Optical Music Recognition

Emil Brissman (emibr948)
Dan Englesson (danen344)
Tina Durmén Blunt (tindu519)

Linköpings Universitet, Campus Norrköping 2010-10-17.

Sammanfattning

Att kunna översätta innehåll i bilder till andra format, så som text eller ljud, gör så att informationen kan bearbetas och användas till mycket mer. Noter är ett skrivspråk som inte överförs med vanliga textsymboler. Att använda sig av mönsterigenkänning i samband med tolkandet av notblad är väsentligt eftersom det, om inga anpassade program används, består av bilder. Följande beskriver ett sätt att ta sig från ett notblad till en textsträng med noter via optisk musikigenkänning.

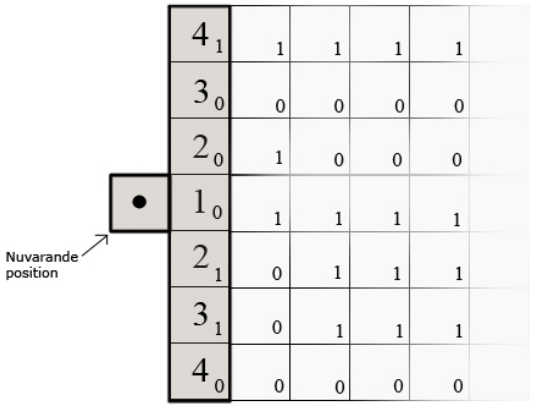
1. Inledning

Automatisk igenkänning av musikaliska noter på så väl scannade pappersark till noter skriva för hand är ett komplext problem att lösa. Detta mest på grund av alla slags symboler som kan finnas i ett notark men också hur bra originalbilden är. För att digitalisera bilden och sedan utföra OMR, som står för Optical Music Recognition (optisk musikigenkänning), kan ske genom scanning av noterna eller fotografering med en digital kamera. Det finns många vägar att gå för att läsa ut noter ur ett notblad. Detta arbete bygger dock på två olika sätt att ta sig till det slutgiltiga målet, som är en textsträng med lästa fjärdedelsnoter och åttondelsnoter ur ett blad av noter.

2. Metod

2.1 Förbehandling

Förbehandlingen har två versioner av förklarliga skäl som kommer att diskuteras under diskussion. Det som skiljer de båda åt är hur bilden transformeras, det vill säga i förbehandlingen av bilden. Analysdelen är den samma. I ena versionen utförs bara rotation utifrån den bästa linjematchningen från radontransform. Den andra versionen bygger på ”stable paths” [2] (säker väg) för att först detektera notlinjer i ett första steg. Detta för att få punkter på linjerna, bland annat start- och slutpunkt, för att sedan utföra en geometrisk korrektion på bilden. Bilden som är förstärkt med ett laplacefilter och trösklad söks igenom med en sökarea enligt figur 1 för att finna notlinjerna.

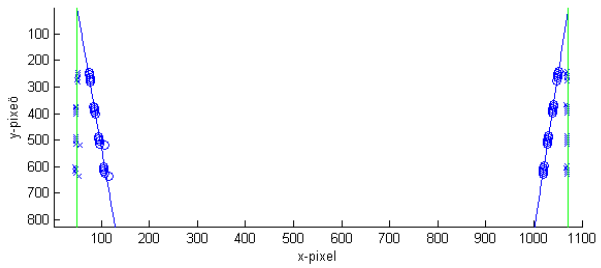


Figur 1. Sökarea som går över bilden. De små siffrorna representerar bilden, de stora siffrorna respresenterar kostnaden.

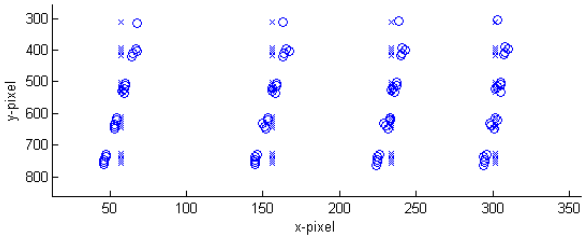
Storleken är sju rader och en kolumn på sökarean. Sökarean är viktad och representerar kostnaden för att gå till nästa pixel från nuvarande position. I situationer då det är möjligt att gå snett nedåt eller snett uppåt, med en kostnad på tillexempel två, slumpas det fram om algoritmen ska gå snett neråt eller uppåt. Ett bitplan håller reda på vart sökalgoritmen har varit och om den funna linjens längd är längre än bildens längd gånger en faktor adderas linjen till ett resultatplan.

