Utveckling av filterbank för fågelklassificering

Sara Elmgart, Albin Franzen, Linnea Sartorius och Theo Tatidis

Lund, 23 februari 2024

Sammanfattning

I den här rapporten har vi utvecklat en metod för att skilja på bofinkar, gråsparvar och talgoxar genom att analysera dess fågelkvitter i tids- och frekvensdomänen. I tidigare studier har Mel-filterbanken använts för detta, men vi ämnade att undersöka om en mer specifik filterbank, utformad efter just de tre nämnda fåglarna, lämpar sig bättre. Vår metod konstruerades genom att identifiera stavelser av fågelkvitter och mellanrum mellan dessa, för att sedan skapa en genomsnittlig stavelse för varje art. Därefter utvecklade vi en filterbank som lägger störst vikt vid de frekvenser som skiljde sig mest åt mellan arterna, för att sedan applicera det på genomsnittsstavelserna och de okända stavelser vi önskade identifiera. Sedan används principalkomponentsanalys (PCA), Naive Bayes klassificerare och ett Matchat filter för att avgöra vilken art fågeln troligen är. Slutligen utvärderas metodens tillförlitlighet jämfört med om Mel-filter hade använts.

Innehåll

1 Teori

Tidigare arbeten inom området har använt klassificeringsmetoder såsom MFCC (*Mel Frequency Cepstrum Coefficients*) [1] och SVD (*Singular Value Decomposition*) [2]. MFCC är en ljudrepresentation där en signal Fouriertransformeras, för att sedan göras om till *MEL-skala* genom att använda triangulära överlappande fönster.

1.1 Effektspektrumskattning

Alla signaler i tid $x(t)$ kan även studeras i frekvensdomänen. Varje frekvens kommer ha olika effekt och det spektrum som visar hur effekten är fördelad över frekvenserna kallas för effektspektrum. Ett periodogram skattar effektspektrumet för en given signal utifrån en Fouriertransformen $X(f)$

$$|X(f)|^2 = |\sum x_N[n]e^{-i2\pi nf}|^2$$

där N är antalet punkter i signalen och n är tidsdiskreta sampel. För att minska variansen i periodogramskattningen kan istället Welch-metoden användas. Welch-metoden delar upp en signal i överlappande fönster och multiplicerar varje fönster med en funktion som avtar kring änderna för att minska *spektral leakage*. Spektral leakage är att frekvenser som uppstår från den skarpa kanten på fönstret skapar ett missvisande effektspektrum. En vanlig fönsterfunktion är Hanning-fönstret som är en skalad sin^2 funktion.

1.2 Filtrering

Inom digital signalbehandling är filter på signaler ett sätt att ändra signaler utifrån deras frekvens. Faltningssatsen säger att $f(n) * g(n) = \mathcal{F}^{-1}(F(f)G(f))$ som innebär att faltning i tidsdomän mellan två funktioner motsvarar en multiplikation i frekvensdomän för den funktionen. Om vi ska applicera ett filter på insignalen räcker det alltså att beräkna filtrets Fouriertransform för att sedan multiplicera det med insignalen.

Det finns många olika användbara filter och ett som används ofta är ett moving-average filter (MA-filter) och framöverallt det enkla MA-filtret som är en faltning med en konstant kort vektor som $m(n) = 1/5[1\,1\,\underline{1}\,1\,1]$ och används för att jämna ut kurvor. Det ses i formeln för faltningen

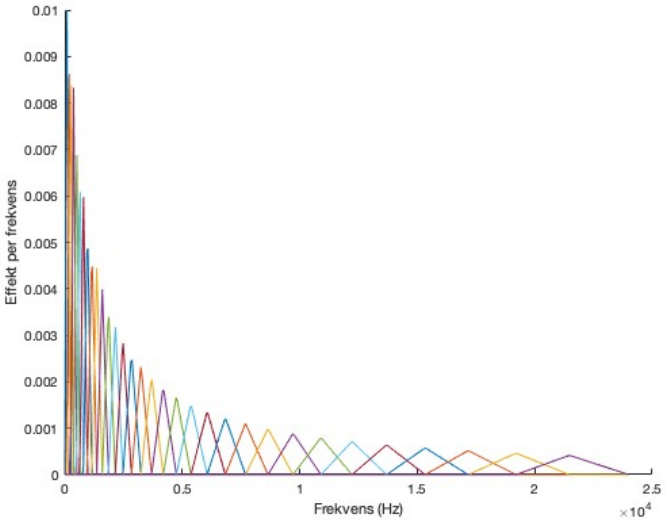
$$\text{Filtered } x(n) = 1/5[1\,1\,\underline{1}\,1\,1]x * x(n) = 1/5\Big(x(n-2) + x(n-1) + x(n) + x(n+1) + x(n+2)\Big)$$

Andra användbara filter är högpassfilter som är 0 under en viss frekvens och 1 över, vilket låter höga frekvenser passera genom filtret. Lågpassfilter fungerar på liknande sätt, fast det är de låga frekvenserna som blir kvar. Bandpassfilter är en kombination av hög- och lågpass filter som bara släpper genom frekvenser inom ett visst band. Om detta bandet är triangelformat i stället för rektangelformat har vi ett triangelfilter som prioriterar frekvenser nära mitten av bandet. Många filter tillsammans kallar för en filterbank och används inom bland annat Mel-spektrumet.

1.3 Melspektrum

Vid hantering av stora mängder data vill man hitta sätt att koka ner den viktiga informationen till färre antal datapunkter utan att förlora viktig information. Då krävs mycket mindre processorkraft och tar kortare tid för att exempelvis köra klassificeringsalgoritmer eller filter på datamängden.[3]

För ljudfiler är Mel-skalan ett sätt att reducera mängden data. Teorin bakom Mel-skalan bygger på att det mänskliga örat är olika bra på att skilja på två toner, beroende på vad tonerna har för frekvens. Frekvenser på 100Hz och 200Hz har vi inga problem att skilja på, medan vi tycker 5000Hz och 5100Hz låter likadant. Inom ljudbehandling, till exempel röstigenkänning genom maskininlärning, vill man ofta efterlikna det mänskliga örat. Gör man en fourieranalys på en ljudfil kommer man få ett lika stort intervall (i Hertz) mellan alla datapunkter på frekvensaxeln. Vill man efterlikna det mänskliga örats känslighet är det dock inte så viktigt att ha lika detaljerad information kring diskanttonerna som bastonerna. Man vill ha kvar detaljnivån i basen men det gör inget att göra exempelvis medelvärden bland diskanttonerna. I figur 1 visualiseras hur en Mel filterbank ser ut. Det är helt enkelt en summa av triangelfilter, där basen av trianglarna blir bredare ju högre frekvens man studerar.



Figur 1: Triangulära fönstren för mel filterbank.

