Բովանդակություն

Ներածություն………………………………………………………………………………… 5

Խնդրի դրվածքը……………………………………………………………………………… 6

Եզրակացություն……………………………………………………………………………….……22

Գրականություն․․․․․․․․․․․․․․․․․․․․․․․․․․․․․․․․․․․․․․․․․․․․․․․․․․․․․․․․․․․․․23

**Ներածություն**

Տեղեկատվական տեխնոլոգիաների զարգացման հետ զուգընթաց մարդիկ պատկերները պահպանում են ոչ թե թղթերի այլ սարքրի մեջ և կարևոր խնդիր է իրենից ներկայացնում այդ նկարների ուսումնասիրությունը։

Այդ ուսումնասիրությունների արդյունքում ստեղծվել են բազմաթիվ համակարգեր որոնք հեշտացնում են մարդկանց կյանքը։

Այդպիսի օրինակներից են մատնահետքերի, աչքերի, անձի և այլ բազմաթիվ ճանաչման համակարգերը։ Պատկերների ուսումնասիրությունը հանդիսնում է արհեստական բանականույան կարևորագույն ճյուղերից մեկը։

Պատկերների ճանաչումը օգտագործվում է ռոբոտաշինության, ավտոարտադրության և այլ բազմաթիվ ճյուղերում։

**Խնդրի դրվածքը**

Խնդիր է դրծավ ուսումնասիրել պատկերների հետ աշխատանքը, որը հանդիսանում է արհեստական բանականության ճյուղերից մեկը, և հետազոտել օբյեկտները պատկերների մեջ։

**Մորֆոլոգիական պատկերի մշակում**

* նախաբան
* էրոզիա
* ընդլայնում
* բացում
* փակում

**Նախաբան**

Մորֆոլոգիանմաթեմատիկան մի լեզու է որը բաղկացած է թեորեմների ցանցից։ Պատկերի մշակման խնդիրների լուծման մեջ մեծ ներդրում ունի մորֆոլոգիական մաթեմատիկան։

Պատկերների արատապատկերում և տրանսլյացիան լայնորեն կիրառվում են ձևաբանության մեջ։

B հավաքածուի արատապատկերում ՝ սահմանվում է հետևյալ կերպ՝

= {w | w = -b, for b }

Եթե B-ն իրենից ներկայացնում է 2D պատկեր ապա -ն իրենից կներկայացնի նույն պատկերը սակայն (x,y) կորդինատներին կհամապատասխանեն (-x, -y) կորդինատները։



B-ի տրանսլյացիան z կետի նկատմամբ z = (, ) ` սահմանվում է հետևյալ կերպ՝

= {c | c = b + z, for b }

Եթե B-ն իրենից ներկայացնում է պիքսելների հավաքածու, որը բնորոշում է նկարի միջի օբյեկտը ապա –ը իրենից ներկայացնում է նույն B –ն որտեղ (x,y) կոորդինատները փոխարինված են (x + , y + ) կորդինատներով:

Ներածությունը թե ինչպես ենք էլեմենտների կառուցվածքը ուսումնասիրում մորֆոլոգիաիմեջ բերված է նկարներում ՝



Առաջին տողը իրենից ներկայացնում է էլեմենտների կառուցվածքը ։

Երկրորդ տողը էլեմենտների կառուցվածքը ձևափոխված ուղղանկյունների տեսքով։



Կատարենք գործողությունները A տարրի նկատմամբ օգտագործելով B տարրը։

Տարրերի այն դաշտերը որոնք մուգ են ներկված համարում ենք որ էլեմենտներ կան․

Նոր տարրը ստանալու համար A –ի բոլոր կետերում տեղադրենք B-ն և եթե այդ կետերում պարունակվի B-ն ապա այդ կետերը կհամարենք էլեմենտ։

**Էրոզիա**

Էրոզիան A-ն B-ի վրա ՝ A 󠇝 B սահմանվում է հետևյալ կերպ՝

A 󠇝 B = {z | A }

Բառերով ասված սա նշանակում է որ էրոզիան A-ն B-ի վրա

դա z էլեմենտների բազմություն է , B-ն ենթարկվել է տրանսլյացիաի z –ով, որոնք պարունակվում են A-ում։

Ուրիշ կերպ այս արտահայտությունը կարող ենք գրել հետևյալ կերպ՝

A 󠇝 B = { z | = }

Որտեղ նշանակում է՝

= {w | w A }

Այսինքն այն էլեմենտները որոնք չեն պարունակվում A –ում։

Նկարում պատկերված է օրինակ՝

|  |
| --- |
|  |
| Օգտագործելով էռոզիան փորձում ենք ջնջել նկարի որոշ կոմպոնենտներ` 486 486 թվային պատկերը էռոզիայի ենք ենթարկում 11 11, 15 15, 45 45 քառակուսի էլեմենտներով։ |

**Ընդլայնում**

Ընդլայնում A-ն B-ի վրա ՝ A 󠇝 B սահմանվում է հետևյալ կերպ՝

A 󠇝 B = { z | A }

Այլ կերպ սա կարող ենք ներկայացնել հետևյալ կերպ՝

A 󠇝 B = {z | A A }

|  |
| --- |
|  |
| Ալգորիթմի օրինակ |

Այս մեթոդը կարող է օգտագործվել տեքստը վերականգնելու համար։

|  |
| --- |
|  |
| Տեքստի վերականգման օրինակ |

**Երկակիություն**

Էրոզիան և ընդլայնումը կապված են իրար հետ։

=

և

=

**Բացում և Փակում**

Բացում A-ն B-ի վրա ՝ A 󠇝 B սահմանվում է հետևյալ կերպ՝

A 󠇝 B = () B

Փակում A-ն B-ի վրա ՝ A 󠇝 B սահմանվում է հետևյալ կերպ՝

A 󠇝 B = () B

|  |
| --- |
|  |
| Նկ 1 |

Իրենց մեջ ևս գոյություն ունի երկակիություն՝

=

և =

**Որոշ մորֆոլոգիական ալգորիթմների օրինակներ**

Եզրագծերի հայտնաբերումը A-ն ՝ (A) սահմանվում է հետևյալ կերպ՝

(A) =

|  |
| --- |
|  |
| Նկարում 1-ը նշված է սպիտակ իսկ 0-ն սև |

**Սեգմենտավորում**

Համակարգչային տեսողության կարևոր խնդիրներից մեկն է հանդիսանում ինֆորմացիայի դուրսբերումը պատկերից։ Որպիսզի կարողանանք մշակել նկարները պետք է այն բաժանենք որոշակի օբյեկտների։