Innan den geometriska korrektionen beräknas nya koordinater utifrån de gamla koordinaterna genom att rotera de funna notlinjernas ändpunkter, se figur 2. Vinkeln erhålls genom att först beräkna ut en rät linje med minsta kvadratmetoden genom startpunkter respektive slutpunkter. Vinkeln mellan

den blåa linjen och den gröna räta linjen i figur 2 kan enkelt räknas ut med trigonometri. Koordinaternas x-värden läggs på samma x-värde för varje rad med koordinater. Figur 3 visar förhållandet mellan gamla koordinater (runda) och de nya (kryss).



Figur 2. Koordinaterna roteras till att bli raka, den gröna linjen.



Figur 3. Efter korrigering av de nya koordinaterna i både y-led och x-led fås följande förhållande mellan gamla och nya koordinater.

Det som är gemensamt för båda versionerna är att en ”tophat” transformation utförs [5]. Denna transformation innebär en erodering av bilden med ett cirkulärt objekt. Bildens bakgrund erhålls och subtraheras från originalbilden vilket gör att belysningsgradienter försvinner från bilden och blir lättare att tröskla. Gemensamt är också klippning av bilden i både x-led och i y-led. Horisontella och vertikala projektioner av en trösklad version av bilden är utgångspunkten för denna metod. De erhållna endimensionella signalerna itereras igenom från vänster och från höger. Då signalvärdet är över en viss tröskel stoppas sökningen och värden i x-led och y-led erhålls som anger hur bilden ska klippas.

Bilden som fås ut av båda versioner trösklas och används senare i analysdelen. Innan analysdelen och evalueringen av vilken tonhöjd noten har måste notlinjerna detekteras. Eftersom vår implementerade ”stable paths” metod enbart används till den geometriska korrektionen innebär att notlinjerna fortfarande måste detekteras. Att ”stable paths” inte kunde användas till detekteringen av notlinjerna kommer att diskuteras under diskussion.

2.2 Detektera notlinjerna

Notlinjerna erhålls genom en horisontell projektion [4] av en förstärkt bild genom filtrering med ett laplacefilter för att ge en mer stabilare projektion. Bilden eroderas även med ett linjeelement som enbart tar ut linjerna, vilket gör projektionen ännu lättare. Från den erhållna signalen används Matlabs funktion ”findpeaks” [1] för att hitta alla maximum i signalen. Dessa representerar notlinjernas positioner. Om antalet notlinjer inte är delbart med fem kommer programmet att avslutas. Annars läggs ytterligare tre notlinjer till för varje notplan för att kunna detektera ytterligare noter som inte ligger inom notplanet. Ett värde på hur stort mellanrummet mellan notlinjerna är räknas ut för att användas i analysdelen som beskrivs nedan.

Sammanfattnings vis får analysdelen en trösklad bild, ett värde på mellanrummet mellan notlinjerna och notlinjernas positioner från förbehandlingsdelen av programmet. Kommande beskriver de analysmetoder som används.

2.3 Mönsterigenkänning och klassificering

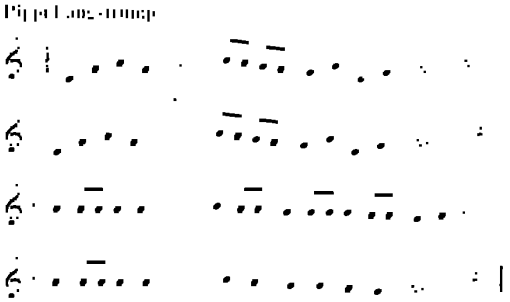
Värdet på mellanrummet mellan notlinjerna används för att uppskatta ett värde på nothuvudenas storlek och på så sätt kunna anpassa strukturelementen för varje enskild bild efter det.

För att klassificera alla objekt i bilden och antingen ta bort de som är ointressanta eller vidare behandla de noter som ska vara med utskriften, används principen att urskilja de intressanta delarna som är avgörande i vad varje objekt har för egenskap. Balkarna, prickarna, flaggorna och notskaften separeras i varsin bild för att sedan analyseras var för sig. Tillexempel innehåller en fjärdedelsnot, se figur 4, ett nothuvud och ett notskaft, medan två ihopkopplade åttondelsnoter, se figur 5, innehåller två nothuvuden, två notskaft och en notbalk.

