Prediktion av känslor

Med integration av Spotify



David Norman

EC Utbildning

Projekt arbete

202410

# Abstract

Projektet utförskar användningen av Deep learning för analysering av röstinspelning och prediktering av användarens humör baserat på röstens tonhöjd, vilket kopplas till personens rekomenderade musikspellistor på spotify. Systemet implementerar FastAPI som är en REST API-tjänst och ett Tkinter baserad användar gränssnit för att spela in rösten och printa ut resultatet. En förtränad CNN-modell används för att klassificera rösten i tre humör kategorier: calm, energetic och happy. Projektet visar en fungerande prototyp och diskuterar möjligheter för förbättring.

# Förkortningar och Begrepp DL – Deep Learning CNN – Convolutional neural network

MFCC - Mel-frequency cepstral coefficients

Innehållsförteckning

[Abstract 2](#_Toc181271884)

[Förkortningar och Begrepp DL – Deep Learning CNN – Convolutional neural network 3](#_Toc181271885)

[1 Inledning 1](#_Toc181271886)

[2 Teori 2](#_Toc181271887)

[2.1 Deep Learning för ljudanalys 2](#_Toc181271888)

[2.2 CNN 2](#_Toc181271889)

[2.3 MFCC 2](#_Toc181271890)

[2.4 FastAPI 2](#_Toc181271891)

[2.5 Applikation 2](#_Toc181271892)

[3 Metod 3](#_Toc181271893)

[3.1 Applikation 3](#_Toc181271894)

[3.2 Pipeline / API 4](#_Toc181271895)

[3.3 Modell 5](#_Toc181271896)

[3.4 Agil arbetsmetodik 6](#_Toc181271897)

[4 Resultat och Diskussion 7](#_Toc181271898)

[4.1 Modellens resultat 7](#_Toc181271899)

[4.2 Användargränssnitt 7](#_Toc181271900)

[4.3 Pipelinen 7](#_Toc181271901)

[4.4 Systemfunktionalitet 7](#_Toc181271902)

[5 Slutsatser 8](#_Toc181271903)

[6 Självutvärdering 9](#_Toc181271904)

[Källförteckning 10](#_Toc181271905)

# Inledning

Deep Learning har blivit en central och snabbväxande teknologi under de senaste åren inom multimedia applikationer med användningsområden som röstigenkänning, emotionell analys och rekommendationssystem. Deep Learning gör det möjligt att automatisera analyser som kan utföras snabbt och effektivt.

I detta projekt utforskar vi hur just Deep Learning kan användas för att analysera och prediktera röstigenkänning och användarens emotionella status för att sedan rekommendera en spellista.

Projektet syftar till att skapa en innovativ och användarvänlig lösning där informationen används för att rekommendera personliga musikförslag via Spotify.

1. Kan Deep Learning användas för just röstigenkänning.
2. Kan vi skapa en applikation för att prediktera korrekt emotionell status och starta en Spotify spellista.

# Teori

## Deep Learning för ljudanalys

Det har visat sig i detta projekt att Deep Learning varit mycket effektiv med just analysering av ljuddata, i projektet användes en CNN modell som sorterade in ljuddata i specifika mönster för att senare identifieras och klassificeras.

## CNN

Konvolutionella neurala nätverk (CNN) är en typ av artificiellt neuralt nätverk, framtaget för att analysera bild och ljuddata. CNN fungerar genom att extrahera funktioner från inmatade data och identifiera specifika mönster som används för klassificering. Dessa nätverk använder sig av en serie lager där konvolutionslager är centrala för att extrahera viktiga funktioner från data.  
Ett konvolutionslager består av en uppsättning filter som läggs på data för att generera aktiverings kartor. Genom att analysera dessa aktiveringskartor kan man identifiera nätverkets mönster som är relevanta för specifika klasser.