Պատկերների սեգմենտացիան օգտագործվում է թվային պատկերները որոշակի սեգմենտների(մասերի) բաժանելու համար, այլ կերպ դա նաև անվանում են պիքսելների ցանց կամ սուպեր֊պիքսելենրի հավաքածու։ Պատկերները սեգմենտավորելու իմաստը կայանում է նրանում որ սեգմենտների հետ աշխատանքը զգալիորեն հեշտացնում է պատկերի հետ աշխատանքը ավելի հեշտ է լինում պատկերների մեջ օբյեկտների հետազոտությունը և փոփոխությունը։

Սովորաբար օգտագործվում է պատկերներում օբյեկտների հայտնաբերման և սահմանների հայտնաբերման համար։ Գոյություն ունեն սեգմենտացիայի բազմաթիվ ալգորիթմներ, սակայն խնդրից կախված ալգորիթմները կարող են փոփոխվել, ասինքն մեկ խնդրի համար տրված ալգորիթմը լինի լավը իսկ մյուսի խնդրի համար վատ։

Սեգմենտավորման ալգորիթմների հիմնական դասակարգումներն են՝

1. Շեմային եղանակներ

2. միջակայքերի աճեցում

3․Սահմանային ալգորիթմները

4. Կլաստերիզացիայի հիմման վրա

**Շեմային եղանակներ**

Պատկերների սեգմենտավորման պարզագույն մեթոդներից մեկը դա շեմային մեթոդն է։ Եթե անհրաժեշտ է ուսումնասիրել նկարը, որի մեջ կան գույների զգալի տատանումներ ապա այդ նկարների համար նպատակահարմար է օգտագործել այս մեթոդը։

Ենթադրենք որ հիստոգրամայի ինտենսիվությունը նկարում համապատասխանում է f(x,y)֊ին, որը իրենից ներկայացնում է լուսավոր օբյեկտները մուգ ֆոնի վրա, այնպես որ պատկերի միջի օբյեկտները և նրանց հետևի ֆոների պիքսելների ինտենսիվության արժեքները խմբավորված են երկու գերիշխող ռեժիմներում նկ10(ա).

|  |
| --- |
|  |
| Նկ10 |

Ակնհայտ եղանակներից մեկը օբյեկտը հետևի ֆոնից առանձնացնելու դա շեմի ընտրությունն է ՝ T որի միջոցով կառանձնացնեք օբյետկը, այսպիսով յուրաքանչյուր (x,y) կորդինատի վրայով կանցնենք և եթե f(x,y) > T կանվանենք օբյեկտի կետ, իսկ մյուս դեպքում կանվանենք հետևի ֆոնի կետ։

Այսպիսով սեգմենտավորված պատկերը կտրվի հետևյալ բանաձևով ՝ g(x,y)

|  |  |
| --- | --- |
| g(x,y) = | Եթե f(x,y) > T |
| Եթե f(x,y) T |

T-ի արժեքը ամբողջ պատկերի համար կարող է լինել նույնը սակայն կախված հարևան պիքսելենրի միջին ինտենսիվությունից կարող է փոփոխվել։

Նկ 10(բ) –ում ցույց է տրված ավելի դժվար շեմավորման պրոբլեմ, որտեղ պիքսելների ինտենսիվության արժեքները խմբավորված են երեք գերիշխող ռեժիմներում , օրինակի համար 2 տիպի սպիտակ օբյեկտները մուգ հետևի ֆոնի վրա։

Այսպիսի իրավիճակում պետք է կատարենք բազմակի շերտավորում, եթե մեր f(x,y) < արժեքից ապա այն պատկանում է 1 օբյեկտին եթե < f(x,y) ապա այն պատկանում է 2րդ օբյեկտին, մնացած դեպքերում այլ օբյեկտներին կամ հետևի ֆոնին f(x,y) > ։

|  |  |
| --- | --- |
| g(x,y) = | Եթե f(x,y) |
| Եթե < f(x,y) |
| Եթե f(x,y) < |

**Կլաստերիզացիայի հիմման վրա**

K-means ալգորիթմը իրենից ներկայացնում է իտերացիոն ալգորիթմ, որը օգտագործվում է պատկերը մասնատելու համար ՝ K կլաստերների։

K-means ալգորիթմը կլաստերիզացիայի հայտնի ալգորիթմներից մեկն է հանդիսանում։ Հորինվել է 1950 ական թվականներին։

**Լցոնում**

Պատկերը սեգմենտավորելու կարևոր ալգորիթմներից մեկն է հանդիսանում flood fill-ը, որը հայտնի է նաև region growing անունով։

**Պատկերի բուրգեր**

Սովորաբար մենք խնդիր ենք ունենում օրիգինալ պատկերի չափերը փոփոխելու, դրա համար գոյություն ունի 2 մեթոդ՝

* Մեծացնել չափերը(մոտիկացնել)
* Փոքրացնել չափերը (հեռվացնել)

Պատկերի բուրգերը իրենցից ներկայացնում են պատկերների հավաքածու որոնք ստացվում են օրիգինալ պատկերից, օրիգինալ նկարը դաունսեմպլինգի(downsampling) ենթարկելով մինչև մի ցանկալի տարանցիկ կետի հասնելը։

Դաունսեմպլինգը գործողություն է որի ընթացքում փոխում ենք տրված պատկերի չափերը 72 dpi. : Սա կարող է զգալիորեն փոքրացնել պատկերի ծավալը։

Դաունսեմպլինգի վատ կողմը կայանում է նրանում, որ երբ պատկերը մոտիկացնում ենք 100%֊-ից ավել ապա արդեն պատկերը սկսում է երևալ ոչ ճշգրիտ։

Այսպիսով յուրաքանչյուր պատկեր դառնում է ¼ մասը իր նախորդի։ Այս ալգորիթմը սովորաբար օգտագործվում է պատկերների սեգմենտացիայի մեջ, փոքրացնում է նկարի չափերը որից հետո հնարավոր է լինում որոշակի գործողություններ կատարել պատկերի հետ և հետո ավելի բարձր որակով պատկեր ետ բերել։

