בס”ד

Project Book

***Data Preparation:***

The process of preparing the data was split into several sub-tasks.

* Downloading the data – **DataDownloader.py**

Our data is consisted by ~1500 videos from YouTube. In order to download the data, we had a JSON file containing all the videos URLs. The JSON has both videos of Duet and Solo music pieces, with 15 classes of musical instruments:

{'Banjo', 'Cello', 'Drum', 'Guitar', 'Harp', 'Harmonica', 'Oboe', 'clarinet', 'Piano', 'xylophone', 'Saxophone', 'Trombone', 'Trumpet', 'Violin', 'Flute', 'Accordion', 'Horn', 'tuba', 'erhu'}

When similar instruments are considered the same (such as piano and xylophone).

In order to download the videos, we used the ‘*pytube’* library which helps with downloading videos from YouTube efficiently.

We downloaded all the videos and separated them efficiently and hierarchically by type (Duet or Solo) and class.

* Splitting the data – **DataSplit.py**

In order to have enough training examples we had to split the videos into parts of 10 seconds each (as advised by the authors of the article).

Using *movipy editor’s VideoFileClip,* we ran a python script to scan iteratively all the downloaded videos and split each one of them into 10 seconds chunks. Each chunk was saved in a separated directory with the class labels of the instruments, 80 frames of the sub-videos (with 8 fps) and the audio of the sub-video.

* Copying the data to a different location

In order to have valuable data labels we used the script ***ArrangeAndMoveDirs.py*** In order to save the labels as the videos’ directory name instead of its YouTube’s name, and in a different location so we have copy of the original data.

* **CreateImageDir.py**

After having all the raw data we wanted to use the given ResNet-101 by the authors of the article in order to have pre-detection of the musical instruments so we can save time while running the ML model. We created a new directory in each chunk called *images,* which holds the .jpg frames from the split video.

* **EnumerateDirs.py**

We scanned all the videos and numbered them, so we have distinct videos ids per each.

* **ExtractFramesFromNpy.py**

We scanned all the chunks and each .npy frame was saved as jpg in the Images dir.

* **DetectData.py**

After having all the .jpg images we could run the resnet-101 detection on the data. We used *subprocess* to run the detections in parallel.

We resulted with a list of bounding boxes and labels for each frame with the musical instruments in it.