Bacheloroppgave

NTNU
Norges teknisk-naturvitenskapelige universitet
Det humanistiske fakultet
Institutt for musikk

Vegard Hult

Dynamisk lysstyring til klubbmusikk ved hjelp av maskinlæring

Bacheloroppgave i Musikkteknologi
Veileder: Andreas Bergsland
Mai 2023

NTNU
Kunnskap for en bedre verden

Vegard Hult

Dynamisk lysstyring til klubbmusikk ved hjelp av maskinlæring

Bacheloroppgave i Musikkteknologi
Veileder: Andreas Bergsland
Mai 2023

Norges teknisk-naturvitenskapelige universitet
Det humanistiske fakultet
Institutt for musikk

NTNU
Kunnskap for en bedre verden

Sammendrag

Klubbopplevelsen kan forsterkes ved hjelp av et koordinert lysshow. Det eksisterer idag flere automatiske løsninger for å synkronisere lyseffekter med musikk, men ingen av disse systemene kan følge oppbygningen i musikken på samme måte som en lystekniker med god forståelse av musikalsk oppbygning. Denne oppgaven vil utforske mulighetene for å utvikle programvare som etterlikner valgene en lystekniker tar for å følge den musikalske energien i en klubbsetting, ved hjelp av maskinlæringsalgoritmer.

Programvaren var ikke funksjonell ved prosjektets slutt, og førte ikke til et resultat som kan evalueres oppimot målsettingen. Denne oppgaven er derfor heller en oppsummering av arbeidet og funnene som ble gjort underveis, og skisser til mulige løsninger for videre utvikling av programvaren.

Abstract

The experience of clubbing can be amplified by a coordinates light show. Multiple automatic solutions for synchronizing lighting effects with music exist today, but none of these are able to follow the energy curve of the music in the same way as a dedicated lighting technician with understanding of musical structure. This thesis explores possibilities for development of software that mimics the way a light technician adapts the lighting to fit the energy of music in a club setting, using machine learning algorithms.

The software was not functional at the end of the project, and the results cannot be evaluated based on the original objective of the project. This thesis will therefore read as a summary of the process and the findings that were made along the way, and sketches for possible solutions for future developments of the software.

Ordliste

Beat	– Et slag plassert på fjerdedelsunderdeling i takt
Downbeat	– Første slag i en takt
Build	– Seksjon i musikk hvor spenning bygges opp, gjerne etterfulgt av et refreng eller drop hvor spenningen oppløses
Drop	– Seksjon med høyt energinivå, kommer ofte etter et build. Kan sammenliknes med et refreng i populærmusikk.
Mapping	– Hvordan et parameter knyttes til et annet
Sound to Light	– Lys som reagerer automatisk på lyd
MIR	– Music information retrieval
ISMIR	– International Society of Music Information Retrieval

Innholdsfortegnelse

1 Introduksjon

Klubbopplevelsen kan være nærmest transe-dannende. Med pulserende rytmer, repetitive musikalske mønstre, flytende folkemengde, dundrende bass, og mørk lyssetting, kan klubbscenen oppleves som et parallelt univers fra utenomverdenen. Det kollektive energinivået i rommet følger musikkens oppbygning, og en DJ har flere musikalske muligheter for å bygge spenning i rommet. Ved å kutte bort bassfrekvenser mister klubberne det rytmiske ankeret sitt, og risere som stiger i frekvens tilsynelatende i det uendelige, bygger tydelig mot et høydepunkt. Ved å oppløse spenningen inn i et drivende rytmisk drop eller refreng, kan klubberne oppleve en felles følelse av eufori. Den rytmiske drivende musikken, og taktile direkte følelsen av sub-bassen som vibrerer gjennom kroppen skaper et voldsomt energiutløp idet spenningen oppløses, men denne følelsen kan også forsterkes av andre sanseinntrykk. I artikkelen «The experience of nature: A psychological perspective» (R. Kaplan & S. Kaplan, 1989) nevnes sterke lys som en av flere stimuli med «direct exciting quality». Ved å sakte senke belysningen parallelt med spenningen som bygges i musikken, og deretter kontrastere mørket med sterke, og gjerne blinkende lys idet den musikalske spenningen oppløses, forsterkes følelsen av eufori blant klubberne.

På større klubbscener finnes det ofte en dedikert lystekniker som koordinerer lysshowet med musikken og energinivået i rommet. Dette fører til en nøye koordinert helhet. I mindre klubber har ofte DJ-ene selv enkel kontroll over strobelys, og kan selv koordinere sterke blinkende lys med droppet i musikken. Dette styrker den euforiske opplevelsen av droppet, men mangler nyansene en dedikert lystekniker kan oppnå.

Det finnes også allerede et mangfold av verktøy for automatiske lysshow, som skal koordinere bevegelse i lyssetting og fargeendringer med tempoet til musikken. Disse baserer seg i stor grad på energien i enkelte transienter, og har ingen overordnet forståelse for opplevd energinivå, eller musikalsk struktur. Spørsmålstillingen for denne oppgaven vil da være:

> *Er det mulig å benytte maskinlæring til å skape dynamiske lysshow, og hvordan kan man gå fram for å implementere dette?*

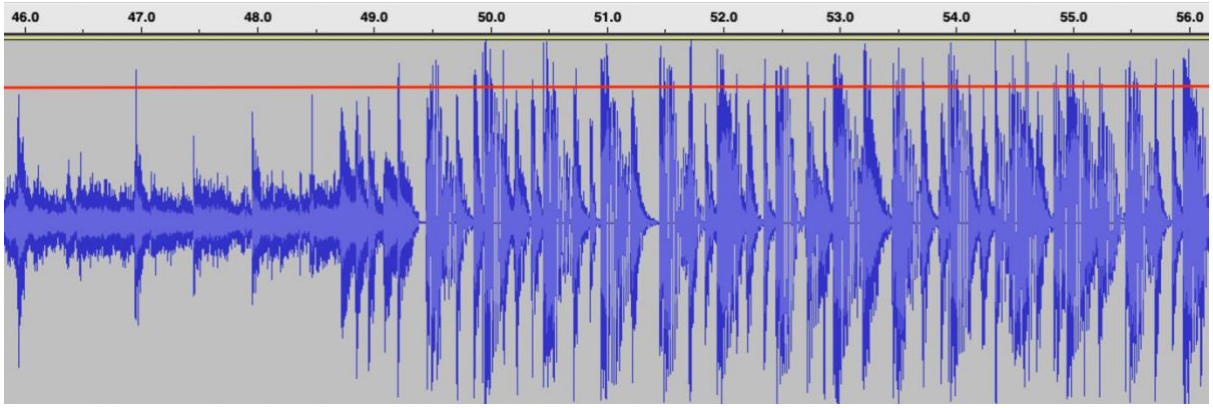
Denne oppgaven vil da undersøke mulighetene for å implementere overordnet musikkforståelse og opplevd energinivå i musikk, i automatisk genererte lysshow i det

digitale domenet. Endemålet vil være å utvikle programvare som kan tilnærme seg funksjonen til en aktiv lystekniker på mindre klubbscener.

