Cash scanner

Проект по изкуствен интелект - документация

|  |  |
| --- | --- |
| *Румен Палетов* | *Фн. 80555* |
| *Диан Николов* | *Фн. 80507* |
| *Александър Панайотов* | *Фн. 80614* |

# Съдържание

[Съдържание 2](#_Toc378260276)

[1. Въведение 3](#_Toc378260277)

[2. Откриване на ръб 3](#_Toc378260278)

[2.1. Sobel 3](#_Toc378260279)

[2.2. Canny edge detector 3](#_Toc378260280)

[2.3. Сравнение между двата алгоритъма 4](#_Toc378260281)

[3. Откриване на окръжност 4](#_Toc378260282)

[3.1. Template –matching 4](#_Toc378260283)

[3.2. Blob detection 4](#_Toc378260284)

[3.3. Hough трансформация 4](#_Toc378260285)

[3.3.1. Откриване на линия 5](#_Toc378260286)

[3.3.2. Откриване на кръг 5](#_Toc378260287)

[4. Текстон 5](#_Toc378260288)

[4.1. Създаване на Текстонен речник 6](#_Toc378260289)

[4.1.1. Филтърни отговори 6](#_Toc378260290)

[4.1.2. K-means Clustering 6](#_Toc378260291)

[4.2. Моделиране на статистическа хистограма 6](#_Toc378260292)

[4.3. Дискусия около класифицирането на текстони 7](#_Toc378260293)

[Библиография 7](#_Toc378260294)

# Въведение

Нашият проблем е разпознаване на обекти , по-специално , разпознаване на монети в естествени изображения. В даденото естествено входно изображение, може да има монети или само кръгове. Нашата задача е откриването само на монети във входното изображение. Съществуват няколко подхода за постигане на тази цел. Използването на известни техники за откриване на ръба може да бъде един възможен начин; Hough трансформиране е друг също би свършил работа за изпълнение на нашата цел. Тези техники, обаче, засичат само информацията за границите на целевите обекти, така че ние имаме нужда от фундаментално различен метод за откриване на монети в оригиналното изображение. Цветът може да бъде една от основните характеристики на монетите, но в този случай, ако даден образ е полутонов, тогава цветът не може да бъде най- уникалното по което да разпознаем монета. Текстурите на повърхността на монетите може да бъдат уникална характеристика на монетите, освен това, текстон се отнася до основните микро - структури в генерични естествени изображения, както и основните елементи в началото на визуалното възприятие. Нашият проблем е съставен от три части, като първата фаза е етап на откриване, втората фаза е Hough трансформация, а последният етап е текстон класификация. Тези фази ще бъдат обсъдени по – нататък в документа.

# Откриване на ръб

## Sobel

Естествените изображения представляват поредица от пискели, с различни стойности за R, G и B. От гледна точка на опростяване на работата на Hough трансформацията, такова изображение не е подходящо за директна обработка. Трансофрамцията използва полутонови изображения, като за целта чрез всеки един пиксел определя линии и криви за правоъгълници и окръжности, затова готово изображение с определени ръбове ще е много полезно. В този пример операторът Sobel в OpenCV се използва за откриване на ръбове. Преди обаче, да изполваме тази операция, е необходимо да изпълним няколко предварителни стъпки. Първо необходимо е RGB изображение да се проемни в полу-тоново. Новото изображение се филтрира от Гаусова операция за редуциране на „шум“. Едва след това полученото изображение се подлага на Sobel операция. Тази операция е лесна за имплементиране но откритите линии са много дебели. За Hough трансформация ни е необходимо намерените линии да са единични. Затова дебелите линии трябва да бъдат „потиснати“. Начинът, по който да постигнем това е да изберем най-добре съвпадащата линия между стойностите на пикселите на съседите. Сега, входно изображение е готово за Hough трансформация.