Figur 4. Fjärdedelsnot

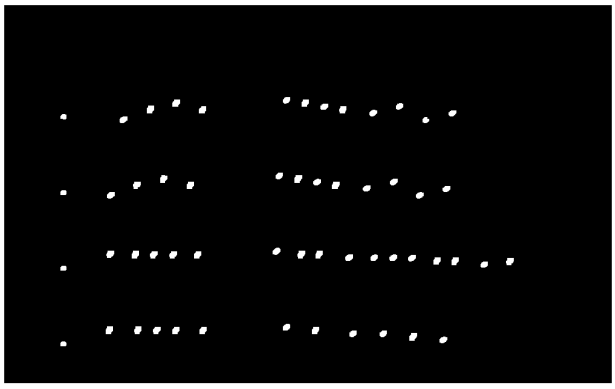
Figur 5. Två åttondelsnoter

En stängning på bilden utförs med ett relativt stort strukturelement vilket eliminerar alla tunna linjer i bilden och lämnar övriga objekt, se figur 6. Bilden normaliseras och trösklas för att få bort eventuellt brus.

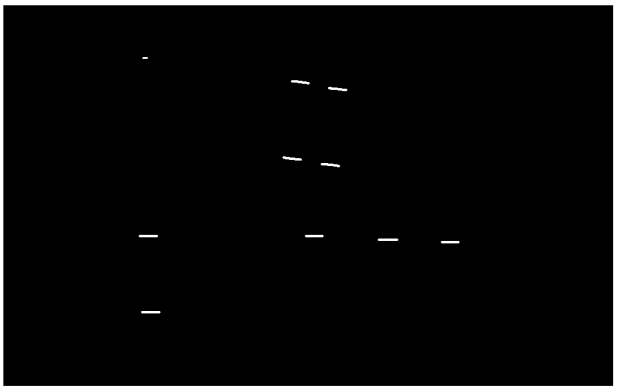


Figur 6. Urskiljning av prickar och balkar.

Alla objekt som är kvar i bilden får en märkning (label) som sedan omringas av en omslutningsarea (bounding box). Varje bounding box area och förhållande mellan längd och höjd på denna används för att urskilja vilken typ av objekt som omges av bounding boxen. Prickarna och balkarna urskiljs och separeras till två olika bilder för att underlätta klassificeringen av de hela objekten, se figur 7 och 8.



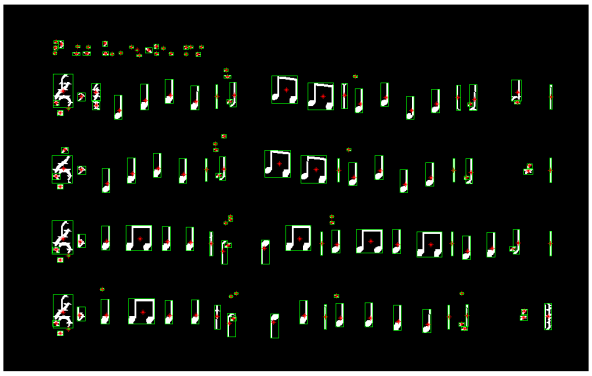
Figur 7. Separerade Prickar



Figur 8. Separerade balkar

För att få ut flaggorna på ensamma åttondelsnoter stängs bilden med två olika strukturelement med lutning åt två olika håll. Denna bild får med väldigt många felaktiga objekt som enkelt kan tas bort genom att subtrahera med både balkbilden och bilden med prickarna. På samma sätt urskiljs notskaften med enda skillnaden att strukturelementet är vertikalt.

De fyra bilderna läggs sedan ihop för att få fram en bild med enbart de önskade notdelarna. Alla objekt får en label och omsluts av varsin bounding box som visas i figur 9.

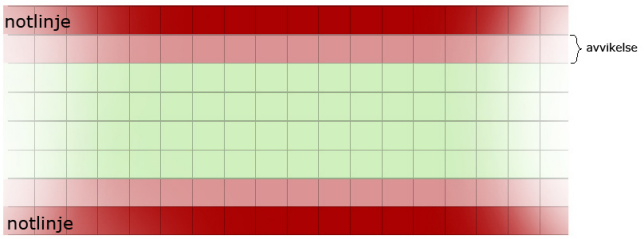


Figur 9. Objekt med bounding box.