I projektet tränades en CNN modell för att upptäcka relevanta mönster relaterat till olika känslotillstånd i ljuddata, så som glad, energisk och överraskad. Modellen kollar efter tonhöjd, intensitet och variation i ljuddata som blir avgörande för att klassificera känslorna. CNN modellen är särskilt lämplig för denna typ av analys eftersom modellen kan fånga spatiala beroenden i ljuddata, vilket gör det möjligt för modellen att skapa en representation av känslomässiga mönster.

## MFCC

Mel-frequency cepstral coefficients. Är en metod för att extrahera funktioner för ljudsignaler som är vanligt förekommande i röst och ljudanalys. MFCC omvandlar tidsdomänsignalen till frekvensdomän, vilket gör det möjligt för modellen att analysera ljudets spektrala innehåll.  
Genom att använda logaritmisk skalning och omvandla data till Mel-skala som bättre liknar hur människor uppfattar ljud, blir det lättare för modellen att identifiera skillnader mellan olika känslor i data.

MFCC ger en representation som är komprimerad och mer robust mot brus och variationer i data, vilket förbättrar modellens prestanda och säkerhet vid känsloigenkänning.

## FastAPI

FastAPI är en modern Python modul för att bygga API tjänster. Det är känt för att vara snabbt och enkelt att använda och det stödjer asynkrona operationer vilket är viktigt för att skapa responsiva och snabba tjänster. I projektet används FastAPI för att hantera all kommunikation mellan Applikationen och CNN modellen. När ljudfilen skapas via applikationen så skickas den via en pipeline som använder FastAPI till modellen för att analyseras och sedan kommunicerar pipelinen med spotify API för att spela upp en spellista baserat på det predikterade känslotillståndet.

## Applikation

TKinter och Custom TKinter används för att bygga grafiska användar-gränssnitt i python för att lättare kunna visualisera och bygga applikationer för diverse produkter. I projektet används TKinter och Custom TKinter för att bygga ett gränssnitt som hjälper användaren att länka sin spotify till pipelinen och tillåter användaren att spela in en 5 sekunder lång ljudfil som sedan används för att analysera och prediktera känslotillståndet.

# Metod

Arbetet utfördes i fyra olika steg

* Applikation

Applikationen skapades med modulen TKinter och Custom TKinter i python för att tillåta användaren att spela in och prediktera sin röst utan att behöva skriva diverse koder.

* Pipeline

En pipeline skapades för att tillåta applikationen att kommunicera med både modellen och APIt, i pipelinen användes bland annat FastAPI och Spotipy modulerna.

* Modell

En CNN modell skapades och kalibrerades med hjälp av audio augmentation för att kunna prediktera röstinspelningarna.

* Inergrering

De tre föregående stegen sätts samman med hjälp av pipelinen.

## Applikation

Applikationen skapades i pythons egna modul Tkinter.

Tanken med applikationen var att ha ett grafiskt interface som tillåter användaren att öppna och länka sin Spotify till piplinen, samt spela in och analysera sin röst för att pipelinen sedan analyserar rösten och rekommenderar sedan en Spotify spellistan.

*Nedantill visas hur applikationen spelar in och kontaktar modellen för analys*

A screen shot of a computer program

Description automatically generated

## Pipeline / API

Ett API skapades som analyserar ljudfiler för att förutsäga användarens emotionella status och föreslå en passande Spotify-spellista. Projektet använder Spotipy för att autentisera mot Spotify API och integrera olika spellistor. För ljudanalys nyttjas Librosa, där MFCC-egenskaper (Mel-frequency cepstral coefficients) extraheras, då de är viktiga för att identifiera timbre, vilket är kopplat till humör. En förtränad TensorFlow-modell används för att analysera dessa ljudegenskaper och förutsäga humörkategorin. Beroende på resultatet returnerar API en Spotify-länk till en matchande spellista. Lösningen möjliggör att automatiskt skapa en anpassad musikupplevelse som speglar användarens humör och ger en interaktiv och användarvänlig rekommendationsfunktion.