Պատկերի բուրգերի հիմնկան տեսակները ՝

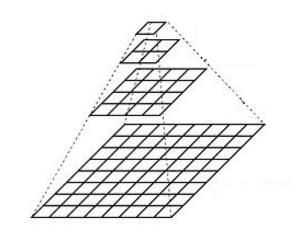
* Գաուսիան բուրգեր
* Լապլասիան բուրգեր

**Գաուսիան բուրգեր ։**  Օգտագործվում է պատկերները դաունսեմպլինգի ենթարկելու համար։

**Լապլասիան բուրգեր**։ Օգտագործվում է upsampled-ի ենթարկելու համար, որի ժամանակ բուրգի պատկերների փոքր չափերը կարելի է մեծացնել։

**Գաուսիան բուրգեր**

* Պատկերացնենք մի բուրգ որի կազմված է բազմաթիվ շերտերից և որքան ավելի վերև ենք բարձրանում այնքան այդ շերտերի չափերը փոքրանում են։



* Շերտերը համարակալենք ներքևից վերև, այնպես որ լինի ավելի փոքր քան -ը։
* Ամեն (i+1)-րդ շերտը ստանալու համար մենք պետք է անենք հետևալ գործողությունները:

1. Կոնվոլուցիայի ենթարկենք -ը Գաուսիան միջուկի հետ։

* Մենք հեշտությամբ կարող ենք համոզվել նրանում որ ելքային պատկերը ստացվում է մուտքային պատկերի ¼ տարածքի չափով։

|  |  |
| --- | --- |
|  |  |
| Օրիգինալ մուտքային պատկերը | Ելքային պատկերը դաունսեմպլիգի ենթարկելուց հետո |

* Մենք նաև կարող ենք իրականացնել հակառակ գործողությունը՝ այսինքն 4 անգամ մեծացնել(upsampling):

|  |  |
| --- | --- |
|  |  |
| Օրիգինալ մուտքային պատկերը | Ելքային պատկերը upsampling-ի ենթարկելուց հետո |

Եթե փորձենք պատկերը մեկ անգամ ենթարկել դաունսամպլինգի մյուս անգամ upsamplingi կստանանք հետևյալ պատկերը՝

|  |  |  |
| --- | --- | --- |
|  |  |  |
| Օրիգինալ պատկեր | Դաունսամպլինգի ենթարկելուց հետո | Upsampling –ից հետո |

**Կոնվոլուցիա։**

Պատկերների մշակման մեջ միջուկը(kernel), կոնվոլյուցիայի մատրիցան կամ դիմակը(mask) իրենցից ներկայացնում են փոքր չափերի մատրիցա։ Դա շատ օգտակար է քողարկման(bluring), եզրերի հայտնաբերման և այլ շատ վայրերում։

Միջուկը օրիգինալ պատկերի հետ կոնվոլուցիայի են ենթարկում և կախված միջուկի մատրիցայի չափից և կառուցվածքից կարող են ստացվել տարբեր արդյունքներ։

|  |  |  |
| --- | --- | --- |
| ՕՊԵՐԱՑԻԱՆ | ՄԻՋՈՒԿԸ | ՊԱՏԿԵՐԻ ԱՐԴՅՈՒՆՔԸ |
| Ինքնության ճանաչում |  | Vd-Orig.png |
| Եզրերի հայտնաբերման համար |  | Vd-Edge1.png |
|  | Vd-Edge2.png |
|  | Vd-Edge3.png |
| Սրում |  | Vd-Sharp.png |
| Տուփի քողարկում |  | Vd-Blur2.png |
| Գաուսիան քողարկում 3x3 |  | Vd-Blur1.png |
| Գաուսիան քողարկում 5x5 |  | Vd-Blur Gaussian 5x5.png |
| Շրջանային դիմակավորում  5 x 5 |  | Vd-Unsharp 5x5.png |

Մատրիցների կոնվոլուցիայի պարզագույն օրինակ կարող ենք բերել հետևյալը՝

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | 1 | 2 | 3 | | 4 | 5 | 6 | | 7 | 8 | 9 | | Օրիգինալ | | | | |  |  |  | | --- | --- | --- | | -1 | -2 | -1 | | 0 | 0 | 0 | | 1 | 2 | 1 | | Միջուկ | | | | |  |  |  | | --- | --- | --- | | -13 | -20 | -17 | | -18 | -24 | -18 | | 13 | 20 | 17 | | Ելքային | | | |

Պատկերը բաժանում ենք փոքր մատրիցաների որոնց չափերը կարող են համընկնել կամ չհամընկնել միջուկի չափերի հետ։

Առաջին քայլով կատարում ենք միջուկի պտույտ 180-ով,

որից հետո ստանում ենք հետևյալ մատրիցը

|  |  |  |
| --- | --- | --- |
| 1 | 2 | 1 |
| 0 | 0 | 0 |
| -1 | -2 | -1 |
| Միջուկ | | |

Այնուհետև կատարում ենք մատրիցաների բազմապատկում։ Միջուկից դուրս ընկած կետերը համարում ենք 0-ներ։

|  |  |
| --- | --- |
| y[0,0] | y[0,0] = 0\*1+0\*2+(-2\*4) + (-1\*5) = -13 |
| y[1,0] | y[1,0] = 0\*1 + 0\*2 + 0\*3 + (-1\*4) + (-2 \* 5) \* (-1 \* 6 ) = -20 |
| y[2,0] | y[2,0] = 0\*2 + 0\*3 + (-1 \* 5 ) + (-2 \* 6) = -17 |
| y[0,1] | y[0,1] = 2\*1+1\*2+0\*4+0\*5+(-2\*7) + (-1\*8) = -18 |
| y[1,1] | y[1,1] = 1\*1 + 2\*2 + 1\*3 + 0\*4 + 0\*5 + 0\*6 + (-1\*7) + (-2 \* 8) + (-1 \* 9) = -24 |
| y[2,1] | y[2,1] = 1\*2+2\*3+0\*5+0\*6+(-1\*8) + (-2\*9) = -18 |
| y[0,2] | y[0,2] = 2\*4+1\*5+0\*7+0\*8 = 13 |
| y[1,2] | y[1,2] = 1\*4+2\*5+1\*6+0\*7+0\*8+0\*9 = 20 |
| y[2,2] | y[2,2] = 1\*5 + 2\*6 + 0\*8 + 0\*9 = 17 |

**Լապլասիան բուրգեր**

* Լապլասիան բուրգը շատ նման է եզրագծերի հայտնաբերման ալգորիթմին
* Բուրգերի պիքսելների մեծ մասը 0 են(սև)
* Այս ալգորիթմը շատ է օգտագործվում պատկերների սեղղման մեջ։