## Canny edge detector

За да се приложи алгоритъмът, трябва да се следва поредица от стъпки. Откриване Първо - изглаждане на изображение с Гаусово филтриране. След изчисляване на величина и ориентация на градиента, използвайки крайна разлика приближения за частните производни. След това се прилага nonmaxima „потискане“ на величината на градиента. Nonmaximum потискане се използва, за да се проследи по ръба в посока на ръба и да потисне всяка пикселова стойност (като я равнява на 0), която не се счита за ръб. Това ще даде една тънка линия в изходния образ. Следващата стъпка е прилагане на double threshold алгоритъм за откриване и свързване ръбове. Изображението е готово за Hough Трансформация.

## Сравнение между двата алгоритъма

Използвайки метода на Sobel, ръбовете от изображения са твърде объркани и данните често загубват част от важна структура. Пикселът на снимките е шумен и ръбовете не са гладки и тънки. Ръбове са два или три пиксела дебели. В резултатните изображения често не се показва важна информация, тъй като повечето от нея е загубена. Въпреки това, в резултатите все още може да се покаже формата на оригиналното изображение.   
За Canny edge detector алгоритъмът, ръбовете се откриват гладъки и почти никакви шумни пиксели не се срещат, като в същото време се запазват важните структурни   
свойствата на оригиналния образ. Canny   
методът произвежда гладки и тънки ръбове, както и по-малко шумни пиксели   
в сравнение със Sobel. Canny методът използва много повече памет от Sobel, a пък Sobel е много по – бърз. Въпреки това качество на ръбовете получавано при Canny edge detector, бързината и по – малкото използвана памет е причината, Sobel да бъде избран за текущата разработка.

# Откриване на окръжност

## Template –matching

Template –matching е техника при която се откриват малки части от дадено изображение, което съвпада с друго шаблонно изображение. При този вид техника е необходимо търсене на огромно количество точки за да се определи най-добро съпвадение. Това прави техниката бавна.

## Blob detection

Blob detection е техника, която открива региони от дадено изображение, които се открояват по дадени характеристики (яркост, цвят…). В този смисъл всички точки в даден регион (също „блоб“) се считат за подобни една на друга.

## Hough трансформация

Hough трансформацията е един много полезен инструмент, за да се намери модел в едно изображение, като линии и криви. С превръщането на една точка в пространство с параметър, лесно се разпознават модели. Особено това е добре за шумни и рядко дигитализирани изображения. Първо, една точка в един образ, която е открита като ръб, прави крива в параметричното пространство, 2-D за линии и 3-D за кръгове. Всеки път, когато една точка от изображението се предели за валиден кандидат за една линия или кръг, се обновява крива в параметричното пространство.

### Откриване на линия

От изображение, Hough трансформацията намира ръбове, които правят прави линии, без да се използва свързани или близки крайни точки. Много точки се получават на границата, когато процес на откриване на ръб е завършен, и дава възможност да се откроят линиите от фалшивите такива в изображението. След това фигуралните очертанията са насложени върху оригиналното изображение. следното уравнение се използва за Hough transformaciq за линии, както и 2-D масив „акумулатор“ се създава за r и Q.

х cosQ + Y sinQ = R

Относителните големи върхове в масива акумулатор са определени като линия и обратно до оригиналното изображение.

### Откриване на кръг

Кривите се откриват по същия начин като линиите но с използването на различно параметризиране. Формулата е:

(x - a)^2 + ( y – b)^2 = r^2

Където a и b са център на окръжност, а r радиусът. Създава се 3-D масив акумулатор и относителните върхове се третират като кръг.

Възможността Hough трансформацията да бъде лесно адаптирана за намиране на окръжности е причината тази техника да бъде избрана за текущата разработка.

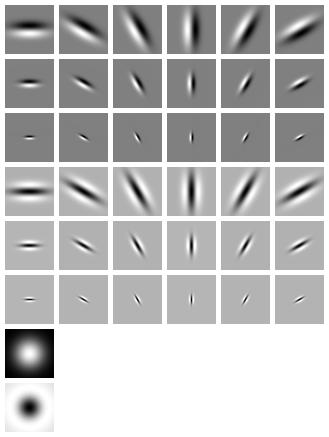


# Текстон

Текстоните се дефинират като фундаментални микро структури на естествените изображения. Всяко такова изображение се състои от текстони, а те са най – малкият елемент който хората могат да забележят. Затова ако могат да се намерят универсални текстони, които да могат да генерират всички текстури в естествения свят, като например азбуката на английски, тогава могат да се създадат много стабилни правила за сегментиране, класифициране и дори синтезиране на изображения. Leung и Malik дават оперативна дефиниция на текстон – клъстърен център в пространство отговарящо на филтър. Това не само позволява текстони да бъдат генерирани автоматично от изображения, но също създават възможността за универсално множество от текстони за всички изображения.