Bounding boxen representerar nu de områden där ett möjligt notobjekt ligger. Dessa områden undersöks var för sig i alla de fyra bilder där de objekt som ligger inom området räknas i varje bild. Objekten kan då klassificeras enligt principen ovan och färgläggs eller tas bort, beroende på vilken typ den passar in på.

2.4 Tonhöjdsbestämmning

För att bestämma varje nots position i bilden krymps först varje nothuvud ner till en pixel. Tonhöjden för varje not kan bestämmas med pixelns position i y-led och ett avstånd från närmaste notlinje. Eftersom notlinjernas positioner enbart anger var den ligger i y-led är det en god idé att införa en avvikelse inom vilket notlinjen fortfarande är en notlinje. I figur 10 visas notlinjen som helt röd. Men eftersom man kan tänka sig att notlinjen har en tjocklek adderas en avvikelse för att definiera notlinjen som lite tjockare. Detta innebär att notlinjen nu är både det röda och ljusröda området i figur 10. Beroende på var notens y-värde ligger i förhållande enligt figur 10 kan tonhöjden bestämmas. Om y-värdet för noten ligger i det gröna området i figur 10 ligger noten mittemellan två notlinjer.

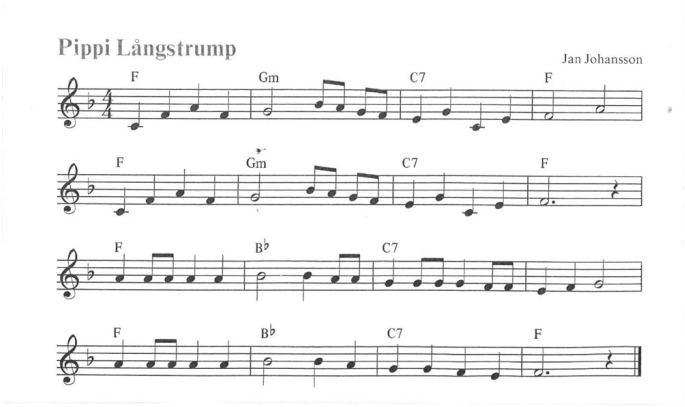


Figur 10. Tonhöjdsbestämmning. Noten kan antingen ligga på den övre eller undre notlinjen eller mitt i mellan dessa.

Innan tonhöjden bestäms måste dock alla noter sorteras i ordningen de ligger i bilden för att få den önskade utsträngen, vilket kan göras genom att sortera dem i x-led och ange vilken notgrupp varje not tillhör. Färgen på varje not avgör om noten är en fjärdedel- eller åttondelsnot.

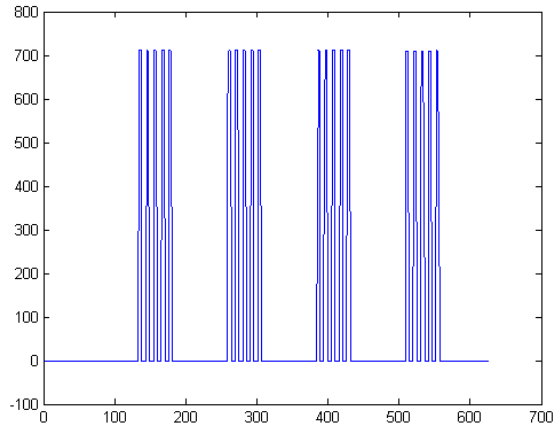
3. Resultat

3.1 Körexempel



Figur11. Original bild.

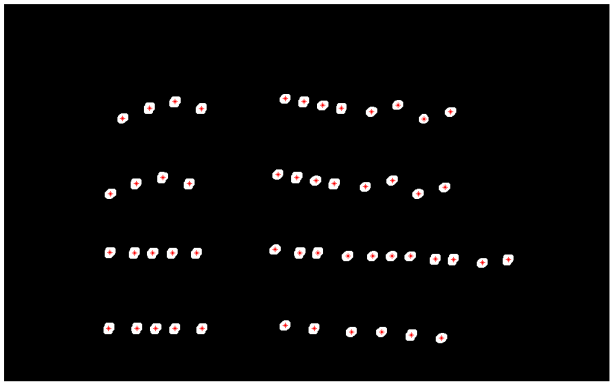
Resultat av horisontell projektionen, figur 12, med tröskling av projektionen för att detektera linjernas position från originalbilden, se figur 11.