1.4 Korskorrelation

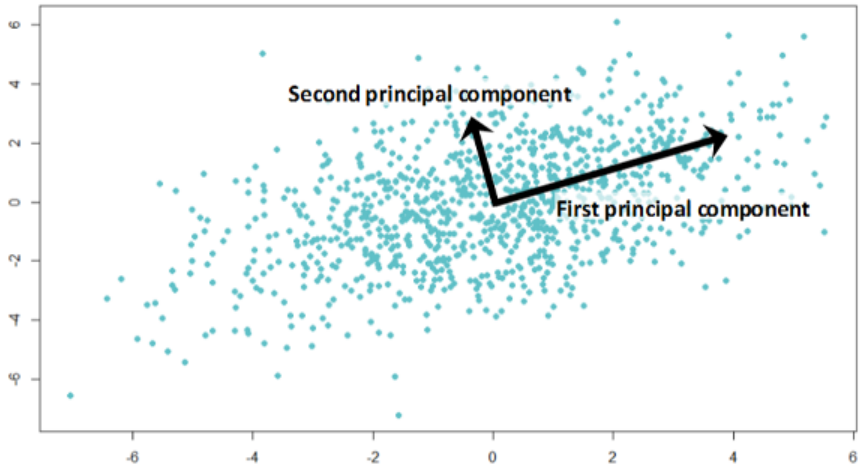
Korskorrelation är ett sätt att detektera hur lika två signaler är, eller om en signal finns i en annan signal. Detta görs genom att falta de två signalerna åt samma håll för att se om korrelationen blir stor. Matematiskt kan korskorrelation uttryckas som

$$f(n) * g(-n) = \sum_{m=-\infty}^{\infty} f(m)g(n+m)$$

Denna faltning returnerar en vektor med värden mellan -1 och 1. Varje värde i vektorn representerar en elementvis multiplikation mellan signalerna för en viss "tidsskillnad" där värden närmare -1 betyder att signalerna inte alls är lika och om outputvektorns värde är 1 så är signalerna precis samma. För just detta projekt är det endast maxvärdet på faltningen som är relevant eftersom vi klassificerar endast beroende på hur lika två signaler är och inte utifrån när två signaler är som mest lika.

Ibland kallas även korskorrelationen för ett matchat filter om det sker inom en signal där den sökta signalen gömmer sig datan. Då är samma signal som den man söker det optimala filtret.

1.5 Principalkomponentanalys (PCA)



Figur 2: Principalkomponenter visualiserade för slumpmässig data [4]

Principalkomponentanalys (PCA) är ett sätt att linjärt projicera högdimensionell data till lägre dimensioner på ett sätt som behåller så mycket information som möjligt. I figur 2 kan det ses att tvådimensionell data reduceras ner till två vektorer som kallas för principalkomponenterna. Den första principalkomponeneten beskriver den högdimensionella datan bäst vilket är ekvivalent med att den maximerar variansen hos projektionen av datan på vektorn.

Principalkomponentanalys kräver att datan är centrerad i origo och slutprodukten blir att principalkomponenterna är normerade. PCA kan användas till att förenkla högdimensionell data för visualisering eller ta bort onödiga delar av datan. PCA ger även insikt till hur datan är fördelat förutsatt att det är linjärt.

1.6 Naive Bayes klassificering

Naive Bayes klassificering är en metod som statistiskt urskiljer vilken klass en observation tillhör med antagande att klasserna är normalfördelade. Metoden kallas naiv eftersom prediktorerna anses vara oberoende av varandra som gör att varje flerdimensionell normalfördelning kommer vara rät i förhållande till varandra. Detta gör modellen enklare än en kvadradiskt diskriminant analys med beroende variabler men lättare att första implementeringen. Sannolikheten att en observation x tillhör en viss klass C_k givet antagendet att klassen är är normalfördelad med medelvärde μ_k och standard σ_k blir därmed

$$P(x|C_k) \in N(\mu_k, \sigma_k)$$

Med Bayes sats kan dessa sannolikheter kombineras så att sannolikheten att en observation tillgör en viss klass är

$$P(C_k|x) = \frac{P(C_k)P(x|C_k)}{P(x)}$$

Där $P(C_k)$ kallas priori-fördelningen och $P(C_k|x)$ kallas posteriori-fördelningen. $P(x)$ kommer vara konstant om det är slumpmässigt vilken observation som används för klassificeringen.

1.7 Problemformulering

Målet med detta projekt var att skapa ett program som har en ljudfil med fågelkvitter som insignal och en slutsats om vilken fågelart det var som utsignal. I projektet begränsade vi oss till att studera bofink, gråsparv och talgoxe.

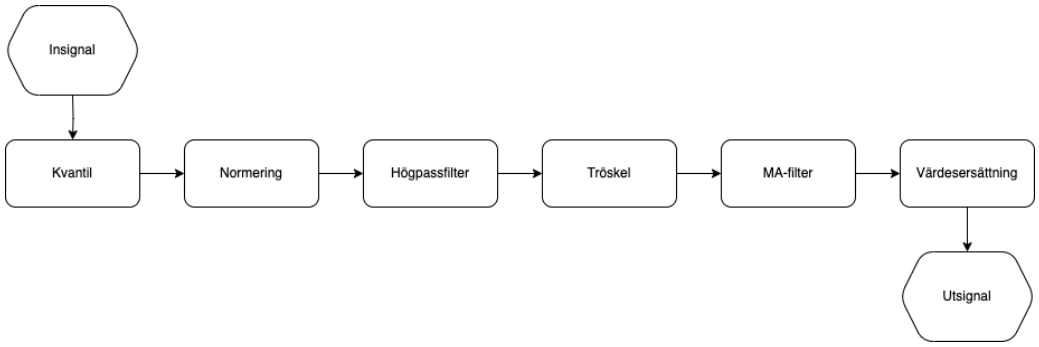
Inom signalbehandling används ofta Mel-skalan för att vikta utefter vilka frekvenser som är av intresse. Eftersom Mel-skalan bygger på den mänskliga hörseln kommer metoden alltså att lägga större vikt vid de frekvenser människan kan urskilja bra. I detta arbete var vårt mål att skapa en bättre viktning än Mel för just detta program. Mel bygger ju på det mänskliga örats funktion och det är ju inte alls säkert att detta är det bästa sättet att skilja mellan fåglar.

2 Metod

För att kunna utveckla en bättre filterbank än mel skalan för fågelsångsklassificering börjar vi med att förarbeta ljudfilerna genom att klippa ut var stavelserna i varje fågel ligger. Därefter designar vi en klassificeringsalgoritm som både arbetar i tid och frekvens. Frekvensanalysen görs först med Mel-skalan, därefter skapades en egen filterbank som förhoppningsvis presterar bättre än Mel-filterbanken. Tidsanalysen är viktig eftersom den kan sätta i perspektiv hur viktig filterbankutvecklingen är och vilken roll ett effektspektrum har till skillnad från längden av de olika fågelarternas stavelser. Sista steget är att testa vår filterbank på data och jämföra utan filter och med mel-filterbanken.

2.1 Stavelse och mellanrumssegmentering

Det första steget för att kunna klassificera fågelarter utifrån deras sång är att segmentera ljudfilerna till stavelser och mellanrum. Vi önskar att programmet kan ha samma parametrar som input oberoende av vilken ljudfil det ska försöka identifiera. Detta lyckades vi med genom att skapa en tröskelbaserad segmenteringsalgoritm. Först sattes alla värden över en viss kvantil till kvantilgränsen. Det gjordes som ett sätt att skala högenergiområdena. Därefter normeras signalen med min-max skalning från noll till ett följt av ett högpassfilter som tar bort de låga frekvenserna, som vind i bakgrunden, men behåller de höga fågelsångsfrekvenserna. Nästa steg är en tröskel där endast amplituder över tröskeln behålls och sedan används ett enkelt MA-filter för detektera var någonstans det endast finns nollor. Där filtret är nollskilt finns en signal, annars är det ett mellanrum. Hela denna process visualiseras i figur 3.



