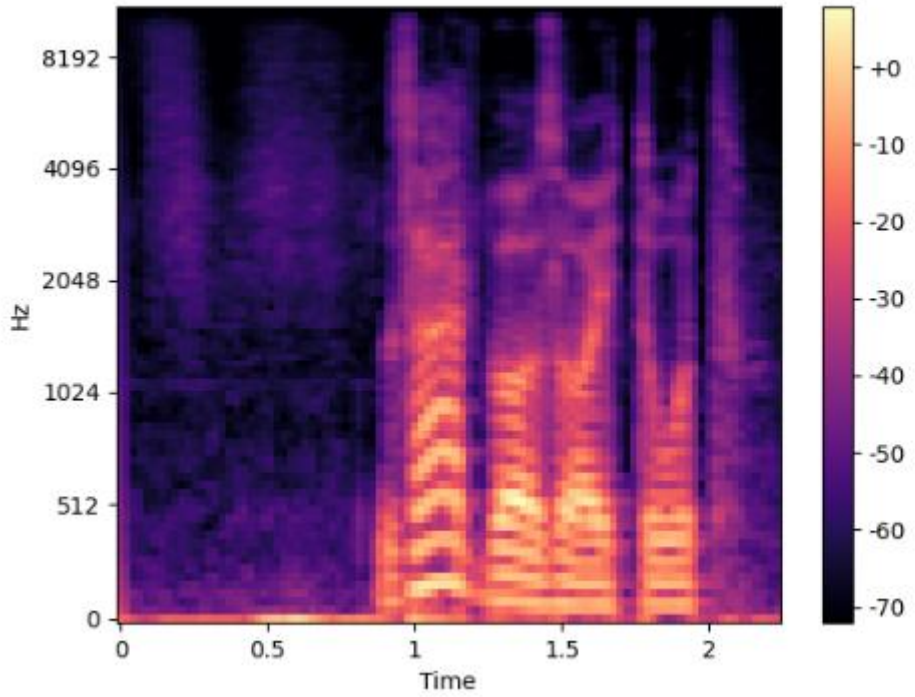
Balogh László
DFQXYU

Képfeldolgozás alap algoritmusai féléves feladat

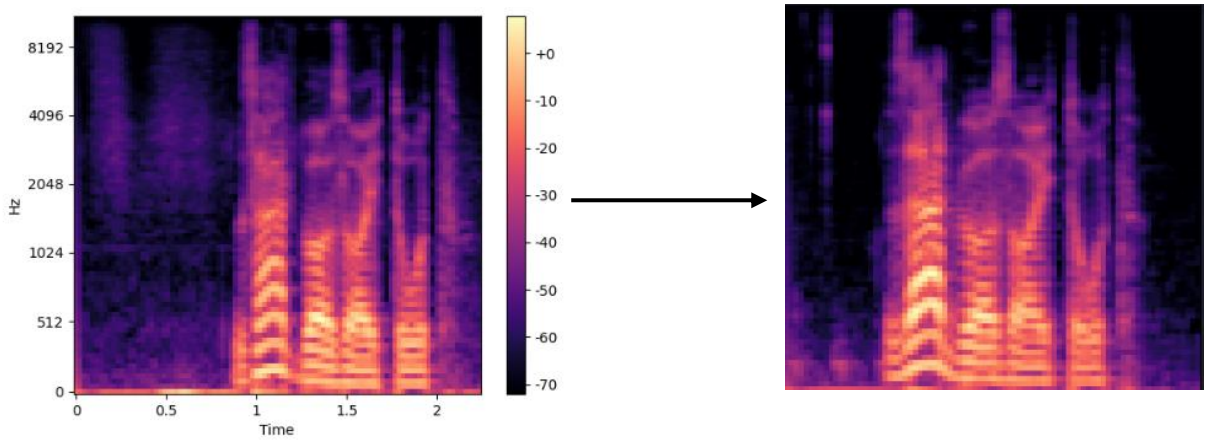
Megvalósítás

A konvolúciós háló betanításához, képekre van szükség. Ezeket a képeket úgy állítottam elő, hogy az elkészített utasításokat elmentettem „wav” formátumba, majd beolvasom őket a Librosa könyvtár segítségével. A hangfájlok utasításokként vannak elmentve mappákba. A beolvasás során végig megyek az audio fájlokat tartalmazó mappákon, beolvasom őket, és minden kategóriának létrehozok egy tömböt, ahol eltárolom őket. A mintavételezési frekvenciát a Librosa 22050 Hz-re konvertálja. Ezután végigiterálok az összes tömbön, ami az audio fájlokat tartalmazza, és egyesével Mel Spectogramokat készítek, ezeket szintén kategóriánként tömbökbe mentem. Ezt követőn a matplotlib könyvtár segítségével, kép formájában elmentem a Mel Spectogramokat.



Így néz ki a matplotlib könyvtár által lementett Mel Spectogram. Ahogy ez látszik is, rengeteg fölösleges adat van rajta, gondolok itt a fehér területekre, a skálákra, valamint a skála beosztásaira. Ráadásul ezek mind megegyeznek az összes képen, tehát a neurális háló

betanításakor fennállna az a veszély, hogy rossz jellemzőkre tanulna rá, rossz egyezőségeket venne figyelembe. Hogy ezt elkerüljem az összes elmentet spectogramot, az OpenCV könyvtár segítségével beolvasom, majd a lényegi részt kivágom és elmentem őket, szintén utasítás szerint rendezve. Az alap spectogram képek 640 x 480-as felbontásúk, a körbevágott képek 398 x 370 felbontásúak.



7 féle utasítást deffiniáltam:

- kapcsold fel a lámpát
- oltsd le a villanyt
- mennyi az idő
- mesélj egy viccet
- kapcsold be a tévét
- állítsd le a légkondit
- zárd be az ajtót

Mindegyik utasításból 50 darab van a tanító dataset-ben, 5 darab a validációs dataset-ben és 5 darab a teszt dataset-ben.

Egy Seguentail hálózatot hoztam létre Keras-ban, aminek 4 darab Conv2D rétege van, mindegyik tangens aktivációs függvénnyel. 3 darab Dense rétege van, abból kettőnek relu az aktivációs függvénye, az utolsónak pedig softmax az aktivációs függvénye, valamint ennek az utolsó rétegnek 7 darab kimenete van, mivel 7 darab utasítás közül kell döntést hoznia, hogy melyikre hasonlít a leginkább. A loss értékét a categorical_crossentropy függvénnyel számoltam. 35 epoch-ot futtattam le, és tízesével adtam be a képeket a hálózatnak. Maga a

betanítás GPU gyorsítással történt, köszönhetően annak, hogy a TensorFlow-nak van GPU megvalósítása. Ennek köszönhetően sokkal gyorsabban történt meg a betanítás.

A TensorFlow környezetének a kialakításához, az Anaconda package manager-t használtam. Az Anaconda segítségével, telepítettem a OpenCV-t a környezethez, a TensorFlow GPU megvalósítását, valamint a cuDNN-t, ami a mély neurális hálózatok GPU-n történő tanításához kell. 3.10.8-as Python verziót és 11.7-es verziójú CUDA-t használtam. A programot Visual Studio Code-ban és azon belül is Jupyter Notebook-ban írtam.

Összegzés

A hálózat pontossága a tanítási adatbázison 1.00, a loss értéke 0,0010 lett, ami felveti az a kérdést, hogy a hálózat esetleg túltanult. Ezt a kérdést megcáfolni nem tudom, viszont az összes tesztelési adattal kipróbálva egyszer sem hibázott a becslésnél a hálózat, ami jó működést enged feltételezni.