## Създаване на Текстонен речник

### Филтърни отговори

Първата стъпка е, че всички трениращи данни са определени да се конволюират с помощта на филтърна банка. 8 класа на монети са дадени в този проблем (1 ст. от пред/от зад, 5 ст. от пред/от зад, 10 ст.от пред/от зад и 50 ст. от пред/от зад монети), и всеки клас има 16 снимки с различни размери, вследствие на което общият брой на трениращите изображения е 128 изображения. Тук, 38 филтри се използват за да се конволюира дадени обучителни изображения.  
 

**Снимка 4.** Пример за матрици прилагани върху пиксели.

* 1 edge filter с по 6 ротации и 3 скалирания
* 1 bar filter с по 6 ротации и 3 скалирания
* 1 гаусов филтър
* 1 лапласов филтър

### K-means Clustering

Втората стъпка е като се използват дадени отговори, центровете на клъстерите да се намерят чрез използването на алгоритъм за векторно квантуване и в този проблем се използва K-means алгоритъм. Ако имаме размер на изображение 100 на 100 и имаме 8 филтър отговора, тогава общо 10000 8-вектора представляват едно изображение. Затова е скъпо и е необходимо да се редуцира репрезентацията на данни. Leung и Malik забелязали, че филтър отговорите не са напълно различни при всеки пиксел над текстура и трябва да има няколко различни фълтър отговорни вектора, а всички останали са техни „шумни“ вариации. Затова K-means подход може да предложи малко множество от прототипни филтър отговори и центровете на клъстери се считат за представляващи вектор отговарящ за тях.

Вариации

Тези филтър отговори са всички свързани зиображения, затова те се конкратенират за да се получат К клъстерни центъра, така наречените текстони.

## Моделиране на статистическа хистограма

Всеки клас има 4 обучителни изображения от всякакви осветяващи условия. Те се използвата за построяването на една хистограма. Ние имаме 8 класа, за това ще бъдат генерирани 8 хистограми. Всеки пиксел от дадено изображение се слага до най – близкия текстон в речникът, който вече имаме. Затова на всеки пиксел се слага точно едно текстоново число и хистограмата се нормализира за да се построи вероятностна плътностна функция (PDF). Интересен факт е, че всички хистограмни графи на статистически модели са близки а PDF на нестотинки са очевидно различни. Това ни позволява да класифицираме стотинки и нестотинки само сравнявайки хистограмите им.

## Дискусия около класифицирането на текстони

Най-големият проблем при класифицирането е скалирането. Ако размерите на тестовото множество са значително по – различни от размерите на учебното множество, нивото на вярност намалява. Възможно решение за този проблем е да добавим повече учебни изображения с различни размери, но така статистическият модел може да се промени. Ако размерите на изображенията се намалят, то тогава изпълнението ще се подобри. Друг проблем е, че класификатора на текстони има нужда от детайлни пространствени характеристики за да тества изображения. Последният проблем е, че в естествено изображение има относително много изображения на нестотинки със същите пространствени характеристики като на стотинки. И тъй като класификаторът не разчита на цвят на соттинките, нестотинки с лице или малък текст в тях ще бъдат разпознати като стотинки.

# Библиография

1. <http://www.robots.ox.ac.uk/~vgg/research/texclass/with.html>
2. <http://en.wikipedia.org/wiki/Canny_edge_detector>
3. <http://en.wikipedia.org/wiki/Blob_extraction>
4. <http://www.robots.ox.ac.uk/~vgg/research/texclass/figs/with/mr8filters.jpg>