Jeg har valgt å benytte meg av programmeringsspråket python, og maskinlæringsverktøyet Tensorflow for å implementere denne funksjonaliteten. Grunnen til dette er at jeg allerede er kjent med programmering i python, og det finnes mange gode læringsressurser for maskinlæring i python. Tensorflow er utviklet av Googles AI team, og er også svært godt dokumentert.

1.1 Kontekst

Flere enkle lamper og lysbord har lydinngang eller innebygget mikrofon som kan reaktivt endre lysstyrke eller farger ved transienter. Dette kan gjøres ved å sette en terskel for hvor sterkt lydsignalet må være før en lysendringen skjer.

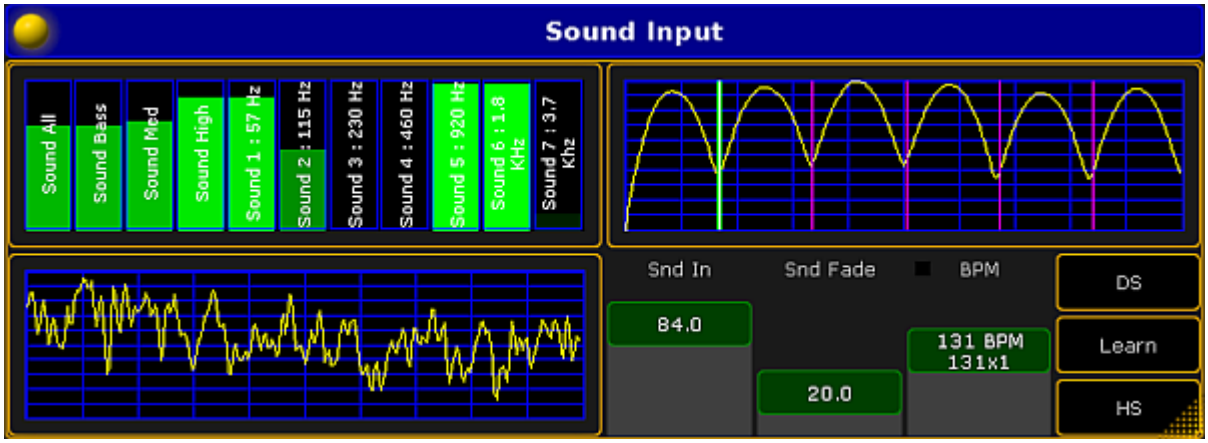


Figur 1 Lydbølgerepresentasjon med terskel

I dette tilfellet vi det skje en endring i lyssetting hver gang lydsignalet overstiger den røde linjen i figur 1. Dette vil føre til lysendringer som korresponderer til transienter i musikken, men det kan oppleves vilkårlig hvilke transienter som utløser endringen. Endringen vil også være lik hver gang, men mere hyppig i seksjonene med høyere volum. Med terskelen i figur 1 vil vi for eksempel kunne få et tilfeldig sterkt blink rundt tidspunktet 46.9, selv om musikken i denne delen av låten er svært rolig. Det vil også være flere aktiveringer refrenget enn de vi føler som musikalsk viktige å markere. Denne tilnærmingen er reaktiv, men forholder seg ikke til musikalske strukturer.

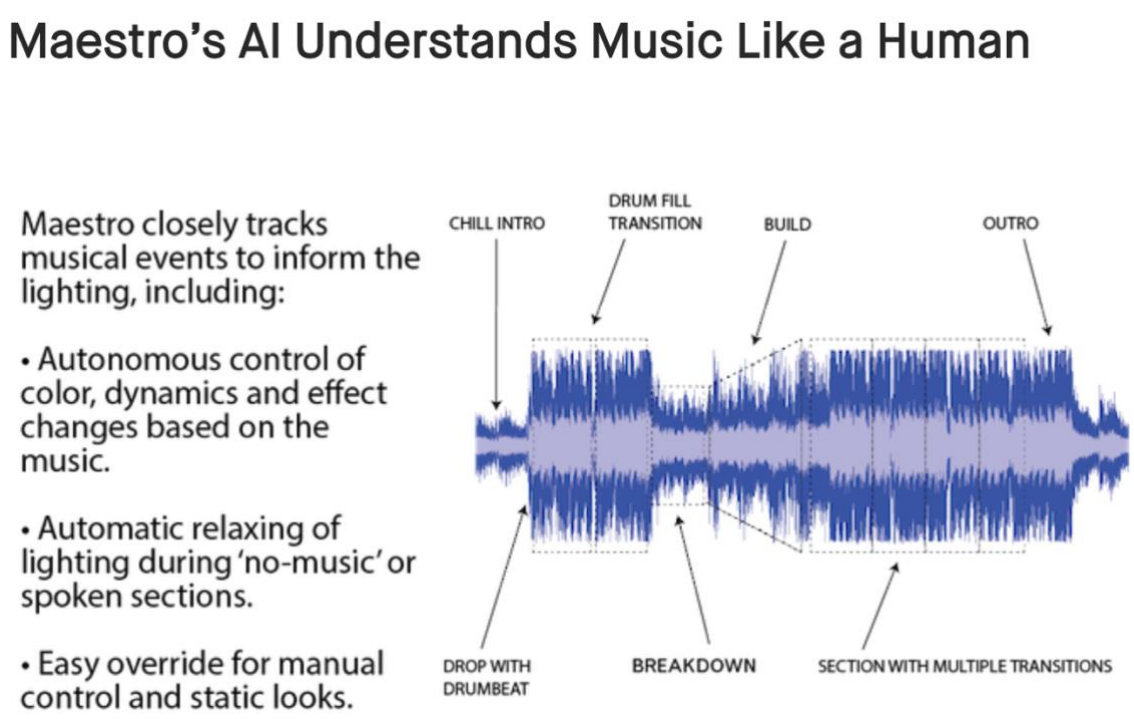
Moderne lysbord har ofte en litt mer kompetent Sound to Light funksjonalitet. Lysbordene til MA Lighting gir tilgang til flere frekvensbånd, og muligheten til å la disse aktivere forskjellige lampegrupper. Lysbordet gjør også et estimat av tempoet til musikken,

som kan brukes til å kjøre synkroniserte effekter. Her estimeres tempo ved å forsøke å tilordne transientene over et kort tidsløp til et fast grid. Tempoestimatet er svært nøyaktig ved musikk med lav rytmisk kompleksitet, men sliter med mer avansert rytmikk som synkoper, polyrytmer og polymetrikk.



Figur 2 MA2 Sound to Light funksjonalitet (MA Lighting, i.d.)

I dag er det flere aktører som utforsker bruken av kunstig intelligens innen lysdesign for klubbscener. Force 1 fra AI Lightshow, og MaestroDMX som i skrivende stund har en aktiv kickstarterkampanje, er produkter som forsøker å forbedre automatiske lysshowløsninger ved hjelp av maskinlæring. Disse har imidlertid hovedfokus på simpelt oppsett og enkel bruk, og tillater lite fleksibilitet. Det er også svært lite åpenhet om hvordan maskinlæringsmodellene er trent, hva de lytter etter i musikken, og hvordan mappingen foregår.



Figur 3 Forklaring av AI-en bak MaestroDMX (MaestroDMX, 27.04.2023)

Forklaringen av AI-funksjonaliteten på kickstartersiden til MaestroDMX er overfladisk, og oppgir ingen informasjon rundt oppbygningen av maskinlæringsmodellen og hvilke data den er trent på. Det er allikevel tydelig at modellen differensierer mellom ulike seksjoner vi forventer å finne i klubbmusikk, og kan benytte dette til å ta valg for lyssettingen.