Լապլասիան բուրգերը ստացվում են Գաուսիան բուրգերի հիմման վրա՝

= – EXPAND[]

= – EXPAND[]

= – EXPAND[]

=

……………………

Այսպիսով մենք ստացանք Լապլասիան բուրգերը Գաուսիանի հիմման վրա, սակայն հնարավոր է և հակառակ գործողությունը։

=

= + EXPAND[]

= + EXPAND[]

= + EXPAND[]

-ը մեզ մոտ կլինի օրիգինալ պատկերը։

**Պատկերների ձևափոխում**

Օգտագործվում է նրա համար որպիսզի կարողանանք պատկերը վերափոխենք այլ տեսքի որի մրջոցով կարողանանք լուծել պատկերների ուսումնասիրության որոշակի խնդիրներ։ Այդ մեթոդների օրինակներ են հանդիսանում արտիֆակտները որոնք օգտագործվում են պատկերի եզրագծերը հայտնաբերելու համար, պատկերը վերափոխելուց հետո մենք կարողանում ենք պատկերի մեջ գտնել գծերը և կլորները։

Ուսումնասիրություններ են կատարվել պատկերների ձգման,նեղացման,կորացման և պտտման գործողություններ մեջ։

Հայտնի վերափոխման ալգորիթմներից մեկն է հանդիսանում Ֆոուրիերի ալգորիթմը։



**//@TODO-1-2 էջ կարամ ավելացնեմ**

Ուսումնասիրություններ են կատարվել դիսկրետ Ֆոուրիերի վերափոխման(DFT) և դիսկրետ Կոզինի վերափոխման(DCT) ալգորիթմներում։

Ուսումանսիրություններ կկատարվեն նաև ինտեգրալ պատկերների մեջ որոնք օգտակար են պատկերների մեջ դեմքի հայտնաբերման ալգորիթմների ուսումնասիրությունների համար։

Այսպիսով մենք կուսումնասիրենք հետևյալ թեմաները։

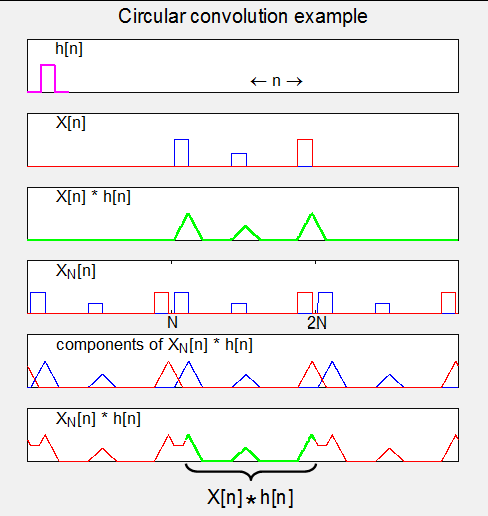
* Gradients and sobel derivatives
* The Laplace and canny transforms
* The line and circle Hough transforms
* Geometric transforms: stretch, shrink, warp, and rotate
* Discrete Fourier Transform (DFT) and Discrete Cosine Transform (DCT)
* Integral images
* Distance transforms
* Histogram equalization

**Gradients and sobel derivatives**

Համակարգչային տեսողության մեջ առանցքային բաներից մեկն է հանդիսանում եզրագծերի հայտնաբերումը պատկերների մեջ, և դա շատ սերտ կապված է պատկերների ածանցյալների մոտիկացման արժեքների հայտաբերման մեջ։ Ածանցյալը ցույց է տալիս փոփոխությունը տվյալ ֆունկցիայի կամ մուտքային ազդանշանի որոշակի չափով։

Երբ մենք գտնում ենք ածանցյալի լոկալ մեծագույնը , դա մեզ կտա այն հատվածները որտեղ ազդանշանը ամենաշատն է տատանվում, որը պատկերի համար կարող է նշանակել եզրագիծ:

Կոնվոլյուցիա մաթեմատիկայում օպերացիա է 2 ֆունկցիաների f և g, որի արդյունքում ստացվում է 3-դ ֆունկցիան, որը իրենից ներկայացվում է փոփոխված ֆունկցիա օրիգինալ ֆունկցիաներից որևիցե մեկից։



Գոյություն ունի դիսկրետ ազդանշանների մոտեցման տարբերակ միջուկի կոնվոլյուցիայի միջոցով։ Կոնվոլյուցիան հինականում նշանակում է պատկերի յուրաքանչյուր հատվածի ձևափոխում։

Հաճախակի օգտագործվող ձևափոխությունը դիֆերենցացիան(differentiation) հանդիսանում է Սոբելի ֆիլտրը, որը աշխատում է հորիզոնական, ուղղահայաց և նույնիսկ միքսված մասնակի ածանցյալների ցանկացած հաջորդականությամբ։

Որպիսզի մոտեցնենք հորիզոնական ածանցյալի արժեքին , հետևյալ Սոբելի մատրիցան կոնվոլյուցիայի ենք ենթարկում մուտքային մատրիցայի հետ։

= \*

Սա նշանակում է հետևյալը, որ յուրաքանչյուր մուտքային պիքսելի, հաշվարկային արժեքը լինում է իր վերևի-աջ հարևանին գումարած 2 անգամ իր ձախ հարևանի գումարած իր ներքևի-աջ հարևանի հանած իր վերևի-ձախ հարևանի հանած 2 անգամ ձախ հարևանի հանած իր ներքևի ձախ հարևանի արժեքը, որի արդյունքում կստանանք նոր պատկեր։

Սոբելի օպերատորը օգտագործում է երկու 3x3 չափերի միջուկներ, որոնք կոնվոլյուցիայի են ենթարկվում օրիգինալ պատկերի հետ որպիսզի հաշվարկենք պատկերների ածանցյալների մոտիկացման արժեքները, մեկը օգտագործվում է հորիզոնական փոփոխությունների համար մյուսը ուղղահայաց փոփոխությունների համար։

Եթե մենք հայտարարենք A-ն որպես մուտքային մատրից և և երկու պատկերներ որոնք յուրաքանչյուր կետում պարունակում են հորիզոնական և ուղղահայաց ածանցյալների մոտիկացումները և դրանք հաշվարկվում են հետևյալ կերպ։

= \* = \*

Որտեղ \* նշանակում է կոնվոլյուցիան երկչափ տարածության մեջ։

Նաև կարող ենք այս հավասարությունները ներկայացնել հետևյալ կերպ։

=

G =

Օգտագործելով այս ինֆորմացիան մենք կարող ենք հաշվարկել նաև գրադիենտի ուղղությունը։

