се извършва ръчно категоризиране от оператора или отхвърлянето му като нерекламен материал.

По-долу организацията е както следва: В първа точка се разглеждат фамилия дескриптори, използвани като „отпечатъци“ (fingerprint) на изображенията, като се прави оценка на приложимостта им в решаването на тази задача. Втора точка съдържа експериментални данни и анализи на избора на конкретния дескриптор и оптималните параметри за използването му. Трета точка се спира на едно оригинално предложение за организиране на базата от данни, при което етикетираните първообрази и последващо намерените близки до тях изображения се съхраняват в обща база. Заключението обобщава получените резултати и предлага насоки за бъдещо развитие.

1. Фамилията Hash дескриптори, използвани като отпечатък на изображения

В практиката има редица приложения, където се получава размножаване на цифрово изображение чрез преоразмеряване, разтегляне, промяна в контрастите и форматите. Като резултат се появяват нови, подобни на първоизточника, изображения. Въпреки че изображенията не са идентични, те продължават да бъдат доста близки. И често се налага решаването на обратната задача – да се намерят близките изображения, които вероятно са трансформации от един и същ първоизточник. Повечето от изследванията са върху търсенето на сродни изображения, които са претърпели различни

цифрови трансформации [3], други изследвания са фокусирани върху проблемите на разпознаване на близост на цифров първообраз, който впоследствие е преминал през печатане и обратно сканиране [4]. Ние в случая се интересуваме от задачата за намиране на близостта на изображения, които са сканирани от различни печатни източници.

Един от предлаганите методи използва хеш функции, построявани на базата на съдържанието на изображенията. За разлика от криптографските методи (като MD5 и SHA1), при които след построяването и сравнението на стойностите на две хеш функции може да се каже само, че ако хешовете са различни, то данните са различни, а ако хешовете са еднакви, то може би данните са еднакви (поради възможността за наличие на колизии), фамилията разглеждани функции дават възможност за сравняване и търсене на близост между изображенията, чиито хеш функции се сравняват. Детайлни описания на алгоритмите са представени в [5,6]. Специално за дескриптора pHash има библиотека с отворен достъп на фирмата Aetilius Inc. (pHash е регистрирана марка от Evan Klinger) [7].

В разглежданата фамилия от хеш функции спадат **Average Hash** (**aHash**), **Difference Hash** (**dHash**) и **Perceive Hash** (**pHash**). **aHash** генерира отпечатък на изображението спрямо средната светлота на изображението. **dHash** следва подобен алгоритъм, но генерира стойността на поредния пиксел в зависимост от това дали е по-ярък от левия си съсед, т.е. той стъпва на оценка на градиентите. **pHash** е на пръв поглед най-финият метод, но за сметка на това и най-бавният. Той се основава на оценка на честотните модели чрез прилагане на дискретна косинусова трансформация на матрицата.

И в трите случая първоначално се извършва редуциране на размера и цвета на изображенията.

Редуцирането на размера до определени ширина l и височина h (в повечето случаи равни) е най-бързият начин за премахване на детайлизацията на изображенията, игнориране на оригиналните размери и съотношение. Всички резултатни изображения стават с едни и същи $l \times h$ пиксела.

Следващата стъпка е **редуцирането на цвета**. При нея полученото изображение се преобразува в сивата гама (grayscale). За тази цел използваме по-финото преобразуване чрез формулата, по която се намира luminosity в цветовия модел YCbCr [8]: $Y = 0.299 * r + 0.587 * g + 0.114 * b$

От получените еднакви по размер обезцветени изображения се изчисляват съответните дескриптори по следния начин:

- Average Hash: За всяко от изображенията се намира средната стойност на сивото в изображението. Дескрипторът aHash представлява поредица от нули и единици с дължина $l \times h$, определящи дали поредният пиксел от линейното разгъване на сивото изображение е по-светъл (стойност „1“) или по-тъмен (стойност „0“) от изчислената средна стойност на сивото за даденото изображение (на фиг.1 са визуализирани стъпките, през които преминава изображението до получаване на дескриптора aHash);