I denne oppgaven ønsker jeg å jobbe mot å utvikle en programvare med følgende funksjonalitet:

- Tempodeteksjon: For å kunne synkronisere effekter med musikken.
- Downbeatestimering: Modellen må kunne identifisere første slag i en takt for å kunne forstå musikalsk struktur, og kunne gjøre endringer på lyssetting i overganger mellom seksjoner i musikken.
- Klassifisering av seksjonstype: Modellen må kunne gjenkjenne seksjoner med lav og høy spenning eller energi.
- Styring av ekstern lysprogrammeringsenhet: Jeg ønsker ikke å implementere funksjonalitet for lys-design og -programmering i denne programvaren, og må derfor heller implementere kommunikasjon til ekstern kontroller for å håndtere lysdesignet.

2 Teori:

Maskinlæring er et komplekst tema, hvor mange avanserte matematiske funksjoner ligger bak funksjonaliteten. Personlig har jeg ikke inngående kunnskap innen matematikken bak maskinlæring, og dette inngikk heller aldri i omfanget av oppgaven. Jeg ønsker derfor ikke å gå i detalj på prosessene bak maskinlæring, men heller gjøre rede for funksjonaliteten til verktøyene jeg har benyttet i denne oppgaven.

2.1 Relevant musikalsk informasjon for lyssetting

For å forsterke rytmikken i musikken, kan de forskjellige lyskasterne i rommet programmeres til å blinke i takt med musikken. For å gjøre dette noe mer interessant kan forskjellige lysgrupper mappes til forskjellige slag for å differensiere mellom funksjonen av disse. En gruppe lamper kan for eksempel blinke på downbeats, og en gruppe kan blinke på upbeats. For å oppnå denne funksjonaliteten i programvare, trenger den en overordnet forståelse av taktstruktur.

For å kunne forsterke energien i musikken ved høydepunktene, må også lyssettingen kunne senke energinivå sammen med musikken for å skape kontraster, for eksempel mellom et build og et drop. For at lysendringen skal skje samtidig som endringene i energinivå i musikken må en lystekniker ha forståelse av musikalsk struktur, og funksjonene til ulike seksjoner i klubbmusikk. Hen må også kunne forutse kommende endringer i musikken slik at lysendringene kan koordineres, og inntreffe på samme tidspunkt som endringene i musikken.

Mye klubbmusikk benytter en struktur hvor større musikalske endringer skjer hver åttende takt. Det virker derfor hensiktsmessig å inkludere en periodeforståelse i maskinlæringsmodellen.

2.2 Maskinlæring med Python og Tensorflow

Tensorflow er et maskinlæringsverktøy utviklet av Googles AI team. Tensorflow kan importeres som et pythonbibliotek, og brukes til å utvikle maskinlæringsmodeller i programmeringsspråket python.

Ved å gi maskinlæringsmodellen eksempler på input og ønsket output i treningsprosessen, lærer modellen selv hvilke funksjoner som fører til riktig output. På denne

måten trenger vi ikke å fortelle modellen hva den skal se etter i lydeksemplene, men heller hvilken informasjon vi vil at den skal hente ut, og eksempler på riktig output.

Dataen modellen trenes på er nødt til å ha samme form som input og output når vi ønsker å estimere verdier ved hjelp av modellen. Det er derfor viktig å formatere datasettet på riktig måte før treningsfasen igangsettes.

3 Metode

Gjennomføringen av prosjektet kan deles inn i tre overordnede faser. Den første fasen gikk ut på å anskaffe et passende datasett som maskinlæringsmodellen kunne trenes på. Deretter kunne maskinlæringsmodellen utformes ved hjelp av nettressurser og eksempelkode fra liknende prosjekter, og trenes på datasettet. Treffsikkerheten til modellen kunne deretter evalueres og utvikles videre, og tilslutt kunne estimeringene til maskinlæringsmodellen mappes til preprogrammerte lyssettinger på en ekstern lyskontrollflate.

For at maskinlæringsmodellen skal kunne oppnå en tilfredsstillende treffsikkerhet på estimering, kreves tilstrekkelig treningsdata. Andre maskinlæringsmodeller som trener på bildedata bruker datasett med over 500 eksempler, og gjerne opp mot flere tusen. Gtzan datasettet, som brukes mye innenfor maskinlæring med musikk, inneholder for eksempel 1000 annoterte lydklipp med lengde på 30 sekunder (Tzanetakis et al., 2001). Flere treningsfiler gir bedre nøyaktig, men for at treningsdataen skal gi gode resultater må den oppfylle et par kriterier:

- Musikkeksemplene i datasettet bør være innenfor passende sjanger. Eksisterende datasett som Gtzan datasettet inneholder et utvalg musikkeksempler innen flere sjangre, men ikke elektronisk klubbmusikk. Derfor ble jeg nødt til å samle og annotere passende musikk til prosjektet.
- I første omgang ønsker jeg å trene maskinlæringsmodellen til å gjenkjenne generiske musikalske trekk som brukes i et stort utvalg av klubbmusikk. Derfor ønsker jeg å kun inkludere klubbmusikk som har tydelig distinkte seksjoner med forskjellig funksjon og energinivå, og har build seksjoner som bla. Kutter ut bassfrekvenser og bruker risere for å bygge spenning. På denne måten vil det være tydeligere for modellen hvilke aspekter som assosieres med ulike seksjonstyper.
- Musikkeksemplene bør ha et stødig tempo gjennom hele lydfilen, for å gjøre annoteringsprosessen enklere.
- Datasettet bør være riktig annotert, slik at modellen kan etterlikne disse annoteringene. Feil annotering vil føre til uforutsigbare prediksjoner fra modellen.

Maskinlæringsmodellen må utformes slik at den aksepterer datasettet som input, og har tilfredsstillende nøyaktighet ved prediksjoner av beats og seksjonstyper utfra spektrogrammene vi mater den med.

Når modellen er funksjonell kan effektiviteten evalueres, og eventuelle endringer i modellen eller datasettets struktur kan utføres for å forsøke å bedre effektiviteten. Til slutt kan kommunikasjon med ekstern lyskontroller implementeres.

4 Resultater

Etter prosjektperioden er ikke den skisserte programvaren funksjonell. Denne delen av rapporten blir derfor en oppsummering av prosessen, funksjonaliteten til verktøyene som ble ferdige i løpet av prosjektperioden, og et forslag til hvordan manglende funksjonalitet kan implementeres.