*Nedan visas pipelinens kod*

*A screen shot of a computer screen

Description automatically generated*

*A screen shot of a computer program

Description automatically generated*

## Modell

Modellen som användes var en en-dimensionell Convolusional Neural Network som är anpassad för att ta emot och hantera en specifik ljuddata, i detta fall användes Wav filer som modellen skulle prediktera känslor från baserat på ljudfilens tonhöjd.

*Nedantill visas modellens uppbyggnad*

A screen shot of a computer program

Description automatically generated

## Agil arbetsmetodik

Projektarbetet delades upp i tre delar där vi alla hade sitt eget ansvarsområde,

David arbetade med Applikationen och samarbetade med integrationen.

Muhannad arbetade med pipelinen och samarbetade med integrationen.

Parsan arbetade med modellen och datasett samt samarbetade med integrationen.

Ming arbetade med applikationen och modellen samt samarbetade med integrationen.  
  
Vi hade möten minst en gång i veckan via teams, samt daglig checkup via teams chatten.

# Resultat och Diskussion

## Modellens resultat

Med vår CNN modell och de datasett vi använt oss utav fick vi det följande resultat när vi tränade vår modell.

*Nedan visar accuracy och loss*



Trots högt accuracy resultat så predikterar modellen endast två variabler. ”calm” eller ”surprised” detta går att anta att modellen kan vara överanpassad eller att modellen har någon outliers som kan fästa modellen i dessa två variabler.

## Användargränssnitt

Tkinter applikationen ger användaren ett enkelt sätt att länka sin spotify, spela in och analysera sin röst samt se resultatet av prediktionen. Användargränssnittet (applikationen) är gjord på ett simpelt sett för att ge en så tydlig användarmöjlighet för användare utanför projektet. Två knappar finns tillgängliga var ena öppnar spotify direkt och den andra startar röstanalysens process. En meny har gjorts i form av en tool bar med möjligheten att länka sin spotify och öppna spotify, samt en about tab som skriver ut dem som har arbetat med projektet.

## Pipelinen

Pipelinens funktionalitet var att hjälpa kommunikationen mellan Applikationen, modellen och Api’t vilket den kunde utföra utan större problem.

## Systemfunktionalitet

Vi lyckades integrera en DL-modell med både desktopapplikation och en REST API för att prediktera emotionell status hos användaren, resultatet varierade mellan ”calm” och ”surprised” vilket gör projektet funktionellt men inte perfekt.

# Slutsatser

* Kan Deep Learning användas för just röstigenkänning?

Ja. I projektet lyckades vi skapa en Deep Learning modell som kunde hantera och analysera ljuddata för att prediktera emotionell status hos användaren.

* Kan vi skapa en applikation för att prediktera korrekt emotionell status och starta en Spotify spellista.

Vi skapade en applikation som kunde med hjälp av DL modellen och pipelinen prediktera den emotionella status hos användaren, dock var prediktionen fast vid två variabler oavsett vilken emotionell status vi försökte få fram. Detta kan vara pga ”overfitting” eller ”noise” i augmenteringen i modellen.

# Självutvärdering

1. Utmaningar du haft under arbetet samt hur du hanterat dem.

Största utmaningen jag personligen haft var integrationen mellan alla tre delar, då alla hade eget ansvarsområde hade ingen hundra koll på vad de andra hade gjort och varför dem gjort just så.

1. Vilket betyg du anser att du skall ha och varför.

Jag anser att jag skall få godkänt på kursen då jag uppfyllt de kraven samt har en god förståelse över projektet.

1. Något du vill lyfta fram till Antonio?

Rolig kurs 10/10 har varit utmanande men roligt.

# Källförteckning

Datasett var hämtat av Parsan Amani från

[Kannada\_Emotions](https://www.kaggle.com/datasets/pragyanaianddsschool/kannada-emotions)

[Ravedess Dataset](https://www.kaggle.com/datasets/descipro/ravedess-dataset)

[Toronto emotional speech set (TESS)](https://www.kaggle.com/datasets/ejlok1/toronto-emotional-speech-set-tess)

[Emotional Voice Conversion: Theory, Databases and ESD](https://hltsingapore.github.io/ESD/download.html)