Figur 3: Blockdiagram för stavelsesegmenteringsalgoritmen

2.2 Tidsdomänsanalys

För att kunna analysera de segmenterade ljudfilerna i tidsdomänen kommer vi att försöka hitta viktiga prediktorer från stavelselängderna i segmenteringen till en modell som kan använda Naive Bayes för att klassificera. Detta kommer innebära mycket prövning för att se om exempelvis genomsnitt, median, viktat genomsnitt med mera på stavelselängder är mest relevanta, och ifall mellanrummet mellan stavelser är viktiga. En annan viktig aspekt är att testa med olika sammanfogningsavstånd från stavelsesegmentering för att se om hög- eller lågfrekventa stavelser skiljer fågelarterna mest. När några prediktorer har funnits kan PCA användas för att extrahera viktig information och skapa mer förklarliga modeller i lägre dimensioner.

2.3 Frekvensdomänsanalys

När inte tidsanalysen räcker till kommer frekvensdomänen användas för att stärka klassificeringen i tid. Frekvenserna i signalen kommer att skattas med Welch-metoden och sedan kommer ett matchat filter som använder

korskorrelation att avgöra vilken fågelart som mest liknar signalen. Alltså för varje ny ljudfil kommer vi att falta med ett tidigare skapat genomsnitt för fåglarna och den korskorrelationen som blir störst motsvarar vilken fågel som signalen mest troligen sjungs av.

2.4 Filterdesign för förbättrad frekvensanalys

För att förbättra den analys som görs i frekvensdomänen kommer vi först applicera en filterbank på signalen innan analysen görs. Utveckling av denna filterbank tar inspiration från Mel-filterbanken som består av en samling överlappande triangulära filter med en höjd beroende på hur viktig en viss frekvens är. För att finna vilka frekvenser är viktigast i vår modell utgår vi ifrån insignalen $x(n) \xrightarrow{\mathcal{F}} X(f)$ som har effekten $P(f) = |X(f)|^2$. Vår klassificeringsmodell är att vi skapar en genomsnittlig effektskattning för varje fågel $g_1(f), g_2(f), g_3(f)$ och väljer den fågeln där $\max(g_i(f) * P(f))$ är störst. De viktigaste frekvenserna är de som kommer maximera avståndet mellan $\max(g_i(f) * P(f))$ alltså de frekvenser som gör

$$d\bigg(\max(g_1(f) * P(f)), \max(g_2(f) * P(f)), \max(g_3(f) * P(f))\bigg) \tag{1}$$

så stort som möjligt, och alltså där signalerna skiljer sig åt som mest. Vi gör ett antagande att det maximala värdet av faltningen sker vid $f/2$, vilket innebär att $\max(g_i(f) * P(f)) = \sum X(f)g_i(f)$. Vi inför även en "metrik" med 3 ingångsvärden och väljer avståndsfunktionen $d(a, b, c) = ||a - b|| + ||a - c|| + ||b - c||$ som är inspirerat av omkretsen till en triangel i det Euklidiska rummet. Med dessa förenklingar och valet av absolutbeloppsnormen får vi

$$d\bigg(\max(g_1(f) * P(f)), \max(g_2(f) * P(f)), \max(g_3(f) * P(f))\bigg) =$$

$$= d\bigg(\sum P(f)g_1(f), \sum P(f)g_2(f), \sum P(f)g_3(f)\bigg) =$$

$$= ||\sum P(f)g_1(f) - \sum P(f)g_2(f)|| + ||\sum P(f)g_1(f) - \sum P(f)g_3(f)|| + ||\sum P(f)g_2(f) - \sum P(f)g_3(f)|| =$$

$$= ||\sum P(f)(g_1(f) - g_2(f))|| + ||\sum P(f)(g_1(f) - g_3(f))|| + ||\sum P(f)(g_2(f) - g_3(f))|| =$$

$$= \sum P(f)(|g_1(f) - g_2(f)| + |g_1(f) - g_3(f)| + |g_2(f) - g_3(f)|)$$

som använder att $P(f) \geq 0$ för att flytta in absolutbeloppet i summan. Utifrån det sista uttrycket framgår det att vikten av en frekvens är

$$w(f) = |g_1(f) - g_2(f)| + |g_1(f) - g_3(f)| + |g_2(f) - g_3(f)| \tag{2}$$

Funktionerna $g_1(f), g_2(f), g_3(f)$ är kända vilket medför att $w(f)$ kan plottas för att visuellt se vilka frekvenser som är viktigast för fågelklassificering. Nästa steg är att sampla $w(f)$ och skapa överlappande triangulära fönster där bredden är invers proportionellt mot energin hos frekvenserna den innehåller med en viss skalningsfaktor.

2.5 Testning av vårt program

När vi kände oss färdiga med vårt program var det dags att testa det. Som tidigare konstaterats utgick vårt program från ett dataset på ca $N = 60$ ljudfiler. Totalt nio av dessa, tre från varje fågelart, valdes dock slumpvis ut och sparades ner separat. Dessa har alltså inte använts till att skapa modellen för programmet, utan istället agerat slutgiltig testdata. Först testade vi dock programmet på våra 51 ljudfiler som agerat övningsdata, för att få en första indikation på hur programmet presterade. Vi testade programmet med tre olika filter i frekvensdomänen: (1) inget triangelfilter, det vill säga endast Welch-metoden, (2) Mel-filterbanken och (3) vår egenkonstruerade filterbank.

