Apache MXNet is an open-source deep learning software framework, used to train and deploy deep neural networks. It is scalable, allowing for fast model training and supports a flexible programming model and multiple programming languages.

Compared to TensorFlow, MXNet **has a smaller open source community**. Improvements, bug fixes, and other features take longer due to a lack of major community support. Despite being widely used by many organizations in the tech industry, MxNet is not as popular as Tensorflow.

* **Must use Python 3.8**. Download miniconda with python 3.8 or make an environment variable with python 3.8 installed

Using Python 3.8 because we will use a zipfile(line no 175 in splidnn.py) for training and “zipfile.Path is supported only for python 3.8 n later”

[Here](https://docs.python.org/3/library/zipfile.html) is the zipfile documentation for python 3.8

[Here](https://docs.python.org/3.7/library/zipfile.html?highlight=zipfile#module-zipfile) is the zipfile documentation for Python 3.7

[Here](https://stackoverflow.com/questions/74391027/attributeerror-module-zipfile-has-no-attribute-path) is the error that occurs with zipfile.Path using Python 3.7

* Now installing mxnet with python 3.8 might throw an error related to numpy. [Here](https://stackoverflow.com/questions/74419220/error-while-installing-mxnet-using-pip-in-python-3-8-and-3-10) is the error.

**Update the visual studio code**. Follow [this](https://stackoverflow.com/questions/64261546/how-to-solve-error-microsoft-visual-c-14-0-or-greater-is-required-when-inst) tutorial.

* Now data preprocessing. Make sure that each of the audio clips is of same length.

Use ffmpeg to clip the audio.

Code for clippimg audio using ffmpeg:

* one audio clip at a time

ffmpeg -ss [start time] -t [end time] -i [input filename] -acodec [outputname]

ffmpeg -ss 0 -t 3 -i HI\_(1).mp3 -acodec copy out1.mp3

* Covert all audios in a folder at once. Run this from within the folder containing all the audio clips

for %a in ("\*.mp3") do ffmpeg -ss 0 -t 3 -i "%a" -acodec copy "out%~na.mp3"

* Model Training

python splidnn.py train -e [no of epochs] [dataset name zipped folder]

python splidnn.py train -e 5 archive.zip

* Model testing:

Make sure to change the clip size in “line 202” to the size of your input audio else it will give a shape error. We had each audio clips of 3 secs hence “sf.samplerate\*3”

python splidnn.py infer --model training/model\_epoch16 Check/outcommonh\_voice(1).mp3