Θ = atan

Որտեղ, օրինակի համար Θ հանդիսանում է 0 ուղղահայաց եզրագծի համար։

|  |  |
| --- | --- |
| C:\Users\gharutyunyan\Downloads\DIP\Օռիգինալ.PNG | Օրիգինալ պատկեր |
| C:\Users\gharutyunyan\Downloads\DIP\ֆակե.PNG | Սոբելի օպերատորը կիրառելուց հետո |

Դիտարկենք օրինակի հիմման վրա՝

Ունենք մուտքային հետևյալ մատրիցը`

|  |  |  |  |
| --- | --- | --- | --- |
| 50 | 50 | 100 | 100 |
| 50 | 50 | 100 | 100 |
| 50 | 50 | 100 | 100 |
| 50 | 50 | 100 | 100 |
| 50 | 50 | 100 | 100 |

Հիմա փորձենք կոնվոլյուցիայի ենթարկել x ուղղությամբ միջուկի հետ։

[1,1] կոորդինատներովի արժեքը կլինի հետևյալը -50 + 0 + 100 – 100 + 0 + 200 - 50 + 0 +100 = 200

Եթե արժեքը լիներ բացասական ապա դա նորմալ կլիներ որովհետև վերջում մենք օգտվելու ենք հետևյալ բանաձևից։

G =

Այսպիսով հաշվելով բոլոր կոորդինատների արժեքները կստանանք նոր մատրից։

**Hough ձևափոխություն**

Hough – ի ձևափոխությունը դա միջոց է, որի օգնությամբ կարող ենք պատկերից առանձնացնել որիշակի ֆորմայով օբյեկտ։ Քանի որ այդ ֆորմաները ներկայացնելու համար մենք պետք կառուցենք այդ ֆորմաներին համապատասխանող որոշակի մաթեմատիկական հավասարումներ։ Սովորաբար այս ձևափոխությունը օգտագործվում է որոշակի կորեր հայտնաբերելու համար, օրինակ գծեր, շրջաններ, էլիպսներ։  
Ընդհանրական Hough – ի ձևափոխությունը կարելի է օգտագործել այն տեղերում որտեղ պարզ վերլուծական բացատրությունը օբյեկտի հնարավոր չէ իրականացնել։

Աշխատանքի սկզբունքը հետևյալն է՝

Հիմնական մոտեցումը օրինակ գծի հայտնաբերման մեջ Hough – ի մեթոդով հետևյալն է յուրաքանչյուր մուտքային չափում (օրինակ կոորդինատային կետ) ցույց է տալիս իր ներդրումը գլոբալ հետևողական լուծման համար(օրինակ ֆիզիկական գիծը որը առաջացրել է այդ պատկերի կետը)։

Որպես պարզ օրինակ կարող ենք դիտարկել հետևյալ ընդհանուր խնդիրը, կիրառումը գծային սեգմենտների խումբը դիսկրետ պատկերների կետերին վրա(օրինակ պիկսելների գտնվելու վայրերը եզրագծերի դետեկտորից)։ Նկար 1-ը ցույց է տալիս որոշ հնարավոր լուծումները այս խնդրի համար։ Այստեղ մենք ունենք գիտելիքների բաց ցանկալի գծային սեգմենտների վերաբերյալ(և անորոշության մասին, թե ինչ է իրենից ներկայացնում գծային սեգմենտը) և անել այս խնդիրը սահմանափակ։

|  |
| --- |
| http://homepages.inf.ed.ac.uk/rbf/HIPR2/figs/hough1.gif |
| Նկար 1 ա)կոորդինատային կետեր b) և c) հնարավոր ուղիղ գծային կցամասեր |

Մենք կարող ենք անալիտիկորեն բացահայտել գծային սեգմենտը մի քանի ֆորմաների մեջ։ Ինչևիցե հարմար հավասարումը որպիսզի կարողանանք բացատրել գծերի խումբը օգտագործում ենք պարամետրավորված կամ նորմալ հասկացողությունները։

xcosθ + ysinθ = r

որպեղ r-ը հանդիսանում է երկարությունը կոորդինատների սկզբնակետից մինչև այդ գիծը θ-ն դա կողմնորոշումն է(օրիենտացիան ) r-ի x առանցքի նկատմամբ(նկար 2), ցանկացած կետի համար (x,y) այս գծի վրայի r և θ հաստատուն են։

|  |
| --- |
| http://homepages.inf.ed.ac.uk/rbf/HIPR2/figs/hough2.gif |
| Նկար 2` ուղիղ գծի պարամետրիկ նկարագրությունը |

Պատկերի վերլուծության համատեքստում, կոորդինատները կետի(կետերի) եզրային սեգմենտների () պատկերի մեջ հայտնի են և հանդիսանում են կոնստանտ պարամետրեր հավասարման մեջ։ r-ը և θ-ն մեր հավասարման մեջ անհայտ պարամետրերն են։

Եթե մենք կառուցենք համապատասխան (r, θ) արժեքները () –ին համապատասխան, ապա կետերը կարտեզիան համակարգի պատկերի համար կհամապատասխանեն կորերին բևեռային Hough-ի պարամետրերի համակարգին։

Կետից դեպի կոր ձևափոխությունները համապատասխանում են Hough-ի ձևափոխմանը ուղիղ գծերի համար։

Եթե այդ հավասարումները փորձենք կազմել շրջանագծերի համար ապա կունենանք հետևյալ տեսքը՝

+ =

Որտեղ a-ն և b-ն շրջանագծի կենտրոնի կոորդինատներն են իսկ r-ը շառավիղը։

**Օբյեկտների Հայտնաբերումը**

Օբյեկտների հայտնաբերման համար գոյություն ունեն բազմաթիվ ալգորիթմներ։

Այս պարագրաֆում մենք կուսումնասիրենք դեմքերի հայտնաբերումը պատկերների մեջ, հայտնաբերման մեջ մենք կօգտվենք AdaBoost-ի մետա-ալգորիթմից, և Haar-ի դասակարգիչներից(classifier)։

**The boosting theory:** Դեմքերի հայտնաբերումը պատկերների մեջ կարելի է կատարել հետևյալ կերպ՝ կարող ենք ամբողջ պատկերը բաժանել փոքր կտորների(պատուհանների) և ստեղծենք դասակարգիչ, որը կբնորոշի, թե այդ փոքր պատուհանը հանդիսանում է դեմք թե ոչ, որ պատուհանը համապատասխանի դասակարգիչին կհամարվի դեմք։