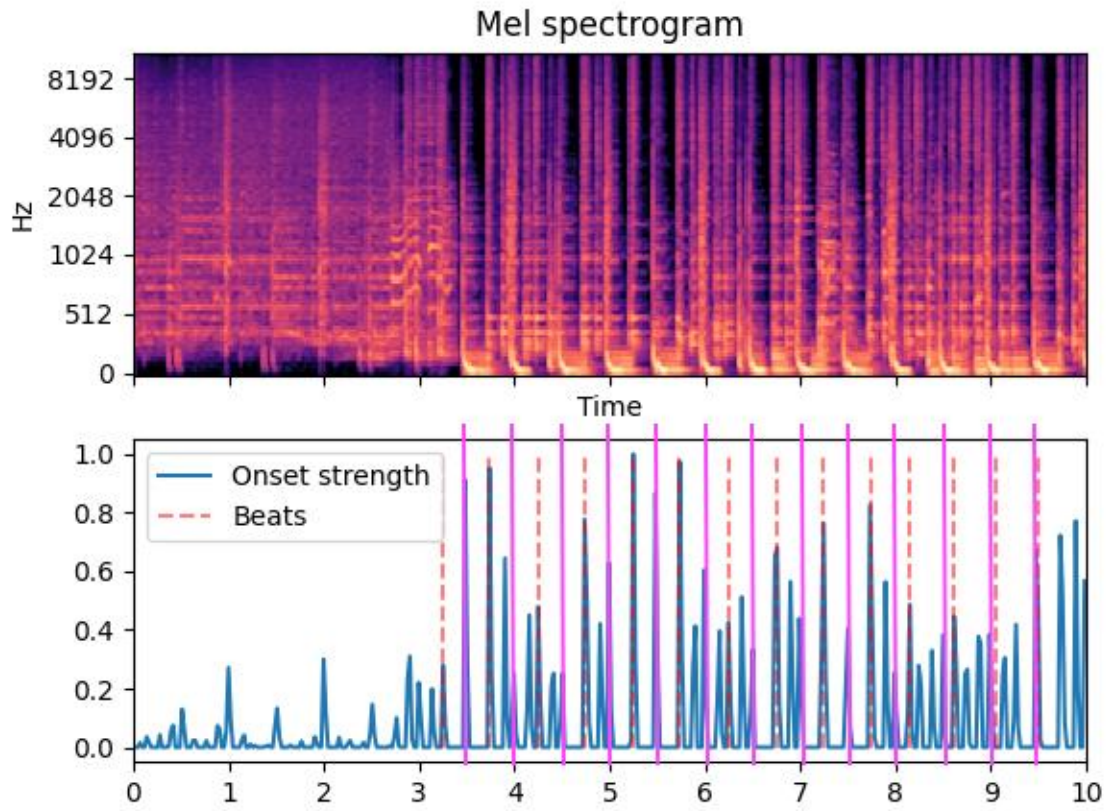
4.1 Datasett

For å kunne gjøre effektive forutsigelser med maskinlæringsmodeller, må treningsdataen inneholde all informasjon vi ønsker å forutsi med modellen. Dataen må også være formatert på en effektiv måte for modellen å lese av, og være riktig annotert. Det finnes et utvalg av datasett bestående av musikk og annoteringer av taktslag, og også noen med annoterte taktstartpunkter. Gtzan datasettet brukes ofte til å trene maskinlæringsmodeller på musikk, og det har en utvidelse som inkluderer taktslagannoteringer, men det mangler taktstartpunkter og klassifisering av seksjoner. Det er heller ingen av disse datasettene som fokuserer på elektronisk klubbmusikk. For å kunne trene modellen på relevant musikk med riktige annoteringer, var det derfor nødvendig å produsere et egnet datasett.

Datasettets formatering startet som en videreutvikling av gtzan_tempo_beat datasettet med beat annoteringer laget av Magdalena Fuentes (05.11.2021). Dette datasettet inneholder en fil med tempo, og en fil med en liste hvor hver rad består av tidspunkt og korresponderende taktindeks for hver beat, per lydfil i gtzan datasettet. Verdiene for taktindeks og periodeindeks, og periodetype ble lagt til i samme rad som tilhørende tidspunkt og beatindeks.

Jeg valgte å ta utgangspunkt i musikk fra «Energy (Deluxe)» albumet fra artisten Disclosure i startfasen av annoteringen, fordi albumet inneholder rytmisk drivende klubbmusikk med tydelige endringer mellom seksjoner. Eksemplene hentet frem i denne oppgaven er fra «Expressing what matters» (Disclosure, 2020).

Manuell annotering er en prosess som er tidkrevende, og maskinlæringsmodeller av denne kompleksiteten krever mange treningsfiler. Derfor ble det naturlig å lage en strømlinjeformet prosess. Librosa biblioteket til python har funksjonalitet for å finne taktslag og estimere tempo ved hjelp av transient deteksjon, men da dette ble implementert i annoteringsprosessen viste det seg å være svært unøyaktig. I figur 4 er librosas foreslåtte taktslag markert med røde stiplede linjer, mens faktiske taktslag er markert med hele linjer i magenta.



Figur 4 Spectrogramfremstilling og transient deteksjoner av «Expressing what matters» fra artisten Disclosure (2020). Foreslåtte taktslag fra librosa er markert i røde stiplede linjer, og faktiske takslag er markert med hele linjer i magenta.

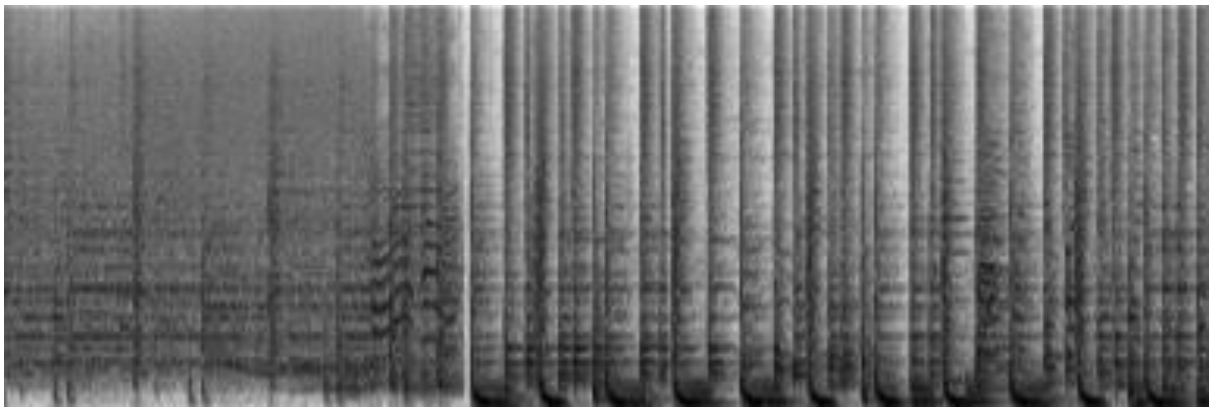
I det ferdige annoteringsverktøyet brukes allikevel librosas forslag til tempo som en pekepinn for brukeren. Og sammen med tap-tempo funksjonalitet mens et utdrag av musikken spiller, kan brukeren benytte disse to estimatene til å oppgi tempo.

Ettersom librosa også viste seg å være unøyaktig ved markering av taktslag, ber annoteringsverktøyet brukeren om å oppgi tidspunkt for det første slaget i en periode bestående av åtte takter, og regner deretter ut tidspunktet for resten av taktslagene og periodene basert på tempoet oppgitt av brukeren i forrige steg.

For å annotere seksjonstype itererer annoteringsprogramvaren over alle seksjonsstartpunktene fra forrige steg, og spiller av noen sekunder av lydfilen. Brukeren oppgir da et heltall fra 0 til 3 som klassifiserer seksjonen som en av fire kategorier:

0: Stillhet	Stillhet eller lyd med lavt energinivå. Intro og outro eller overganger mellom låter.
1: Vers	Lyd med mindre musikalsk energi
2: Build	Lyd med økende spenning
3: Drop	Lyd med høy energi. Ofte plassert etter en «build» seksjon.