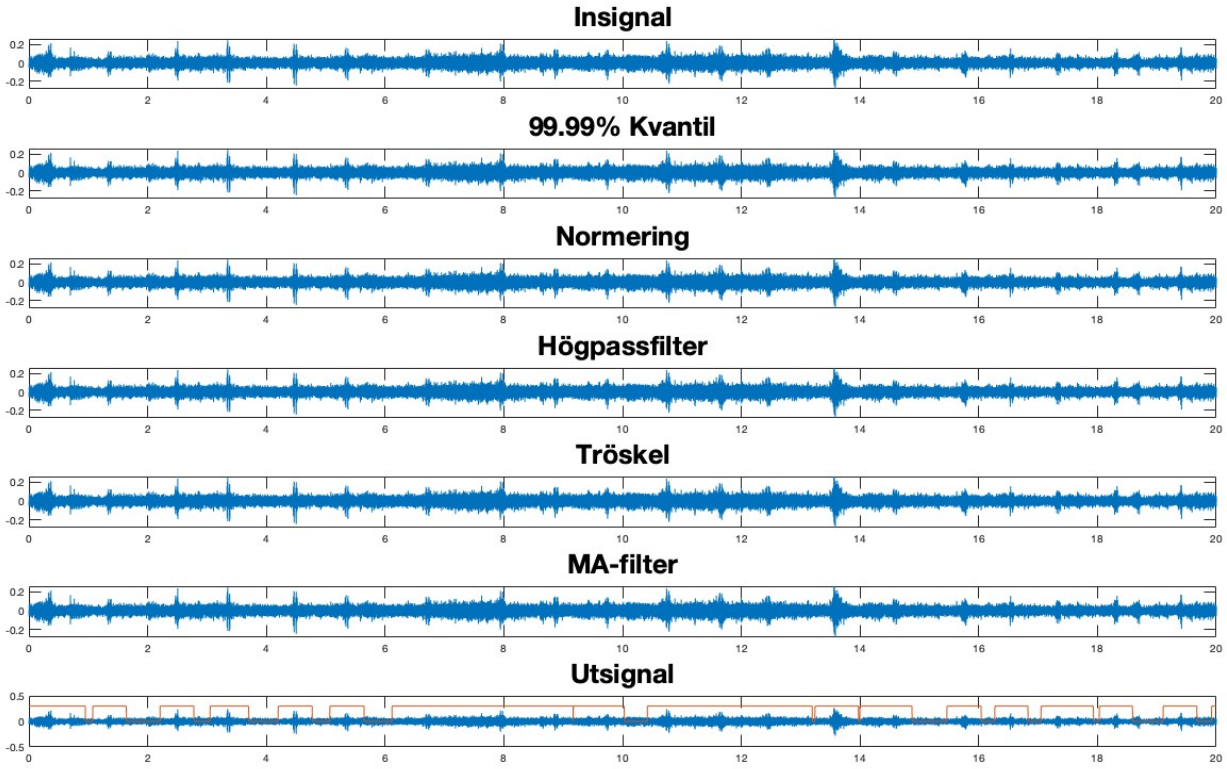
3 Implementering

3.1 Förbehandling

Ljudfiler till fågelklassificering var givna i förhand av vår handledare och således behövde ingen praktisk datainsamling göras. Ljudfilerna var nerladdade från hemsidan `https://xeno-canto.org/` och var indelade i tre mappar, en för vardera av fågelarterna gråsparv, talgoxe och bofink. Vissa ljudfiler var stereoljud och bestod alltså av flera kanaler. Då valdes endast den första kanalen och hela ljudfilen används för dataanalysen. Datasetet hade en storleksordning av ungefär $N = 60$ inspelningar av fågelsång och delades upp i ett träningsrespektive testset varav 9 observationer, tre från varje art, lades i testsetet.

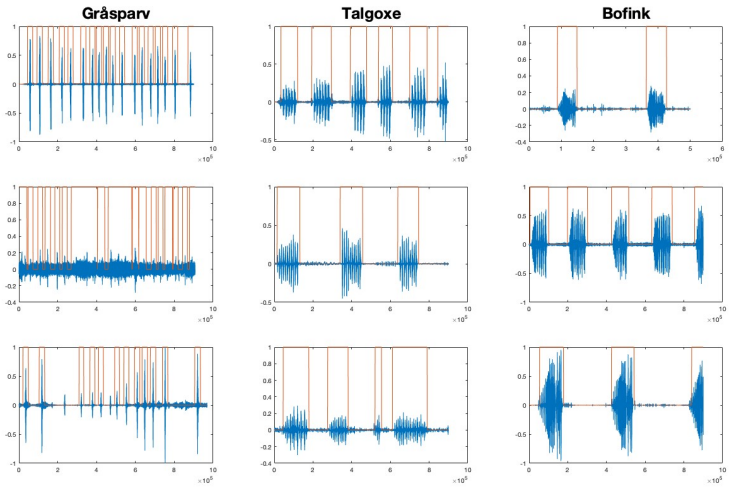
3.2 Stavelse och mellanrumssegmentering

Det första steget i vår metod för att analysera data från ljudfilerna är att kunna segmentera vilka delar av filen som innehåller en stavelse och vilka delar av filen inte gör det. Detta gjordes med den algoritm som presenterades i metoden med en kvantil på 99.99%, min-max normering, ett högpass filter över 1500Hz, tröskel på 0.27, enkelt MA-filter av 500ms och sedan ersattes alla nollskiljda värden med 1. Dessa steg visualiseras i figur 4 och exempel på hur programmet klippt ut stavelser syns i figur 5.



Figur 4: Olika stegen i metod för stavelsesegmentering

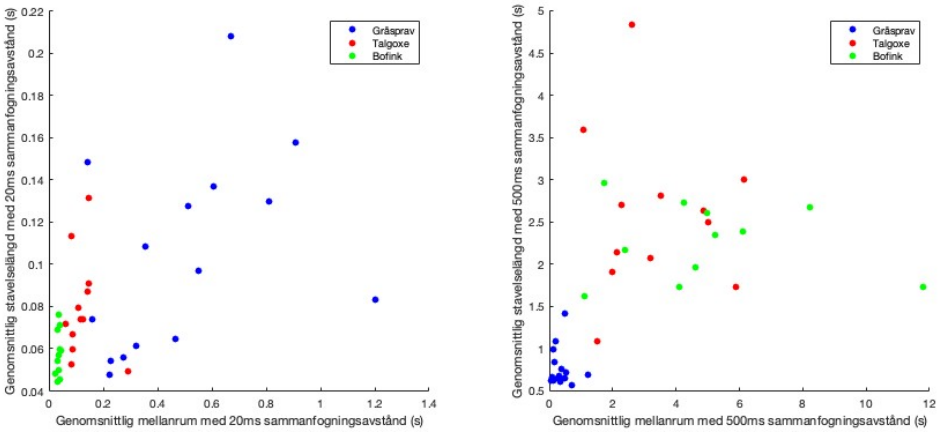
Segmenteringen hade endast en parameter, vilken vi kallar sammanfogningsavstånd. Det anger det minsta avståndet (i tid) mellan vad programmet detekterat som två stavelser för att vi ska anse att det egentligen bör räknas som en och samma stavelse. Detta innebär att samma parameter kan användas för att klassificera olika längder på stavelser utan att man behöver veta på förhand vilken fågel som sjunger. Det kan dock noteras att algoritmen har svårt för stavelser nära varandra och det rekommenderas att ta bort några av de kortaste och längsta stavelserna i varje ljudfil för att få en bättre insikt till hur stavelserna inom en ljudfil beter sig.



Figur 5: Segmenteringsalgoritmen med 9 olika ljudfiler

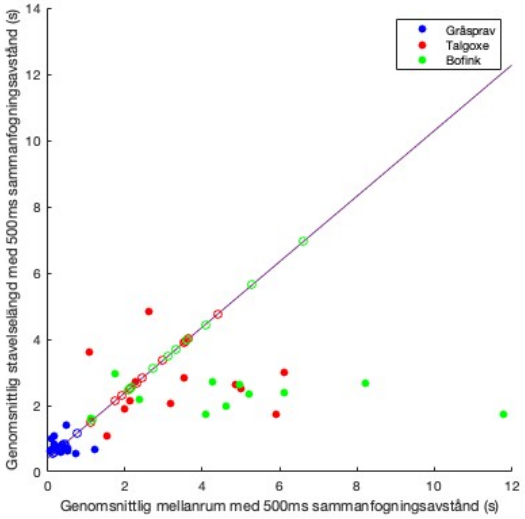
3.3 Tidsdomänsanalys

Efter att stavelser hade segmenterats analyserades både stavelser och mellanrum. Vi önskade ta bort avvikande värden och använde därför genomsnittet av de 50% mittersta längderna på stavelser och mellanrum som prediktorer för modellen. Vi noterade nämligen att algoritmen vi skapat hade svårt för stavelser som ligger väldigt nära varandra i tid och vi tog därför bort några av de kortaste och längsta stavelserna i varje ljudfil på detta sätt för att få en bättre insikt till hur stavelserna inom en ljudfil beter sig. För att analysera längd på mellanrum och stavelser kan de plottas mot varandra för ett sammanfogningsavstånd på 20ms och ett sammanfogningsavstånd på 500ms. Plotten med 500ms kommer kunna skilja på större stavelser medan de med 20ms kommer skilja på högfrekventa tidsförändringar inom stavelser.