Figur12. Trösklad horisontell projektion

Alla notdelar (nothuvud, notbalk och notskaft) är räknade för varje separerat objekt i bilden och objekten färgsätts efter vilken typ av not det är.

Pixeln i mitten av varje nothuvud, utritat i rött i figur 13, vars position används för att bestämma tonhöjden på varje enskild not.



Figur13. Nothuvudenas masscentrum används för att avgöra tonhöjden.

Utskriften för bilden ovan blev:
C2F2A2F2b2a2g2f2E2G2C2E2n
C2F2A2F2b2a2g2f2E2G2C2E2n
A2a2a2A2A2B2a2a2G2g2g2G2f2f2E2F2n
A2a2a2A2A2B2A2G2G2F2E2

3.2 Utdata

Programmet klarade på ett bra sätt av att ta ut information från inscannade bilder, då simpla rotationer och minimal korrektion av belysningen behövdes göras. Dessa bilder höll hög kvalitet på utdatan och gav bara enstaka fel ibland. Bilder fotade rakt uppifrån klarade programmet av också, även de med ojämn belysning. Felfaktorn ökade dock en del med de fotade bilderna, men inte allt för mycket. Version två klarade av att genomföra en någorlunda bra geometrisk korrektion på fotade bilder och gav ett sämre resultat av utadat.

4. Diskussion

4.1 Förbehandling

Eftersom den geometriska korrektionen tillsammans med ”stabel pats” metoden inte alltid gav korrekta resultat valdes att dela upp programmet i två delar, en version utan det nämnda ovan (version ett) och en version med (version två).

Vid en närmare undersökning av ”stable paths” metoden som implementerades, så hittar den inte alltid alla linjer. Detta kan bland annat bero på den slump som är inbyggd då funktionen ska välja att gå snett uppåt eller snett nedåt från nuvarande position - valet kostar lika mycket. Det kan då förkomma situationer som gör så att metoden kommer in på en annan notlinje eller nottak och på så vis får fel slutpunkt.

Detta visar sig sedan i en felaktig minsta kvadratmetod och geometrisk korrektion. Även en dålig bild kan resultera i färre funna notlinjer och det blir svårare för ”stable paths” metoden att hitta dessa. Den implementerade metoden är inte stabil men hade kanske kunnat bli om mer tid hade varit till förfogande.

Ett annat stort problem i förbehandlingen var att efter den horisontella projektionen, som gav en endimensionell signal, hitta alla maximum i signalen. Dessa representerar, som nämnts i metoden ovan, notlinjernas positioner. Här använde vi matlabs inbyggda funktion ”findpeaks” för att hitta alla maximum.

Om signalen har ett maximum som är av samma värde i mer än ett steg i x-led hittas inget maximum för den toppen i signalen. Så blir även om toppen ser ut som ett m tillexempel. Då finns två maximum för den toppen och antalet notlinjer är inte längre delbart med fem. En lösning skrevs ihop för att enbart få ett maximum för varje topp. Dock är lösningen väldigt enkel och kanske inte fungerar för alla situationer på signalen.

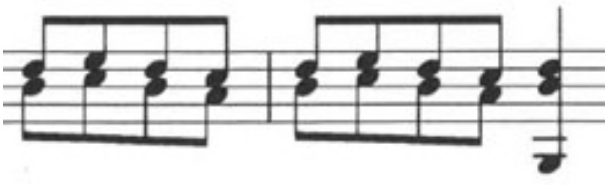
Den metod som beskrivs av Dzenan Kapidiz [4] med derivering av projektionen i horisontalled valdes bort för att den på ett sätt gav samma sak som projektionen. Dock ger deriveringen tjockleken på notlinjen men det måttet ansågs vara samma som att sätta en avvikelse som gav i princip samma resultat.

4.2 Analys

Under projektets gång uppkom många problem i analysdelen. Ett av dem var att ljusintensiteten varierade markant från bild till bild. Detta försvårade trösklingen och påverkade hur pass bra analysen kunde göras på bilden. Problemet kvarstår men är någorlunda stabil på de testbilder vi hade till förfogande.

Då objekt inte är separerade i originalbilden försvåras urskiljningen av objekten i den metod som användes. Om två objekt sitter ihop från början

räknas de som ett objekt och får endast en bounding box, vilket gör så att de aldrig kan klassificeras som den typ av not de ska vara. På ett par av de bilder som användes som träningsunderlag var ett antal nothuvuden sammankopplade som skulle klassificeras som två olika objekt, se figur 14.