Utviklingen innen maskinlæring har i stor grad fokusert på bildedata, og det finnes ikke noen gode verktøy for å trene maskinlæringsmodeller på lydfiler. Derfor konverterer annoteringsverktøyet lydfilene til spektrogrammer, som gir informasjon om lydstyrke og fordeling over frekvensspekteret over tid. Spektrogrammet inneholder ti sekunder med lyd representert over 385 piksler på x-aksen, og frekvensspekteret representert med 185 piksler på y-aksen. Spektrogrammet konverteres også til gråskala for å minimere filstørrelsen. Mørke områder representerer høy lydenergi, og lyse områder representerer lav lydenergi.



Figur 5 Spektrogramrepresentasjon av lydklipp

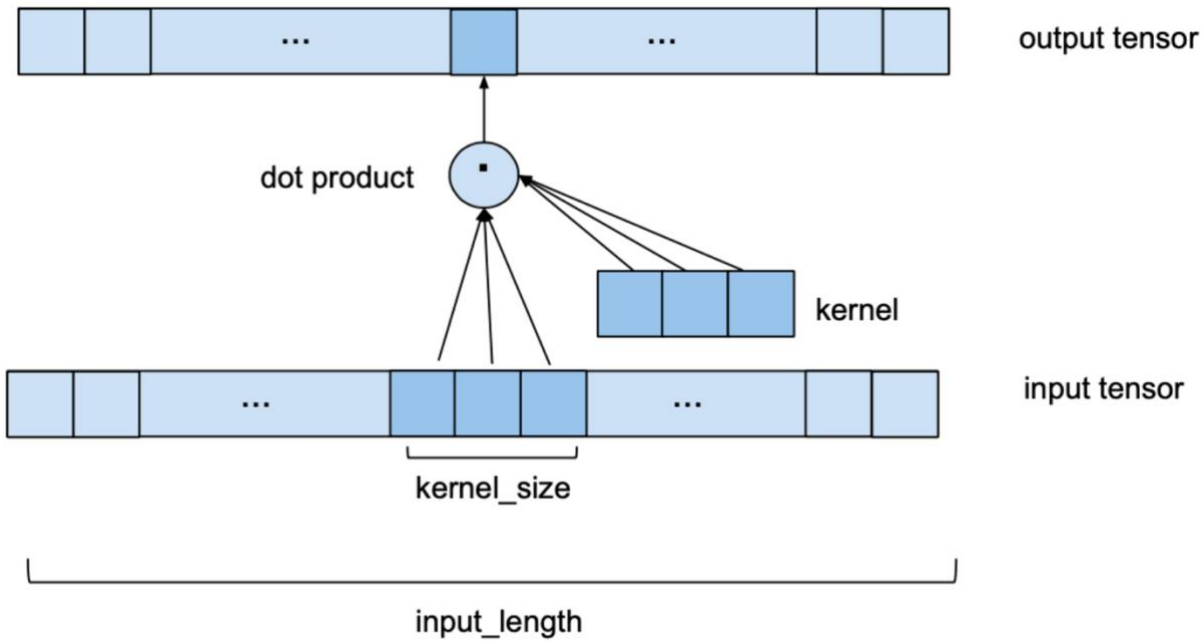
For å utføre annoteringen trenger programmet «annotateBeats.py» en mappe med lydfiler i wav format med navn på formen *filnavn.00001.wav*. Ved kjøring ber programmet om et filnavn, og hvor mange filer som skal leses av. Programmet bruker librosas innebygde beat_track funksjon, for å gi et estimat av tempo, og lydfilen spilles av og brukeren kan bruke tap-tempo funksjonalitet i terminalen for å gjøre et eget estimat og oppgi tempo. Tap-tempo funksjonaliteten er hentet fra TapTempo biblioteket utviklet av Modulo T. (23.09.2016). Programmet viser deretter en spektrogramrepresentasjon av de fem første sekundene fra lydsporet, og ber brukeren oppgi et tidspunkt for en beat som er det første slaget i både takten og en åttetakters periode. Programvaren regner så ut posisjonen til alle tidligere og kommende beats.

Ettersom treningsdataen er nødt til å være av samme type og størrelse som dataen som skal brukes til estimering, deler programmet «createDataset.py» de annoterte filene opp i lydforløp på ti sekunder representert som spektrogrammer i png format. For bruk til trening av maskinlæringsmodellen på egen maskin, inneholder første element i hver rad i datasettet en filsti til tilhørende spektrogram, og andre element er en liste med alle annoteringer. Et nytt

lydforløp på ti sekunder ble hentet ut med startpunkt annethvert sekund. På denne måten ble tre låter konvertert til et datasett med 370 treningsfiler.

4.1 Implementering av maskinlæringsmodell

Det finnes flere eksempler på implementasjon av tempo, taktslag og taktstarts estimering ved hjelp av maskinlæring. «Tempo, beat and downbeat estimation» modellen til ISMIR benytter seg av et *temporal convolutional neural network*, som vil si at modellen benytter en rekke naboverdier til indeksen i input dataen nå den utfører en estimering av output dataen. Dette er effektivt når vi skal finne taktslag og takstartpunkter, fordi det er umulig å si noe om funksjonen til et punkt i en lydfil uten å vurdere det opp mot nærliggende lyddata.

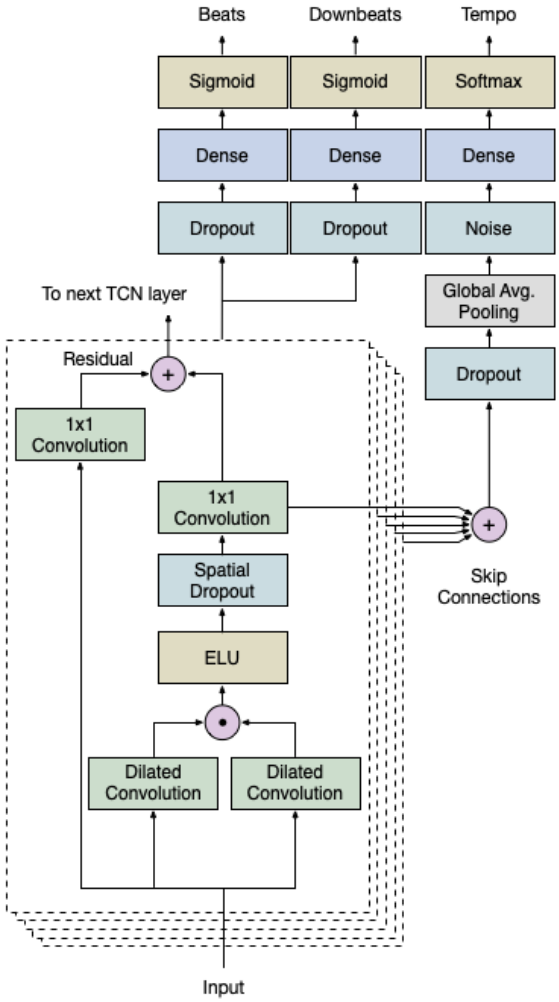


Figur 6 Illustrasjon av et Temporal Convolutional Neural Network (Lässig. i.d.)