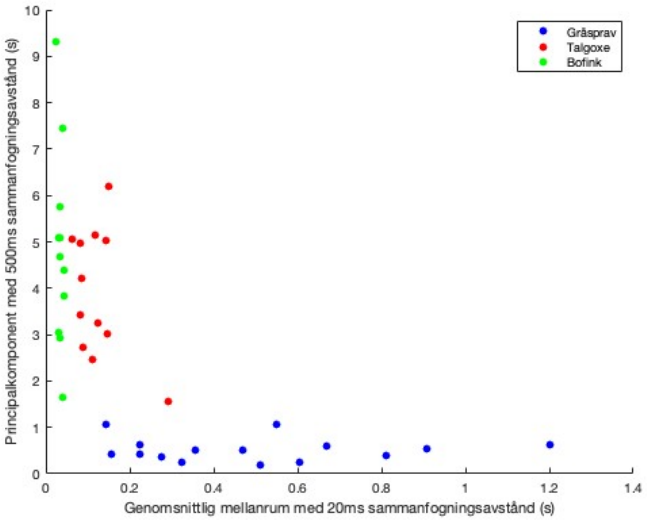


Figur 6: Genomsnittligt mellanrum mot genomsnittlig stavelselängd för 20ms (vänster) och 500ms (höger) sammanfogningsavstånd (s)

I figur 6 kan det ses att för 20ms sammanfogningsavstånd verkar endast det genomsnittliga mellanrummet på stavelserna vara den viktiga faktorn som urskiljer fåglarna. För 500ms sammanfogningsavstånd verkar det finnas en linjär beslutsgräns vilket innebär att en PCA som maximerar variansen hos gråsparven borde kunna agera som en bra prediktor för att urskilja den från de andra arterna.



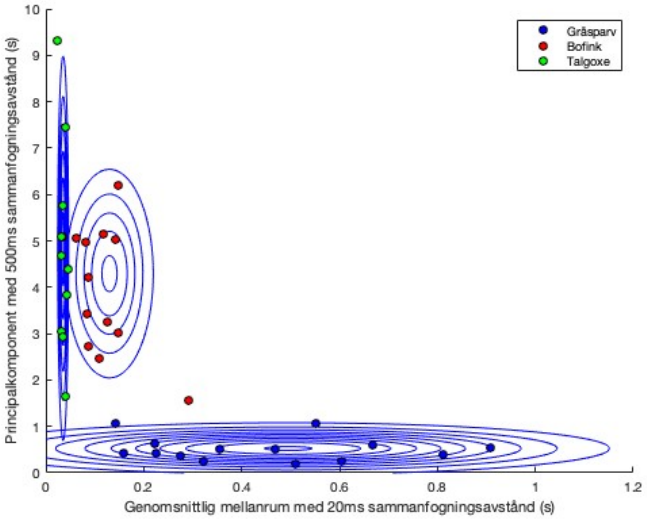
Figur 7: Genomsnittlig mellanrum mot genomsnittlig stavelse för 500ms sammanfogningsavstånd (s) med principalkomponent för gråsparven



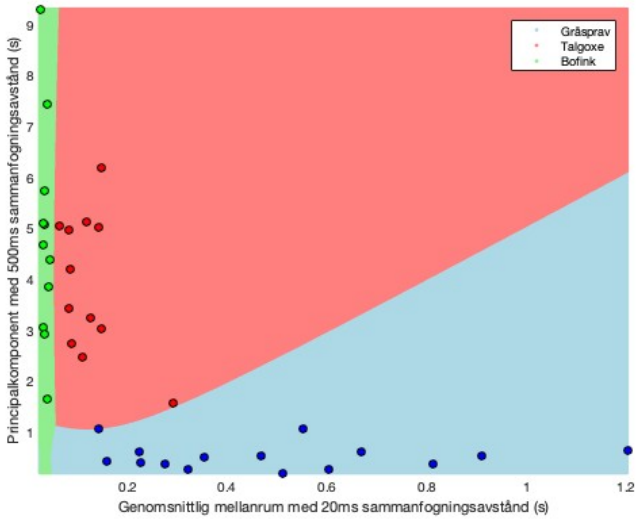
Figur 8: Genomsnittlig mellanrum för 20ms sammanfogningsavstånd (s) mot principalkomponent för 500ms sammanfogningsavstånd (s)

Figur 7 visar hur PCA med 500ms sammanfogningsavstånd används för att hitta den linjen som beskriver gråsparven bäst. Det euklidiska avståndet från linjens skärning med y-axeln används sedan som variabel i figur 8 mot det genomsnittliga mellanrummet för 20ms sammanfogningsavstånd. Tillsammans leder dessa två prediktorerna till ett schema som väldigt tydligt kan användas för att skilja på arterna. Det kan även dras slutsatsen att för att klassificera fåglar är mellanrummet mellan varje stavelse väldigt viktigt då det dyker upp som en prediktor i båda axlarna.

För att kunna skilja på arterna görs detta genom att skatta en flerdimensionell normalfördelning kring varje fågel, med antagandet att apriori-fördelningen är likformigt fördelat. Algoritmen som används är Naive Bayes med antagandet av oberoende variabler som innebär att kovariansmatrisen för normalfördelningarna är diagonal. Implementering använder matlabs färdiga funktion `fitcnb()` som implemterar maximum-likehood metoden för att skatta parametrarna.

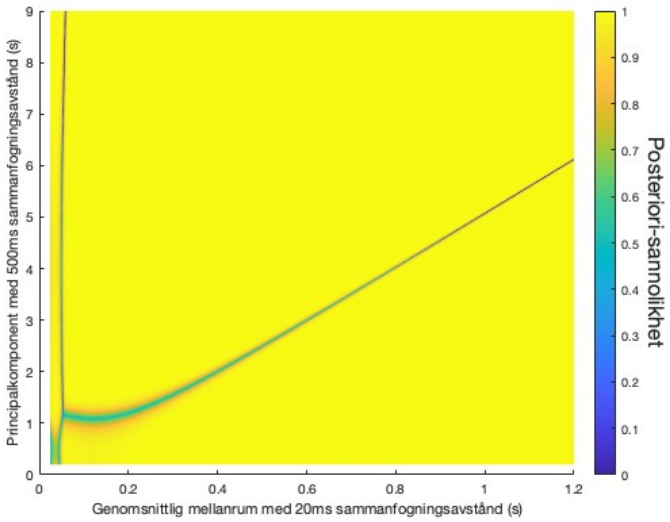


Figur 9: Skattade normalfördelningar från Naive Bayes klassificerare på genomsnittlig mellanrum för 20ms sammanfogningsavstånd (s) och principalkomponent för 500ms sammanfogningsavstånd (s)

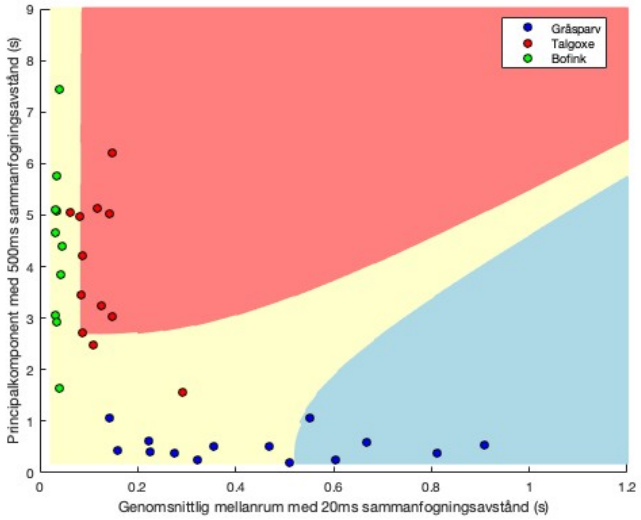


Figur 10: Beslutsgräns för Naive Bayes klassificerare på genomsnittligt mellanrum för 20ms sammanfogningsavstånd (s) och principalkomponent för 500ms sammanfogningsavstånd (s)

Figur 9 och 10 visar de skattade normalfördelningarna och beslutsgränsen mellan dem. Det är intressant att notera att det finns en talgoxe som ser väldigt lik ut som en bofink och att det även är ett väldigt liten område som avgör ifall en fågel är en gråsparv eller en bofink. För att undersöka osäkerheten hos modellen närmare kan vi kolla på posteriori-sannolikheterna för varje punkt och se vilka delar av modellen behöver mer information i klassificeringen.