– Difference Hash: Дескрипторът dHash се получава чрез линейно разгъване на матрицата с дължина $l \times h$, чиито елементи имат стойност единица, ако левият пиксел на трансформираното изображение е по-тъмен от текущия и нула – в противен случай. За получаване на стойността на най-лявата колона сравнението се прави между пикселите от последната колона и текущите;
– Perceive Hash: Алгоритъмът стъпва на намиране на отклоненията на честотите спрямо средната DCT честота, което позволява запазване на стабилността му при корекции на изображението като силни промени в цвета и осветеността, които продължават да запазват общата му структура. Изчислението обаче протича през няколко фази на определяне на DCT коефициенти и последващо редуциране, което прави алгоритъма в пъти по-бавен от останалите [5,6].

(а) Първоначално изображение

(б) Редуциране до определени ширина и височина (в случая 16x16)

(в) Преминаване към grayscale представяне

(г) Преобразуване на пикселите в черно/бяло спрямо средната стойност на сивото

aHash = (1111111111111111 1000111000000011 1111111111111111
1111010011111111 …
… 1000000000011111 0000000001111111 0000000011111011
0000000111111111)

Фигура 1. Визуализация на стъпките по получаване на вектора aHash

Оценка на близостта на изображенията

Получените вектори (независимо по кой от методите – aHash, dHash или pHash) могат да се използват като дескриптори на изображенията. Близостта между два дескриптора $p = (p_1, ..., p_{l \times h})$ и $p = (p_1, ..., p_{l \times h})$ съответства на степента на близост между представените с тези дескриптори изображения.

Сравнението става на базата на оценка на Хеминговото разстояние между двата дескриптора:

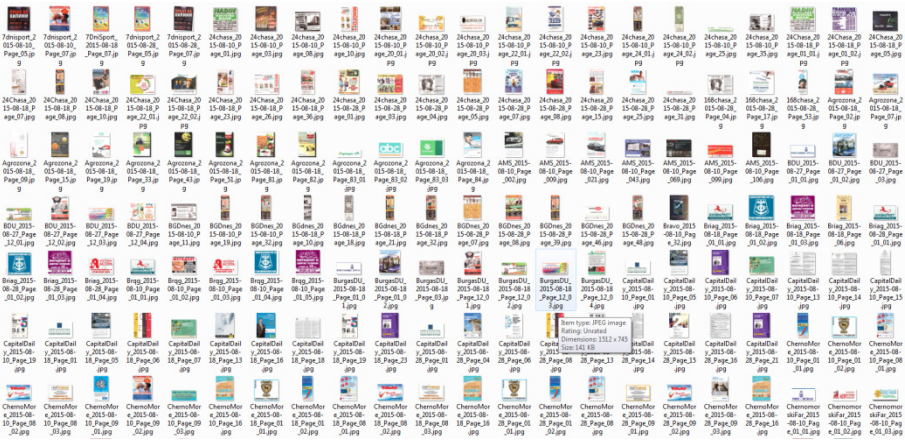
$$d(p,q) = \sum_{i=1}^{l*h} (p_i - q_i)$$

Практическата задача предполага автоматизирана обработка на изображенията поради честата поява на изображения, които са нови реклами и от една страна следва да бъдат етикетирани от оператора, а от друга възниква небходимостта той лично да прегледа верността на класифицираните екземляри. Затова в този случай, алгоритъм, който бързо изчислява избрания дескриптор, а след това позволява достатъчно силно стесняване на потенциалните кандидати за съвпадение с постъпващите изображения, е достатъчно приемлив. По тази причина по-нататъшните експерименти ще проведем с първите два дескриптора – aHash и dHash.

2. Резултати от експеримента

2.1. Тестово множество

Тестовото множество се състоеше от 900 изображения, които представляват изображенията на потенциални реклами, изрязани от около 40 издания от три последователни дни (фиг.2). Имената на самите файлове носят информация за изданието, деня и страницата, където са били, както и пореден номер (в случай че има няколко реклами от една и съща страница). Част от изображенията са текстови съобщения, съдържащи информации за обявяване на търгове и други подобни. Те следва да се класифицират отделно, но също подлежат на обработка.