Som vist i figur 7 hentes beats og downbeats ut fra nettverket ved hjelp av sigmoid aktiveringsfunksjoner, som vi si at dataen ved output er et flyt-tall mellom 0 og 1, som representerer modellens sikkerhet på at gitt lydområde inneholder en beat eller downbeat. ISMIR modellen har også et output lag som benytter en softmax aktiveringsfunksjon, som gir en liste med tall mellom 0 og 1 som representerer sannsynligheten for at musikken hører til hvert tempo.

Ettersom modellen fra ISMIR ikke hadde real-time funksjonalitet, og manglet estimering av seksjon eller energi, valgte jeg å utvikle en maskinlæringsmodell fra bunnen av. Jeg valgte å benytte Googles maskinlæringsgrensesnitt for python, Tensorflow, ettersom jeg har tidligere erfaring med programmering i python, og det finnes en rekke læringsressurser for Tensorflow biblioteket.

I startfasen av maskinlæringsimplementasjonen ble en enklere modell utviklet med formål å bli kjent med bruk av Tensorflow i python. Denne modellen er vedlagt oppgaven som «tensorflowFunker.py». Koden i denne filen oppretter et datasett bestående av 10 000 tallrekker med ti flyt-tall mellom en og null. Tallene i rekken er enten stigende eller synkende med tilfeldig intervall og startpunkt. Denne modellen viste seg å raskt bli svært treffsikker i estimering. Deretter startet justeringen av denne koden for å tilpasse den til datasettet med annoterte musikkeksempler. Dette viste seg å være langt mere tidkrevende enn forventet, og jeg ble stående fast ved flere punkter i utviklingen grunnet manglende kunnskap om maskinlæringsmodeller. Den ufullstendige koden for maskinlæringsmodellen jeg forsøkte å utvikle ligger som vedlegg til oppgaven, men videre diskusjon om mulige løsninger for implementasjon vil ta utgangspunkt i eksempelkoden fra ISMIR (Davies et al., 2021).



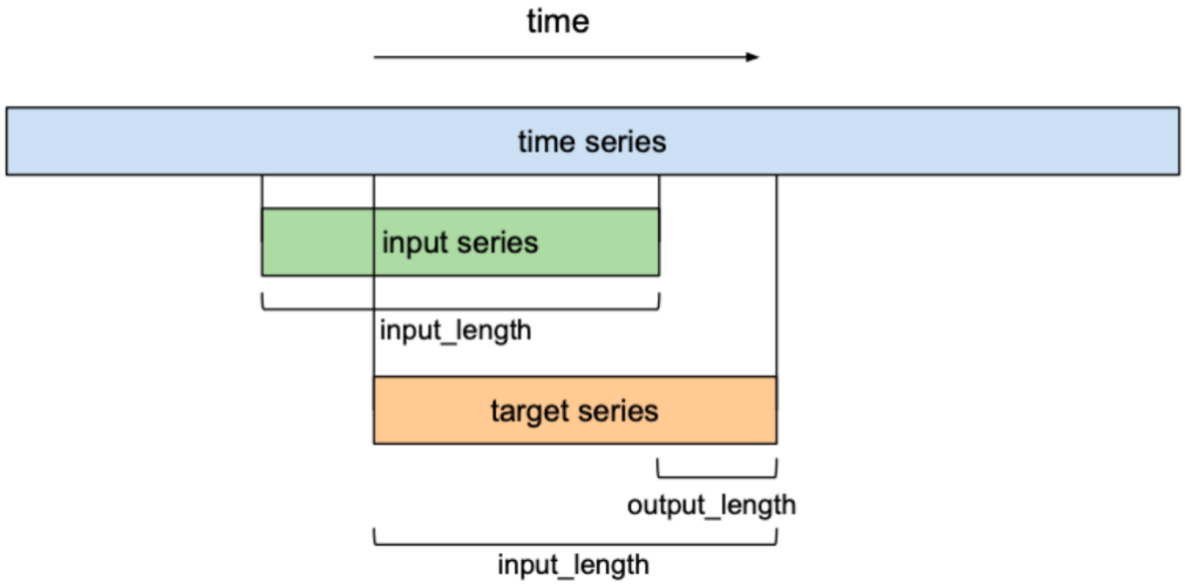
Figur 7 Nettverksstruktur fra ISMIR 2021 Tempo, beat, and downbeat estimation tutorial (Davies et al., 2021)

For å inkludere klassifisering av seksjonstyper i ISMIR modellen, ønsker jeg å bruke de estimerte taktstartpunktene til å dele opp lydklippene i hele takter, og deretter klassifisere disse som en av de fire seksjonstypene. For at modellen skal kunne akseptere de nye oppdelte lydklippene som input, må de ha samme form. Dette har de nødvendigvis ikke etter oppdeling, da musikkeksemplene kan ha forskjellig tempo, og takter vil ha forskjellige varigheter. Spektrogrammene kan da skaleres, slik at lydmateriale av ulik lengde allikevel fyller bilder med samme størrelse.

4.2 Forslag til real-time implementering:

IMSIR modellen har ikke funksjonalitet for real-time musikkanalyse. For at programvaren skal kunne reaktivt endre lyssetting til live musikk, må programvaren kjøre løpende analyse. For å kontinuerlig oppdatere estimatene til modellen må den analysere ny lydinput kontinuerlig. For å opprettholde den overordnede forståelsen av musikalsk struktur må et lenger lydforløp inkluderes i analysen ved hver iterasjon. Dette kan oppnås ved å ha en buffer med tilstrekkelig størrelse, og skrive kortere lydklipp fra input inn i denne bufferen. Etter lytting på lydklipp i forskjellige lengder, virker en buffer på ti sekunder tilstrekkelig for oppnå en viss musikalsk forståelse av struktur. Med denne lengden vil det ofte være minst en overgang mellom to seksjonstyper i bufferen. Hvis de korte lydklippene som skrives inn i bufferen mellom hver iterasjon er for lange, vil neste iterasjon med analyse bli nødt til å vente tilsvarene tid før bufferen er klar til å analyseres. Hastigheten til maskinlæringsmodellen vil påvirke hvor lange lydklippene som skrives til buffer kan være, men jo kortere disse er, jo mer real-time vil analysen være.

Hvis programvaren skal kunne aktivere en ny lyssetting idet det skjer endring i musikken, må modellen kunne forutsi hvordan musikken kommer til å utvikle seg frem i tid. Vi kan da benytte en maskinlæringsmodell utviklet for prognoser. Uten testing er det vanskelig å si hvor treg modellen vil være, men ved å tidsforskyve verdiene modellen forsøker å forutsi, som vist i visualiseringen av prognosefuksjonalitet i figur 8, kan de estimerte verdiene, markert som «output_lenght», inkludere en eller flere fremtidige tidspunkter for beats/downbeats, og forventet seksjonstype. På denne måten kan programvaren aktivere endringen i lyssetting idet neste tiden til neste downbeat blir lik null. Dersom modellen er effektiv i å forutsi neste seksjonstype vil dette kunne føre til en langt mer real-time funksjonalitet.



Figur 8 Visualisering av "forecasting"-prinsippet fra Unit8 (Lässig, i.d.).