Figur 11: Maximala posteriori-sannolikheter för Naive Bayes klassificerare på genomsnittligt mellanrum för 20ms sammanfogningsavstånd (s) och principalkomponent för 500ms sammanfogningsavstånd (s)

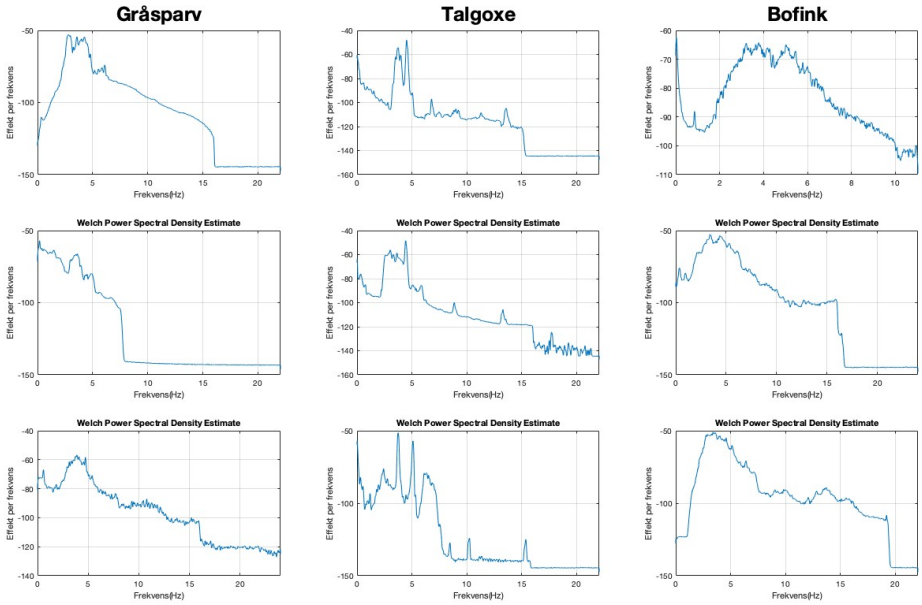


Figur 12: Posteriori-sannolikheter mindre än 1 med Matlabs känslighet (gula området) för Naive Bayes klassificerare på genomsnittligt mellanrum för 20ms sammanfogningsavstånd (s) och principal komponent för 500ms sammanfogningsavstånd (s)

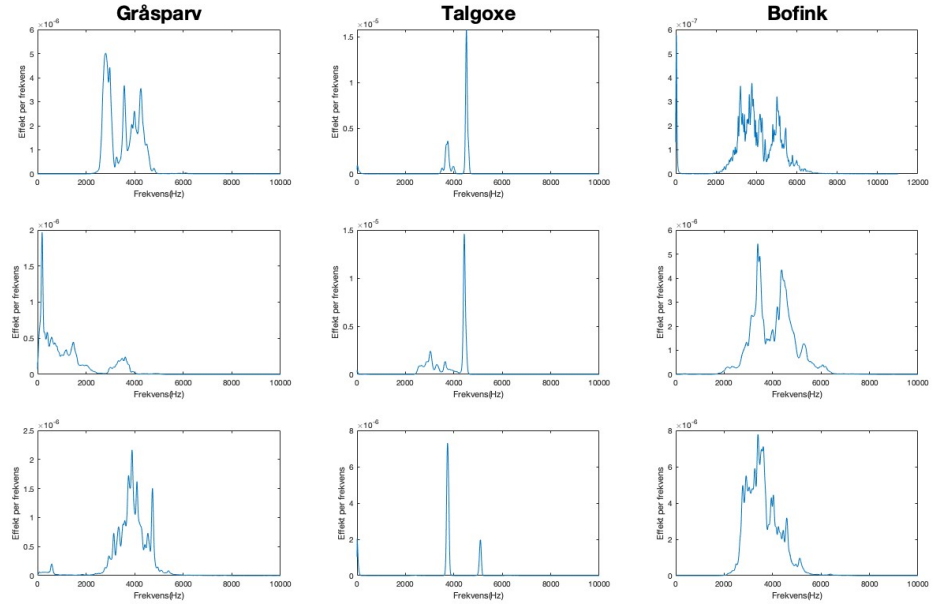
I figur 11 kan det ses att vår modell anser sig vara väldigt säker utifrån vilken klass en art tillhör från dess högsta posteriori sannolikhet. Detta beror på att de olika fågelarterna befinner sig på helt olika delar av figuren nästan utan något överlapp mellan arter. För att införa lite osäkerhet kollar vi när matlab avrundar sannolikheten till 1 (fel$< 10^{-16}$) som kan beaktas i figur 12. De gula områdena kommer behöva göras vidare frekvensanalys på medan de röda och blåa är helt säkra. Posteriori sannoliketerna för varje art i tidsdomän kommer även föras vidare i modellen för att avgöra hur viktig frekvensanalysen kommer att vara.

3.4 Frekvensdomänsanalys

För att analysera datan i frekvensdomänen behöver ett effektspektrum skattas. Detta görs med Welch-metoden som delar in signalen i ett antal fönster (vi använde fönsterlängd = 1024) med 50% överlapp, applicerar ett Hanning-fönster som kompenserar för spektral leakage och en FFT längd på 8192 som gör beräkning av koefficienter för Fourier-serien mer effektiv då det är en tvåpotens.



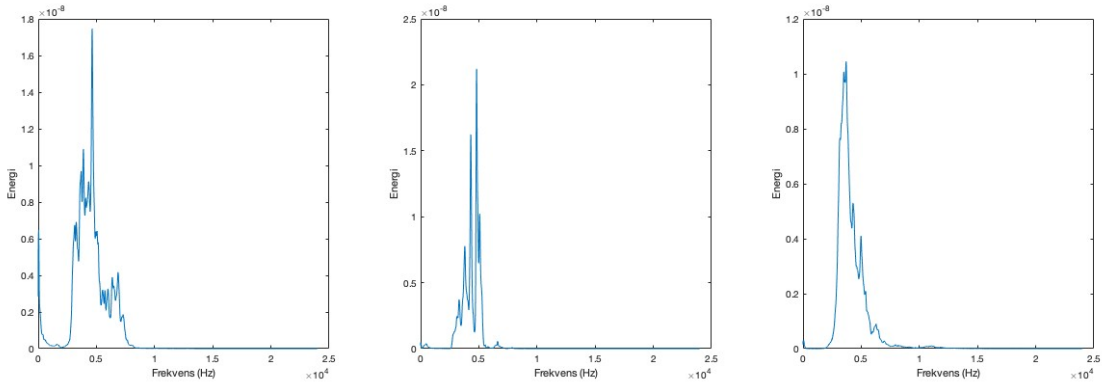
Figur 13: Effektspektrumsskattning i decibelskala för 3 gråsparvar, talgoxar och bofinkar



Figur 14: Effektspektrumsskattning för 3 gråsparvar, talgoxar och bofinkar

Figur 13 och 14 visar hur vår effektskattning fungerar på olika ljudfiler. Vi väljer att analysera datan utan decibelskala som i figur 14 eftersom decibelskalan är anpassad för det mänskliga örat som inte nödvändigtvis är

det bästa för fågelsångsklassificering. Idén för att klassificera fågelsång är att vi skapar en slags genomsnittlig stavelse och sedan gör en korskorrelation mot nya stavelser för att se vilken fågel korrelerar mest. Detta görs genom att skapa en genomsnittlig stavelse för varje fågel genom hela träningsdatan och addera ihop dem.