Фигура 2. Отрязък от изображенията от тестовото множество

2.2. Инструментариум

За целта беше изградена експериментална система, която позволява изчисляването на съответния дескриптор с различни размери на трансформираното изображение (задават се параметрично), изчисляване на разстоянията между изображенията и на тази база последващ анализ на процентите на различие между изображенията, които да се вземат като ограничител при показването на евентуалните кандидати.

Фигура 3 показва резултати от визуализацията на двата дескриптора при различни размери за една от рекламите, съдържащи се в тестовото множество.

Фигура 3. Визуализация на двата дескриптора при размери 8x8 и 24x24 (бяло=„1“, черно=„0“)

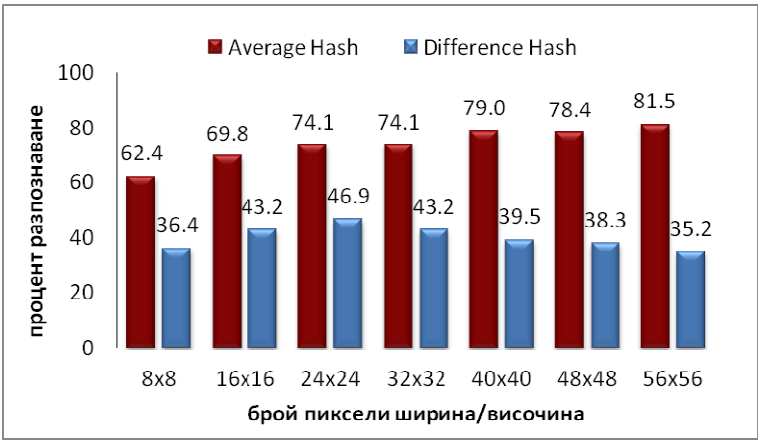
Проведените експерименти за оценка на точността при класификация бяха направени с използването на класификатора LB1 от Weka [9], който се базира на оценка на различията между отделните атрибути, което в случая съответсва на Хеминговото разстояние между дескрипторите.

2.3. Експерименти и анализ

Първата група експерименти бяха проведени с цел изследване на точността на разпознаване при използване на aHash и dHash за различни размери.

В експеримента участваха общо 260 етикетирани изображения, като 98 от тях бяха в обучаващата извадка, а останалите 162 – в тестовата извадка (практическата задача произвежда голямо количество класове с немного на брой представители).

Резултатите показаха, че за разлика от твърдението в [6], че dHash дава по-добри резултати от aHash, в конкретния случай със сканираните изображения dHash показа лоши резултати – под 50% разпознаваемост, за разлика от aHash, който даже и при малките размерности дава добра разпознаваемост на класовете (фиг. 4).



Фигура 4. Резултати от разпознаването при прилагане на aHash и dHash

Във втората група експерименти бяха включени допълнителни 50 текстови съобщения (обяви, които също се третират като рекламни съобщения) и още 590 неетикетирани изображения (нови реклами).

При класифицирането с aHash (в случая беше използвана 10-стъпкова крос валидация при размер 40x40) коректно бяха класифицирани 73.44 %, което за целите на намаляване на извадката за операторски избор е достатъчно добър резултат.

Допълнителен анализ (със собствени софтуерни средства) показа, че при неразпознатите класове изображението с верен клас се намира в първите шест най-близки класа в 43.2 % от случаите (границата 6 в случая е избрана предвид психологическата възможност за възприятие на човека до седем обекта).

3. Програмен проект и организация на базата от данни

Анализът на резултатите от проведените по-горе експерименти доведе до избор на използването на aHash дескриптора като оптимален спрямо получената точност и времето за изчисление. Размерността, до която да се свиват изображенията, беше оставена като параметър, който операторът да избира с цел последващо натрупване на информация относно качеството на класифициране на изображенията. Друг такъв параметър е броят етикетирани изображения, близки до постъпващото ново изображение, от което операторът да потвърждава класа му или да поставя нов етикет.