4.2 Kommunikasjon med ekstern lyskontroller

For å kunne utnytte estimatene fra maskinlæringsmodellen til å dynamisk endre lyssetting uten innebygd funksjonalitet for lys-design og -programmering, må en kommunikasjonsmetode implementeres. Denne kommunikasjonsmetoden må være rask for å unngå treghet i lysendringene, og kunne kommunisere informasjonen maskinlæringsmodellen har estimert på en effektiv måte. De aller fleste profesjonelle lysbord og lysprogrammeringssoftware har i dag støtte for MIDI-protokollen. Det virker derfor naturlig å benytte seg av denne ved kommunikasjon med eksterne lyskontrollere.

Å sende tempoinformasjon via MIDI er tilsynelatende enkelt. MIDI protokollen har et eget klokkeparameter, som sender pulser for synkronisering i en rate på 24 pulser per fjærdedelsnote (MIDI Association, i.d.). I noen tilfeller har lyskontrollere begrenset MIDI funksjonalitet, og lysbordet MA2 støtter for eksempel kun note-on og note-off beskjeder, og ikke clock beskjeder. Dette lysbordet har derimot mulighet til å knytte bestemte MIDI noter til knapper på kontrolleren, og innebygget funksjonalitet for tap-tempo. En løsning på tempokommunikasjon kan da være å sende note-on og note-off meldinger i tempoet som detekteres av programvaren, men dette vil da ha noe treghet ettersom lysbordet også må tolke tempoet. Det er allikevel sjeldent at klubbmusikk inneholder plutselige drastiske hopp mellom forskjellige tempoer. Derfor kan trolig denne løsningen fungere godt nok selv om kommunikasjonen ikke er svært effektiv.

For å velge lyssettinger som passer til energinivået i musikken, kan programvaren også tilegne de forskjellige seksjonstypene hver sin MIDI note, og sende en note-on beskjed nå endringene skal forekomme. MIDI-noten vil da aktivere funksjonene den er mappet til på lyskontrolleren, og endringen vil skje med svært lite forsinkelse.

6 Diskusjon

Selv om prosjektperioden ikke endte i funksjonell programvare, tyder liknende prosjekter på at en slik tilnærming kan være effektiv. Prosjekter som estimerer tempo, beats og downbeats, og maskinlæringsprosjekter som klassifiserer bilder, ser ut til å ha tilstrekkelig treffsikkerhet til å utføre estimeringene jeg har forsøkt å utrette i prosjektet.

6.1 Datasett

Dette prosjektet går innom flere fagfelt hvor jeg hadde lite tidligere erfaring. Derfor anså jeg det som ønskelig å få en minimumsfunksjonalitet på plass i alle ledd så tidlig som mulig. Ved sette meg litt inn i alle elementene som måtte til for å implementere ønsket funksjonalitet, kunne elementer som kom til å være mer tidkrevende identifiseres tidlig. Derfor valgte jeg å fokusere på hvilke data som skulle inkluderes i datasettet, og utviklingen av annoteringsverktøy. Istedenfor å annotere store mengder musikk, for å skape et fullstendig datasett, ble kun tre sanger annotert for å ha et minimalt datasett med riktig struktur for maskinlæringsmodellen. Planen var da å implementere maskinlæringsmodellen, og deretter gå tilbake og annotere flere lydeksempler. Selv om egen implementering av maskinlæring ikke ble vellykket, kunne et fullstendig datasett vert en ressurs for senere maskinlæringsprosjekter innen MIR.

Verktøyet utviklet for annotering effektiviserer annoteringsprosessen betraktelig, men er lite brukervennlig og ikke er ferdig produkt for generelle brukere. Men ved videre utvikling av prosjektet kan verktøyet allikevel bli benyttet til å lage større datasett som kan brukes i andre prosjekter.

6.2 Maskinlæringsimplementasjon

Resultatene etter avsluttet prosjektperiode bærer tydelig preg av manglende kunnskap innen maskinlæring. Underveis i de tidligere fasene av prosjektet virket implementeringen av maskinlæringsmodellen overkommelig, ettersom det fantes mange tilsynelatende gode læringsressurser på nett. Etter testforsøket med den enklere modellen som ble trent til å estimert om tallrekker er stigende eller synkende, virket det forholdsvis enkelt å adaptere denne modellen til mitt formål. Dette viste seg imidlertid ikke å stemme, og det var flere hindre på veien.

Først av alt var datasettet med musikkeksempler langt mer avansert. Bildedataen måtte oppgis som en flerdimensjonal liste, og inputlaget i modellen måtte tilpasses. Deretter viste det seg at de ulike estimatene modellen skulle gjøre krevde ulike aktiveringsfunksjoner, og nettverket måtte derfor omstruktureres. Ved innlasting av datasettet viste det seg også at modellen brukt i testkoden ikke aksepterte input i forskjellige størrelser, og ettersom musikkeksemplene hadde forskjellig tempo, hadde de også forskjellig lengde på listene med beats. Til slutt oppdaget jeg at andre modeller, som ISMIRs «Tempo, Beat and Downbeat Estimation» (2021), benyttet seg av *temporal convolutional neural networks.* Dette ville tilsynelatende være langt mere effektivt i denne typen estimering enn modellen som ble brukt i testkoden, men dette hadde jeg ikke tid til å implementere innenfor tidsrammene til prosjektet.

Alle disse problemene ser ut til å ha løsninger som gjør implementasjonen av ønsket funksjonalitet mulig, men var altså for omfattende å utføre med manglende tidligere erfaring og kunnskap innen maskinlæring.

7 Konklusjon

I denne oppgaven har jeg sett nærmere på mulighetene for å implementere maskinlæringsmodeller på musikk for å hente ut data som kan benyttes innen dynamiske lysshow til klubbmusikk. Prosjektet førte ikke til et fullstendig produkt, slik som opprinnelig var målet, men har allikevel vert en kartlegging av hvilke parameter som bør inkluderes i en slik modell, metoder for å annotere musikk og formatere datasett, og hvordan output data fra et slikt nettverk kan oversettes til lysdesign.

For å effektivisere annoteringsprosessen ble det utviklet et verktøy for annotering, som automatisk plasserte markører for beats og downbeats ved å spille av korte utdrag av lydeksemplene og be brukeren om enkel input. Dette fører til langt raskere annotering om muliggjør kreasjon av et langt større datasett.

I forsøket på å implementere maskinlæring har det også blitt kartlagt flere mulige implementasjoner av en slik modell. Det virker hensiktsmessig å benytte et *temporal convolutional neural network*, som tar en samling nærliggende verdier på tidsaksen i betraktning ved estimering av hver verdi i output. Forskjellige aktiveringsfunksjoner bør benyttes for ulike formål, sigmoid funksjoner for estimering av sannsynlighet for at en tidsramme inneholder en beat, og softmax funksjon for å klassifisere generelle aspekter som tempo.

Til slutt har det blitt reflektert rundt prosessen, og årsaker til manglende resultat har blitt identifisert som mangel på viktige forkunnskaper som viste seg å være viktige med tanke på gjennomføring av prosjektets målsetting. Allikevel har mulige løsninger blitt diskutert, og ønsker overordnet struktur på programvaren er skissert. Dette legger en god grunnmur for videre arbeid med prosjektet.