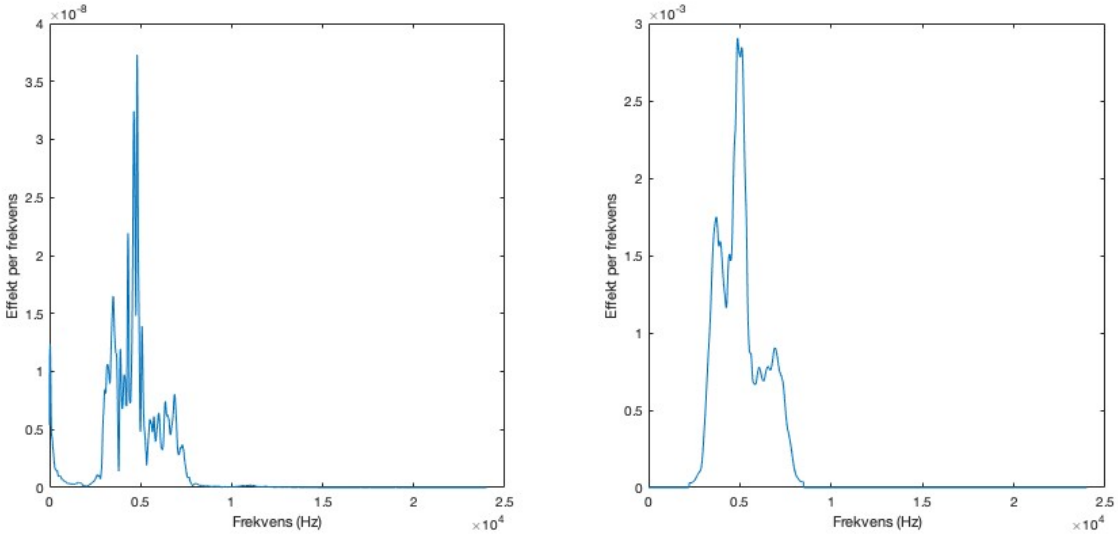


Figur 15: Ofiltrerat genomsnittlig effektspektrum för gråsparv (vänster), talgoxe (mitten) och bofink (höger)

I figur 15 kan den genomsnittliga stavelsen för gråsparvar, talgoxer och bofinker ses. Det kan noteras att alla fåglar har en stark frekvens kring 5000Hz och att talgoxen är vänster skev medan gråsparven och bofinken är högerskeva. Gråspraven har även två plateauer vid 4000Hz och 6000Hz medan bofinken är mer växande/avtagande. Dessa egenskapar kan nu användas för att urskilja på fågelarterna med en faltning.

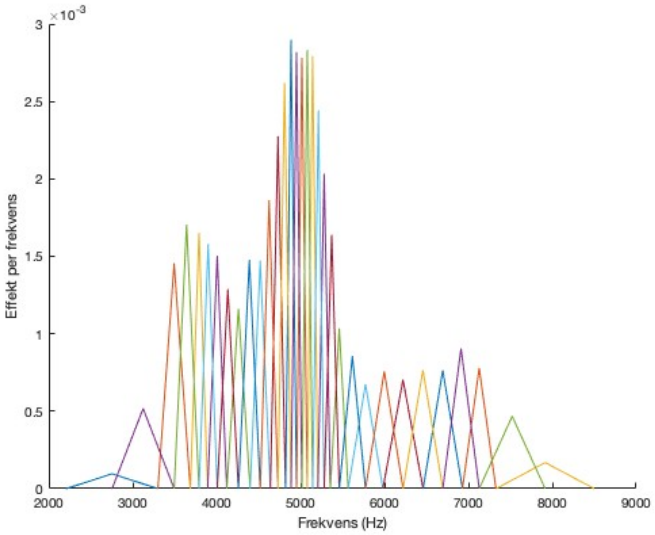
3.5 Filterdesign för förbättrad frekvensanalys

För att kunna förbättra frekvensanalysen kan en filterbank appliceras på signalen först som förstärker de viktigaste frekvenserna där fågelarterna skiljer sig mest. Det optimala filtret görs genom att vi använder vi ekvation (2) för att hitta $w(f)$ följt av ett MA-filter som jämna ut signalen och tar bort extremvärden samt ett bandpassfilter för att ta bort frekvenser vi ansåg inte påverkar klassificeringen.



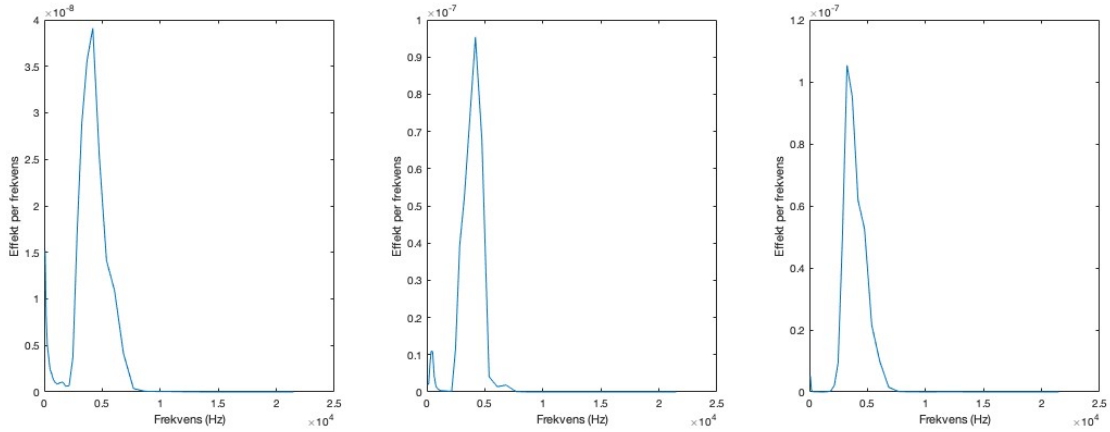
Figur 16: Skillnaden på ofiltrerad genomsnittlig effektspektrum från figur 15 (vänster) och med bandpassfilter mellan 2300Hz till 8500Hz och MA-filter av 100 värden (höger)

Nästa steg är att skapa fönsterlängden som är invers-proportionell mot effekt per frekvens. För mel-filterbanken görs det genom att invertera effektspektrumet från expontiell till logaritmisk men eftersom vårt filter inte är inverterbart kan detta göras iterativt istället. Vi börjar från vänster till höger och adderar ett genomsnitt mellan nästa frekvens och nuvarande.