Հիմա կփորձենք պարզել թե ինչ է իրենից ներկայացնում դասակարգիչը և ինչպես կարող ենք մենք պատրաստել այն: Դասակարգիչների խնդիրներից կարևորը հանդիսանում է այն թե մեզ հայտնի կատեգորիաներից որին է համապատասխանում մեր դասակարգիչը։ Որպես օրինակ կարող ենք դիտարկել հետևյալը թե մուտքային պատկերը համապատասխանում է բանանի,խնձորի, կամ տանձի կատեգորիաներին, որը մենք կարող ենք պարզել մրգի դասակարգիչի օգնությամբ։ Մեր պարագայում կլինի 2 կատեգորիա դեմք և ոչ դեմք։ Այս գլխում մենք կուսումնասիրենք ժամանակավոր ալգորիթմները որպիսզի կարողանանք ստեղծել ուժեղ դասակարգիչ օգտվելով թույլ սովորողների բազմությունը(set of weak learners)։

Թույլ սովորողների բազմությունը իրենցից ներկայացնում են դասակարգիչներ հիմնվելով որոշակի հատկությունների վրա, որոնք չեն կարող բաժանել ամբողջ բազմությունը 2 կատեգորիաների, բայց անում են լավ գործ բազմությունների մի մասի համար։ Oրինակի համար պատկերի մեջ փորձում ենք գտնել մորուք որպիսզի պարզենք այդ դեմքը տղամարդու է թե ոչ, եթե այդ դասակարգիչը չկարողանա գտնել բոլոր տղամարդկանց պատկերի մեջ համենայի դեպս կկատարի լավ գործ։  
AdaBoost-ը իրենից ներկայացնում է մետա-ալգորիթմ որը մեզ կoգնի որպիսզի կառուցենք դասակարգիչ։ Նրա հիմնական գործը կայանում է նրանում որպիսզի կառուցի ուժեղ դասակարգիչ հիմնվելով թույլ դասակարգիչների վրա, որոնք ավելի լավ են քան պատահականը(այսինքն ավելի լավ են բնորոշում պատկերը քան պատահականության սկզբունքի վրա հիմնվելով)։

H(x) = sign((x) + (x) + … + (x))

Sign օպերատորը կվերադարձնի +1, երբ փակագծերի մեջինը լինի դրական և -1 հակառակ դեպքում։ Այսպիսով դասակարիչը ասում է այո(+1) կամ ոչ(-1) այսիքն պատկերի մեջ գտնվում է թե ոչ։ Այսպիսով -ն տրված (x) դասակարգիչի քաշն է, տրված x-ի համար T դասակարգիչների մեջ։

Որպես օրինակ կարող ենք դիտարկել հետևյալ դեպքը՝ ունենք մուտքաային պատկեր և նրա մեջ ուզում ենք հայտնաբերել արդյոք պատկերի վրայի մարդը կին է թե տղամարդ։

Որպես թույլ դասակարգիչներ կարող ենք ընտրել հետևյալները՝

* : Եթե բարձրությունը մեծ է քան ~175սմ, ապա այդ մարդը տղամարդ է հակառակ դեպքում կին։ Կան շատ կանայք որոնք ավելի բարձրահասակ են քան տղամարդիք, բայց սովորաբար հակառակն է։
* : Եթե մարդը ունի երկար մազեր ապա այդ մարդը կին է։ Կան շատ տղամարդիք որոնց մազերը ավելի երկար են քան կանացը, բայց սովորաբար հակառակն է։
* : Եթե մարդը ունի մորուք ապա այդ մարդը տղամարդ է։ Այստեղ մենք կարող ենք սխավել որովհետև մարդիք թրաշվում են։

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Անուն** | **Հասակ ()** | **Մազեր ()** | **Մորուք ()** | **Սեռ (f(x))** |
| Աննա | 1.69 | Երկար | Անթրաշ | Կին |
| Վահան | 1.75 | Կարճ | Անթրաշ | Տղամարդ |
| Գևորգ | 1.80 | Կարճ | Անթրաշ | Տղամարդ |
| Արամ | 1.83 | Կարճ | Թրաշով | Տղամարդ |
| Սարա | 1.77 | Կարճ | Անթրաշ | Կին |

**-**դասակարգիչը ճիշտ է ցույց տալիս 3 մարդու, **-**դասակարգիչը ճիշտ է ցույց տալիս 4 մարդու, -ը 3 մարդու։ Հետո մենք պետք է ընտրենք -ը քանի որ այնտեղ ամենաքիչն ենք սխալվել և նշանակենք նրա -ն։ Սրանից հետո մենք պետք է մեծացնենք սխալ դասակարգված տվյալների(Սարա) քաշը և փոքրացնենք մնացածի քաշը(Աննա, Վահան, Գևորգ, Արամ)։ Հետո մենք պետք է նայենք մյուս ավելի ճիշտ հայտնաբերած դասակարգչին։ Այս քայլից հետո քանի որ Սարա-ի քաշը ամենամեծն է մենք պետք է կենտրոնանանք նրա վրա։

Մեր մուտքային դասակարգիչները նշանակենք t տառով։ Յուրաքանչյուր (t= {1, 2, … T}) ասոցացվում է քաշ որը օգտագործբում է բոլոր դասակարգիչների արդյունքները հավաքելիս։ X-ը մուտքային հատկությունների վեկտոր X = (, , …, , … , ) որտեղ i-ն հատկության համարն է։ Պիտակները նշանակված են երկուականով։ (i)-ն մեզ մոտ պետք է սկզբում արժեքավորված լինի արժեքներով որը ցույց կտա թե մեզ վրա ինչքան թանկ կարող է նստել սխալ գնահատված հատկությունը։ Այս ալգորիթմի լավ կողմը կայանում է նրանում որ եթե մի թույլ դասակարգիչում սխալ ենք հայտնաբերել մի օբյեկտ ապա մյուս կետում կենտոնանում ենք այդ կետի վրա և այդպես շարունակ, որը գնալով կնվազեցնի սխալվելու հավանականությունը։

Ալգորիթմի աշխատանքը հետևյալն է՝

1. (i) = 1 / m for i = 1, …, m։
2. t = 1, …, T:
   1. Գտնել դասակարգիչը որը մինիմալացնում է (i) քաշի սխալը։
   2. = arg որտեղ = (for ≠ ())