Самият алгоритъм за изчисляване на aHash дескрипторите на постъпващите изображения беше оптимизиран посредством прилагане на многонишкова обработка на изображенията, чрез която значително се повишава скоростта на работа. Например, при проведен тест на компютър с процесор с 8 логически ядра (4 физически, всяко от тях с по 2 нишки) времето за обработка намаля около 3 пъти.

Съхраняването на данните за изображенията съдържа както информация за източника на изображението (издание, дата, страница, място на съхранение на отрязъка), така и информация за това как е класифицирано изображението. От гледна точка на разпознаването на класа има два основни типа изображения (данни за тях). Единият тип са т. нар. *първообрази* – изображението се появява за пръв път в базата и тогава операторът ръчно му задава клас, към който принадлежи. Като първообраз може да попадне и изображение, каквото вече е имало в базата, но разпознаването му е било твърде лошо и не е било предложено сред най-близките класове. От гледна точка на задачата това няма значение, освен че операторът трябва да го етикетира още веднъж. Вторият тип са т. нар. *повторения* – системата е предложила първообразите на най-близките класове спрямо дескриптора на постъпващото изображение, от които операторът е посочил първообраза, към чиито клас принадлежи постъпващото изображение, или потвърдил избора на системата като първи най-близък клас.

При организацията на базата се предложи оригинален подход, при който данните за първообразите и последващите повторения се съхраняват в обща база, като първообразите съхраняват векторите, с които се сравняват новите изображения, и таговете, въведени от оператора при първото срещане, а повторенията само сочат към идентификаторите на съответните първообрази. За целта се използва ArM32 като многомерен метод, позволяващ компактното съхраняване на разнородна информация в обща база.

Заключение

Проведените експерименти показаха, че прилагането на разглеждания дескриптор aHash успешно може да се приложи и в разглеждания случай на разпознаване на сканирани от различни източници изображения. Експериментална софтуерна система, която да обедини отделните стъпки и да позволи последващ анализ на резултатите, е вече поектирана и в начален етап на разработка. Задачи, които екипът си поставя са бъдещо разрешаване, са:

- намиране на подходящ индексен механизъм с цел оптимизиране на търсенето по близост;
- изследване на възможностите за допълнително ускорение чрез търсене първо в същите издания, клъстера от издания и накрая в останалите;
- оценка на времевия интервал, в който средно се задържа дадена реклама, с цел оптимизиране на търсенето.

Литература

1. Медиа анализ, http://mediazoom.bg/
2. TinEye – MatchEngine, http://www.tineye.com/
3. Miller M.L., I.J. Cox, J.-P.M.G. Linnartz, T. Kalker: A Review of Watermarking Principles and Practices. Marcell Dekker Inc., New York (1999) pp. 461–485 (Chapter 18)
4. Wua D., X. Zhoub, X. Niuc: A novel image hash algorithm resistant to print–scan. Signal Processing J. Special Section: Visual Information Analysis for Security. Volume 89, Issue 12, Dec. 2009, pp.2415-2424.

5. Krawetz N., http://www.hackerfactor.com/blog/?/archives/432-Looks-Like-It.html
6. Krawetz N., http://www.hackerfactor.com/blog/?/archives/529-Kind-of-Like-That.html
7. pHash – The open source perceptual hash library, http://www.phash.org/
8. JPEG File Interchange Format, Version 1.02, http://www.w3.org/Graphics/JPEG/jfif3.pdf
9. Weka Data Mining Software, http://www.cs.waikato.ac.nz/ml/weka/index.html

Adverts, adverts,… – finding without staring

Miroslav Ivanov, Krassimira Ivanova,
Iliya Mitov, Evgenia Velikova

Abstract: The goal of this research is to examine the opportunities to facilitate the discovery of equal advertisements extracted from the press, which is one of the goals of media monitoring. The possibility to find duplicates quickly using the descriptor Average Hash (aHash) is analysed. Original approach for organization of the metadata keeping labeled specimens and found repetitions in a common database is proposed also.