Referanser

Disclosure. (28.08.2020). Expressing What Matters. På *ENERGY (Deluxe)*. Island records

Fuentes, Magdalena. (05.11.2021). gtzan_tempo_beat. *TempoBeatDownbeat.* Hentet 21.05.2023 fra: https://github.com/TempoBeatDownbeat/gtzan_tempo_beat

Kaplan, R., & Kaplan, S. (1989). The experience of nature: A psychological perspective. Cambridge University Press. Hentet 19.05.2023 fra: https://www.hse.ru/data/2019/03/04/1196348207/%5BRachel_Kaplan,_Stephen_Kaplan%5D_The_Experience_of_(b-ok.xyz).pdf

Lässig, Francesco. (i.d.). Temporal Convolutional Networks and Forecasting. *Unit8.com*. Hentet 21.05.2023 fra: https://unit8.com/resources/temporal-convolutional-networks-and-forecasting/

MA Lighting. (i.d.). Sound Input Window. *Help Pages.* Hentet 20.05.2023 fra: https://help2.malighting.com/Page/grandMA2/ost_sound_input_window/en/3.3

Matthew E. P. Davies, Sebastian Böck, Magdalena Fuentes. (2021). Tempo, Beat and Downbeat Estimation. *Github.io*. Hentet 20.05.2023 fra: https://tempobeatdownbeat.github.io/tutorial/intro.html

MIDI Association. (i.d.). MIDI 1.0 Detailed Specification. *Official MIDI Specifications.* Hentet 21.05.2023 fra: https://www.midi.org/specifications/midi1-specifications/m1-v4-2-1-midi-1-0-detailed-specification-96-1-4

Modulo T. (23.09.2016). taptempo.py. *Github.com.* Hentet 28.04.2023 fra: https://github.com/defaultxr/taptempo.py

Tzanetakis, George and Essl, Georg and Cook, Perry. (2001). Automatic Musical Genre Classification Of Audio Signals. *The International Society for Music Information Retrieval.* Hentet 21.05.2023 fra: https://www.tensorflow.org/datasets/catalog/gtzan

Vedlegg

Kildekode og datasett utviklet underveis i prosjektet ligger vedlagt i en separat zip-fil til denne oppgaven. Lydfilene brukt til å skape datasettet er ikke vedlagt pga. opphavsrett.

Kildekode

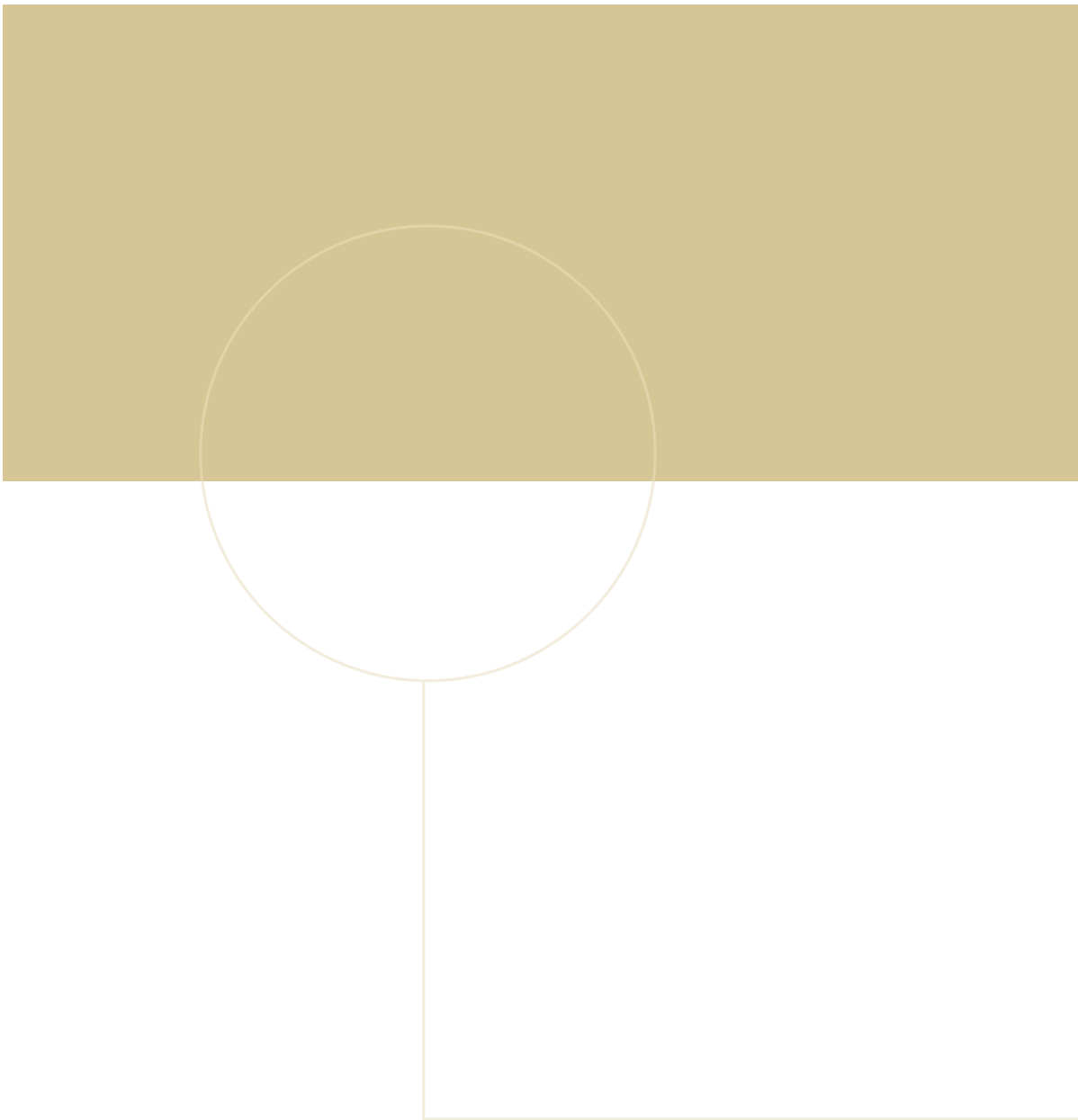
Kildekoden utviklet underveis i prosjektet ligger i mappen «Kode» og inneholder følgende:

annotateBeats.py	Inneholder hovedprosessen som brukes under annoteringen av musikkeksempler.
annotateUtils.py	Inneholder funksjonene som benyttes i «annotateBeats.py» og «createDataset.py».
createDataset.py	Formaterer annoterte lydfiler til et komplett datasett.
tensorflowMusic.py	Inneholder ufullstendig forsøk på å implementasjon av maskinlæringsmodell.
tensoflowFunker.py	Inneholder en enkel maskinlæringsmodell brukt til testing tidlig i implementeringsfasen.

Datasett

Datasettet produsert til prosjektet ligger vedlagt i mappen «Datasett», som inneholder følgende:

Csv	Mappe med annoteringer til lydfilene i csv format.
Spektrogram	Mappe med bilderepresentasjoner av ti sekunders utdrag av lydfilene, i png format.
datasett.csv	Datasett med en rad med annoteringer for hvert spektrogram, i csv format.



NTNU

Kunnskap for en bedre verden