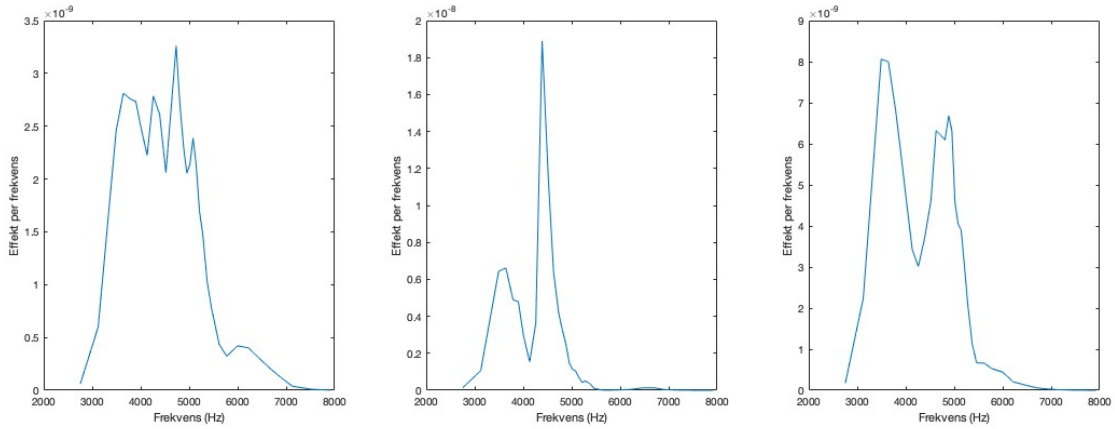


Figur 17: Triangulära fönstren för vårt designade filter

Vårt filter går mellan 2207 Hz och 8500 Hz och har 32 bins precis som mel-spektrogramet. Det finns tydligt större betoning på frekvenser kring 5000Hz till skillnad från mel-filterbanken där de låga frekvenserna får mer vikt. Filterbanken kan sedan appliceras på vår träningsdata för att skapa nya genomsnitt att jämföra med.



Figur 18: Genomsnittlig effektspektrum för gråsparv (vänster), talgoxe (mitten) och bofink(höger) med Mel-filter



Figur 19: Genomsnittlig effektspektrum för gråsparv (vänster), talgoxe (mitten) och bofink(höger) för vårt filter

4 Resultat

I tabell 1 syns andelen korrekt identifierade av fågelarter från vår övningsdata samt den sammanlagda korrekta klassificeringen. På tabellens första raden har inget filter, utöver Welch, applicerats på stavelserna efter att de överförts till frekvensdomänen. På andra raden har den etablerade Mel-filterbanken applicerats på stavelserna och på den tredje har det optimala filtret i figur 17 applicerats innan korskorrelation.

Tabell 1: Klassificeringsprecision på träningsdatan för olika filter på träningsdatan uppdelad per art

Filter	Gråsparv	Talgoxe	Bofink	Totalt
Inget filter	0.790	0.692	0.500	0.661
Mel-filterbank	0.526	0.231	0.700	0.476
Förbättrad filterbank	0.947	0.846	0.400	0.786

Tabell 2: *Confusion matrix* för vår förbättrade filterbank på träningsdatan.

	Gråsparv	Talgoxe	Bofink
Gråsparv	11	0	2
Talgoxe	0	18	1
Bofink	1	5	4

I tabell 2 syns antal gånger programmet klassat en viss fågel som respektive av de tre arterna, där den korrekta fågelarten representeras på raderna och den klassificerade arten på kolonnerna. Tabell 3 visar de två

Tabell 3: Klassificeringsprecision på testdatan

Metod	Korrekt klassificering
Tidsanalys	9 av 9
Filtrerad frekvensanalys	7 av 9

klassificeringsalgoritmernas testresultat på de 9 okända ljudfilerna

5 Diskussion och Slutsatser

Resultatet visar att vår förbättrade filterbank presterar med högre noggrannhet än om inget filter används, vilket i sin tur presterar bättre än mel-filterbanken. Det här beror förmodligen på hur mycket genomsnittsstavelserna för de olika fågelarterna skiljer sig åt. Mel-genomsnittet har buntat ihop många av de frekvenser som visat sig vara extra viktiga att studera för fåglar och förstärkt många av de låga frekvenser där arterna inte skiljer sig så mycket åt, se figur 18. Detta gör att medelstavelserna ser väldigt lika ut. När varken Mel eller vårt egna filter appliceras på signalen blir medelstavelserna mycket hackigare, detta syns tydligt på figur 15. Inga särskilda frekvenser förstärks, så det finns mycket information i stavelsen som inte är relevant för att differentiera fågelarterna. Man kan säga att dessa medelstavelser har högt brus, vilket är ett störningsmoment när korskorrelation mellan två signaler utförs. Medelsnittsstavelserna där vår egenkonstruerade filterbank använts har större skillnader sinsemellan jämfört med de ofiltrerade medelsnittsstavelserna. Alltså förstärker vår filterbank alla stavelsers unika egenskaper i form av frekvenstoppar och dalar, som påverkar eventuella korskorrelationer på ett betydelsefullt sätt.

Tyvärr verkar både klassificeringen i tidsdomänen och frekvensdomänen ha en tendens att blanda ihop samma fågelarter, i figur 7 syns det hur de gröna punkterna för bofinkar är ganska nära de röda talgoxarna. Ett liknande fenomen syns i tabell 2 där bofinkarna ofta klassificeras som talgoxar, dock inte vice versa. Förhoppningen var att de skulle blanda ihop två olika par av fågelarter, i det fallet hade metoderna kunnat kombineras där sannolikheter viktas olika för olika fågelarter och metodernas osäkerheter tar ut varandra. Detta är alltså inte möjligt just nu då båda metodernas osäkerheter är mellan samma två fågelarter, dessa sammanfallande osäkerheter gör att en kombination av metoderna blir överflödig då klassificeringen i frekvensdomänen endast är en lite sämre klassificering än den i tidsdomänen. Med mer tid kanske detta hade kunnat åtgärdas men som det ser ut nu är den bästa metoden för klassificera fågelarterna att endast ta hänsyn till tidsdomänen, detta syns väl i tabell 3 där klassificeringen i tidsdomänen presterar bättre än den i frekvensdomänen. Att få frekvensdomänens osäkerheter att skilja sig från tidsdomänen hade varit intressant att undersöka vidare i mån av tid, då en förbättring där hade kunnat medföra en ännu mer exakt kombinerad klassificeringsalgoritm. Dock finns det inte mer utrymme för detta i tidsramen för detta projekt.

Eftersom vår filterbank är väldigt specifik för just de fågelarter som vi har valt att analysera är den troligtvis inte användbar för en godtycklig annan fågelart. Då skulle vi behöva införa data från den nya fågelarten, och genomgå alla steg i metoden igen, för att kunna identifiera just den fågelarten. Om vi skulle gjort detta för en stor mängd fågelarter är då möjligt att få en ny mer generell filterbank, som kanske skulle efterlikna Mel fast vara bättre lämpad för högre frekvenser. Detta medför dock att den antagligen skulle förlora precision, och då skulle det inte längre vara säkert att den skulle ge bättre resultat än Mel.

Referenser

[1] C.-C. C. Chang-Hsing Lee, Chin-Chuan Han, "Automatic classification of bird species from their sounds using two-dimensional cepstral coefficients," *IEEE Transactions on Audio, Speech, and Language Processing*, 2008.

[2] M. Hansson-Sandsten, "Classification of bird song syllables using singular vectors of the multitaper spectrogram," *2015 23rd European Signal Processing Conference (EUSIPCO)*, 2015.

[3] H. J. v. d. H. L. J. P. van der Maaten, E. O. Postma, "Dimensionality reduction: A comparative review," *Journal of Machine Learning Research*, vol. 10, no. 1, pp. 66–71, 2009.

[4] Avcontentteam. (2024) What is pca and how it works. [Online]. Available: https://www.analyticsvidhya.com/blog/2016/03/pca-practical-guide-principal-component-analysis-python/