Figur 14. Sammankopplade nothuvuden

Problemet löstes genom att kontrollera arean av varje bounding box som omger nothuvudena och hur stor del av den som var täckt av objektet. Genom att använda sig av det och även förhållandet mellan längden och höjden på bounding boxen är det möjligt att avgöra om det ligger två nothuvuden inom en bounding box och då dela objektet i mitten.

Överlappande bounding boxar var även ett stort problem då det orsakar att räkningen av notdelar blir inkorrekt. Om en notdel ligger i två olika bounding boxar kommer det att räknas med i klassificeringen av objekten i båda boxarna. Om till exempel ett nothuvud kommer med i en bounding box den inte ska tillhöra, räknas den med i objektets klassificering och typen för noten kommer med största sannolikhet att bli felaktig. Ingen ultimat lösning till problemet hittades utan objekten fick anpassas så väl som möjligt för att inte skapa överlappande bounding boxar. Detta gjordes med diverse olika morfologiska operationer på bilden för att krympa och utvidga objekten.

Då linjerna i bilden är krökt i ändarna skapar detta problem. Radontransformen som roterar linjerna tar inte hänsyn till om linjerna är böjda och det blir problem senare då ändarna inte har samma position i y-led som resten av linjen. Noterna som ligger i dessa områden får då fel tonhöjd eftersom de utgår ifrån den linjepositionen som hittades i den vertikala projektionen beskrivet ovan. Med hjälp av geometrisk korrektion löstes detta problem men bara på vissa bilder.

Eftersom alla objekt skiljer i storlek från bild till bild skapas problem med att sätta en korrekt storlek på strukturelementen i varje bild. Är strukturelementet för litet tas oönskade objekt med och är det för stort rensas bilden för mycket och nödvändig information försvinner. Detta löstes genom att sätta strukturelementens storlek i förhållande till avståndet mellan två notrader i bilden. Även om objekten skiljer sig i storlek från bild till bild så är de alltid lika stora i förhållande till varandra i varje bild.

När tonhöjden ska evalueras görs detta utifrån vart tonen ligger i höjdled mellan två notlinjer. Funktionen som gör detta fungerar och gör det den ska. Problem uppstår dock om nothuvudets masspunkt har centrerats fel. Vad som kan göras åt det här är svårt att säga. En bra utgångs punkt kan vara att centrera ett helt runt nothuvud från början för att på så sätt få ett mer exakt masspunktscentrum.

4.3 Alternativa metoder

Problemet med notbladen går att lösa på väldigt många olika sätt. En alternativ metod som berördes men valdes bort var att använda sig av projektioner inom varje bounding box. Varje objekt i bilden får då två grafer som representerar dess projektion i både x- och y-led. Dessa kan jämföras med grafer från redan specificerade noter. Metoden ”K-nearest neighbor” [3] kan användas för att avgöra vilken typ varje objekt ligger närmst. Alltså vilken av de redan specificerade projektionsgraferna som det okända objektet är mest lik.
Om denna metod ska användas krävs att skalningen på varje bounding box stämmer mycket väl överrens med storleken på den specificerade noten. Jämförelsen är även mycket känslig för små ändringar i objektet som inte stämmer överrens med utseendet på den specificerade noten. Brus i bounding boxen eller små skillnader i lutningen på balkarna är saker som kan skapa stora problem.

Ytterligare en metod vi valde bort var att använda sig av korrelation. Mest för att beräkningarna skulle ta lång tid och alla möjliga notmallar som var tvungna att skapas.

4.4 Övrigt

Vi är nöjda med vårt arbete även om tiden var knapp och vi var tvungna att dela upp programmet i två versioner. På grund av tidsbegränsningen hann vi inte heller med andra förbehandlingar av bilden som brus och suddighet. Resultatet har en viss faktor med fel men fungerar ändå relativt bra som kan ses under reslutat ovan.

5. Referenser

[1] Matlab Help, Image Processing Toolbox.

[2] IEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 31 NO. 6, JUNE 2009, p. 1134-1139.

[3] Optical recognition of music symbols, A comparative study, R. Rebelo G. Capela Jaime S. Cardoso.

[4] Automatic Optical Music Recognition of printed music, Dzenan Kapidzic, Norrköping 2003-10-10.

[5] Digital Image Processing, Third Edition, Rafael C. Gonzalez Richard E. Woods.