Եթե < 0,5 ուեմն կանգնում ենք այդ կետում։

**2․3** Ընտրում ենք R սովորաբար = որտեղ մինիմալ սխալանքնե է նախորդ քայլի։

**2․4** Թարմացնում ենք քաշերը։

(i) =

Որտեղ իրենից ներկայացնում է նորմալիզացնող պարամետր ամբողջ տվյալների ցուցիչների համար։

ԵՎ վերջում ունենում ենք հետևյալ պատկերը՝

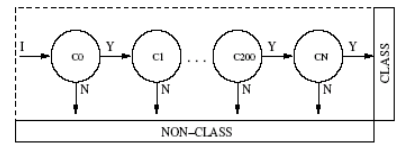
H(x) = sign()

**Viola Johnes Օբյեկտների Հայտնաբերում**

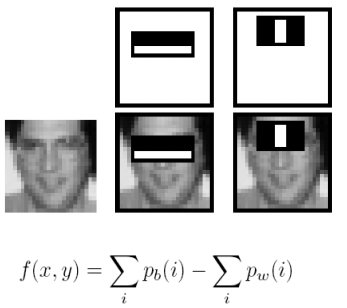
Viola-Jones հայտնաբերիչի միջոցով կառուցվում են որոշ թույլ դասակարգիչները։ Այս հայտանաբերիչը աշխատում է հիմնվելով ստատիստիկ մեթոդների վրա։

Յուրաքանչյուր թույլ հայտնաբերիչ իրենից ներկայացնում է չափազանց պարզ երկուական դասակարգիչ։

Ուսումնասիրությունները կատարվում է կասկադների միջոցով

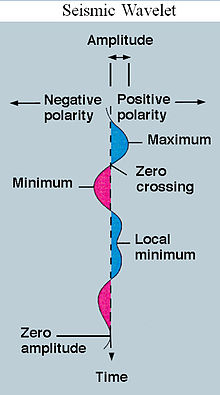


Եթե պատկերից առանձնացրած ուղղանկյունը անցնում է բոլոր կասկադների միջով և տալիս է դրական արդյունք ապա ասում ենք որ այդ ուղղանկյունը պատկանում է տրված դասին հակարակ դեպքում ոչ։ Հիմնական թույլ դասակարգիչները հիմնված են հասարակ վիզուալ առանձնահատկությունների վրա(հիմնականաում այդ առանձնահատկությունները սովորաբար անվանում են Haar-ի նման առանձնահատկություններ)։

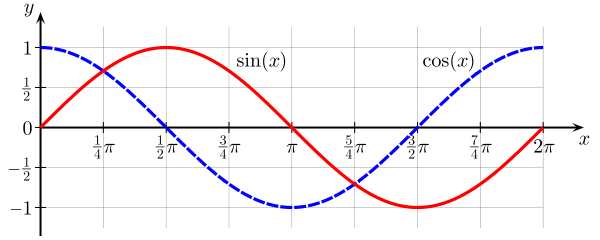


**Haar-like features**

Haar-like features իրենցից ներկայացնում են թվային պատկերների առանձնահատկություննները որոնք օգտագործվում են օբյեկտների հայտնաբերման մեջ։ Այս ալգորիթմը իր անունը ստացել է Haar-ի wavelet-ի հիմման վրա, որի միցոցով առաջին անգամ կառուցվել է ներկա-ժամանկի(real-time) օբյեկտների հայտնաբերիչը։



Wavelet անալիզը բավականին նման է Ֆյուրեի- անալիզին։ Նրա էությունը կայանում է հետևյալում մաթեմատիկայում Ֆյուրեի անալիզը ուսումնասիրում է թե ինչպես ընդհանուր ֆունկցիաները կարող են ներկայացվել կամ բացատրվեն ավելի պարզ տրիգրոմետրիկ ֆունկցիաների միջոցով։ Մաթեմատիկայում տրիգրոմետրիկ ֆունկցիաները անվանում են նաև շրջանաձև ֆունկզիաներ և իրենցից ներկայացնում են անկյունային ֆունկցիաներ։ Ամենահայտնի և սովորաբար օգտագործվող տրիգոնոմետրիկ ֆուկցիաներն են sine, cosine.



Wavelet-ները իրենցից ներկայացնում են ալիք օրինակ տատանում, որը ունի ամպլիտուդ և սկսում է 0-ից բարձրանում հետո նվազում հետ մինչը 0։

Wavlet անալիզը մեզ հնարավորություն է տալիս թիրախային ֆունկցիան որոշակի ինտերվալում ներկայացնել օրթոնորմալ բազիսի մեջոցով։ Haar-ը օգտագործել է ֆունկցիաները որպիսզի բերի օրինակ օրթոնորմալ համակարգի, քառակուսային-ինտեգրացվող ֆունկցիաների համար միավոր ընդմիջումից[0,1]։ Այս հատկությունը իրանից ներկայացնում է առավելություն սիգնալների անալիզի ժամանակ,կարող է օգտագործվել որպես մեքենաների մոնիտորինգի գործիք։

**Viola Johns դեմքերի հայտնաբերում**

Եզրագծերի հայտնաբերումը պատկերի մեջ կատարվում է կոնվոլյուցիայի մատրիցի միջոցով։

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Image result for edge detection horizontal line |  | |  |  |  | | --- | --- | --- | | **-1** | **-1** | **-1** | | **2** | **2** | **2** | | **-1** | **-1** | **-1** | |  | Image result for edge detection horizontal line |

Այս միջուկի միջոցով մենք փորձելու ենք հորորիզոնական գծերը։

**Haar հատկություններ**

Haar-ի հատկությունները նմանատիպ են կոնվոլյուցիայի միջուկին որը հնարավորություն է տալիս հայտնաբերել որոշակի հատկնություններ տրված պատկերի մեջ։

|  |
| --- |
| Image result for haar feature 5 types |
| Նկ․1 |

Այստեղ մենք աշխատելու ենք դեմքի հայտնաբերման հետ։ Ի սկզբանե ալգորիթմը պետք է ունենա շատ դրական նկարներ(որոնց մեջ կա դեմք) և շատ բացասական նկարներ(որոնց մեջ չկա դեմք) որպիսզի վերապատրաստի դասակարգիչը։ Դրանից հետո մենք պետք է կարողանանք առանձնացնել հատկությունները։ Դրա համար օգտագործվում են haar-ի հատկությունները որոնք բերված են նկ1-ում։

Յուրաքանչյուր հատկության արդյունքը դա եզակի թիվ է, որը ստացվում է սպիտակ ուղղանկյան պիքսելների գումարից հանելով սև ուղղանկյան պիքսելներ գումարը։ Հիմա բոլոր հնարավոր չափերը և դիրքերը յուրաքանչյուր միջուկի օգտագործվում է հաշվարկելու համար հատկությունների մեծ քանակություն։ Եթե վերցնենք 24x24 պատուհան ապա արդյունքը կլինի 160000+ հատկություն։ Յուրաքանչյուր հատկության հաշվարկի համար մենք պետք է գտնենք պիքսելների գումարը սպիտակ և սև ուղղանյունների միջի։ Այս խնդիրը լուծելու համար օգտագործվում են ինտեգրալ պատկերները։ Դա բավականին հեշտացնում է պիքսելների հաշվարկի գործնթացը։

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | **1** | **1** | **1** | | **1** | **1** | **1** | | **1** | **1** | **1** | | **Գումարը վերև և ձախ** | |  |  |  | | --- | --- | --- | | **1** | **2** | **3** | | **2** | **4** | **6** | | **3** | **6** | **9** | |
| **Մուտքային պատկեր** |  | **Ինտեգրալ պատկեր** |

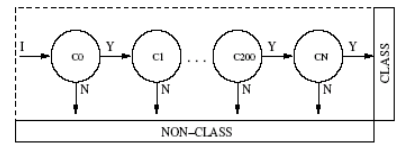
Բայց բոլոր այդ հատկությունները որ մենք հաշվարկում ենք դրանց մեծ մասը չունեն նշանակություն։

|  |
| --- |
| haar.png |
| **ՆԿ2** |

Նկ2 ում վերևի տողում ցուցադրված են 2 լավ հատկություննր։ Առաջին հատկորոշիչը հիմնված է նրա վրա որ սովորաբար աչքերի մասը լինում է ավելի մուգ քան քթի և թշերի մասը։ Երկրորդը հիմնված է նրա վրա որ ախքերի մասը ավելի մուգ է քան քթի կամուրջը։ Բայց այս հատկորոշիչները եթե տեղադրենք թշերի վրա կամ մեկ այլ վայրում մեզ ոչնչով չեն օգնի։ Այսպիսով մենք պետք է ընտրենք այդ 160000+ հատկություններից լավագույնները որոնք ավելի լավ կբնորոշեն մարդու դեմքը։ Այդ ընտրությունը կատարվում է Adaboost-ի ալգորիթմի միջոցով։

Դրա համար մենք կիրառում ենք յուրաքանչյուր հատկությունը բոլոր վերապատրաստող պատկերների վրա։ Յուրաքանչյուր հատկության վրա, նա գտնում է լավագույն շեմը, որը դասակարգում է դեմքերը դրական են թե ոչ։ Սակայն ակնհայտ է, որ այնտեղ կլինեն նաև սխալներ։ Մենք կընտրենք այն հատկությունները որոնք ավելի լավ կբնորոշեն թե պատկերը դեմք է թե ոչ։ Բոլոր պատկերները որոնք որ սկզբում պետք է ուսումնասիրենք ունենում են նույն քաշը։ Ամեն դասակարգումից հետո սխալ որոշված պատկերների քաշերը բարձրանոմ են։ Որից հետո կատարվում է նույն գործողությունը, հաշվարկվում նոր սխալները։ ԵՎ այդ պրոցեսը շարունակվում է այնքան ժամանակ քանի դեռ չենք ստացել մեզ անհրաժեշտ ճշտությունը կամ սխալանքի կոէֆիցենտը, կամ չենք ստացել մեզ համար անհրաժեշտ հատկությունների քանակը։

Վերջնական դասակարգիչը ստացվում է այդ թույլ դասակարգիչների հիմման վրա։ Դրանք կոչվում են թույլ քանի որ միայնակ այդ դասակարգիչը չի կարող դասակարգել պատկերը։ Վերջում մենք ստանում ենք 160000-ից մոտ 6000 հատկություն որոնք բնորոշում են դեմքը։ Այսպիսով եթե մենք վերձնենք 24x24 վրա պատկեր ապա նրա վրա պետք է կիրառենք 6000 հատկություն։ Այդ պատկերի վրա ռեգիոնների մեծ մասը դեմք չի։ Այսպիսով ավելի լավ է ունենալ հատկորոշիչներ որոնք կգտնեն այդ պատուհանը հանդիսանում է դեմք թե ոչ այսինք ինչ հատկությունների բավարարման դեպքում պատուհանը չի համարվի դեմք։ Եթե այդ հատկություններին չբավարարի կարող ենք նրա մեջ էլ դեմք չփնտրել։ Այսպիսով այստեղ գալիս է կասկադների գաղափարը որը իրենից ներկայացնում է հետևյալը 6000 ստուգում իրականացնելու փոխարեն մենք կանենք հետևյալ քայլերը մենք քայլ առ քայլ առաջ կգնանք եթե պատկերը չի բավարարում որևէ հատկության ապա այն դեմք չի հակառակ դեպքում շարունակում ենք դիտարկել։



Այսպիսով մենք փոքրացնում ենք 6000 թիվը։ Դեմքի հայտնաբերման համար ամենալավ դասակարգիչներն են համարվում վերևում բերված 2-ը։ Ներքևի նկարում ևս բերված են դասակարգիչներ որոնք ևս շատ կարևոր են։



**Եզրակացություն**

Այդ ուսումնասիրությունները մեծ կարևորություն ունեն մարդկանց համար, օրնակ տեսախցիկները որոնք վերահսկում են երթևեկությունը աշխատում են այդ ուսումնասիրությունների հիմման վրա, կամ դեմքի հայտնաբերումը նկարի մեջ։

Ուսումնասիրվեցին բազմաթիվ պատկերների հետ աշխատելու ալգորիթմներ որոնք հնարավորություն կտան հետագայում առանձնացնել օբյեկտները պատկերների վրայից։

**Գրականություն**

* Digital Image Processing 3rd ed. - R. Gonzalez, R. Woods
* Computer and Machine Vision 4e - Theory, Algorithms, Practicalities By E R Davies
* Solomon C., Breckon T. Fundamentals of Digital Image Processing.. A Practical Approach with Examples in Matlab (Wiley-Blackwell, 2011)
* Computer Vision: Algorithms and Applications Richard Szeliski (2010)