ECS271_project

December 6, 2022

Image generation using Speech emotion recognition.

To run the code, we need to follow these steps:

- Run the cell which installs the libraries (Cell 1)
- Run the cell which imports the libraries and performs some basic configurations (Cell 2)
- Run the cells which define the necessary helper and training functions for VQGAN+CLIP (Cells 3, 4, 5, 6 and 7)
- Run the cells which defined methods for extracting features from the audio file and loading the dataset(Cells 8, 9 and 11)
- We will skip the cells which train the models, since they are already trained and will run the 2nd last and the last cell.
- Speak for 3 seconds after pressing ENTER on the prompt.
- The speech will be converted to text and emotions and an image will be generated after 300 interations.

First we need to install the required dependencies. These include installing pytorch and tensorflow for training and loading the trained models. We also need to install the VQGAN and CLIP trained models. Lastly, we need dependencies for converting speech to text and extracting features from audio files.

[ ]:
```
# ML Libraries
!pip install --user torch==1.9.0 torchvision==0.10.0 torchaudio==0.9.0␣
 ↪torchtext==0.10.0 pytorch-lightning
!pip install tensorflow
!pip install sklearn

# Trained models
!git clone https://github.com/openai/CLIP
!pip install taming-transformers
!git clone https://github.com/CompVis/taming-transformers.git

# Speech recognition and feature extraction
!pip install ftfy regex tqdm omegaconf
!pip install kornia
!pip install imageio-ffmpeg
!pip install einops
!pip install pynvml
!pip install librosa soundfile numpy SpeechRecognition pydub
```

```
!pip install setuptools==59.5.0
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-
wheels/public/simple/
Collecting torch==1.9.0
  Downloading torch-1.9.0-cp38-cp38-manylinux1_x86_64.whl (831.4 MB)
     |                  | 831.4 MB 14 kB/s
Collecting torchvision==0.10.0
  Downloading torchvision-0.10.0-cp38-cp38-manylinux1_x86_64.whl (22.1 MB)
     |                  | 22.1 MB 1.5 MB/s
Collecting torchaudio==0.9.0
  Downloading torchaudio-0.9.0-cp38-cp38-manylinux1_x86_64.whl (1.9 MB)
     |                  | 1.9 MB 42.1 MB/s
Collecting torchtext==0.10.0
  Downloading torchtext-0.10.0-cp38-cp38-manylinux1_x86_64.whl (7.6 MB)
     |                  | 7.6 MB 49.6 MB/s
Collecting pytorch-lightning
  Downloading pytorch_lightning-1.8.3.post1-py3-none-any.whl (798 kB)
     |                  | 798 kB 40.8 MB/s
Requirement already satisfied: typing-extensions in
/usr/local/lib/python3.8/dist-packages (from torch==1.9.0) (4.1.1)
Requirement already satisfied: pillow>=5.3.0 in /usr/local/lib/python3.8/dist-
packages (from torchvision==0.10.0) (7.1.2)
Requirement already satisfied: numpy in /usr/local/lib/python3.8/dist-packages
(from torchvision==0.10.0) (1.21.6)
Requirement already satisfied: tqdm in /usr/local/lib/python3.8/dist-packages
(from torchtext==0.10.0) (4.64.1)
Requirement already satisfied: requests in /usr/local/lib/python3.8/dist-
packages (from torchtext==0.10.0) (2.23.0)
Collecting tensorboardX>=2.2
  Downloading tensorboardX-2.5.1-py2.py3-none-any.whl (125 kB)
     |                  | 125 kB 67.9 MB/s
Collecting lightning-utilities==0.3.*
  Downloading lightning_utilities-0.3.0-py3-none-any.whl (15 kB)
Requirement already satisfied: packaging>=17.0 in /usr/local/lib/python3.8/dist-
packages (from pytorch-lightning) (21.3)
Requirement already satisfied: PyYAML>=5.4 in /usr/local/lib/python3.8/dist-
packages (from pytorch-lightning) (6.0)
Requirement already satisfied: fsspec[http]>2021.06.0 in
/usr/local/lib/python3.8/dist-packages (from pytorch-lightning) (2022.11.0)
Collecting torchmetrics>=0.7.0
  Downloading torchmetrics-0.11.0-py3-none-any.whl (512 kB)
     |                  | 512 kB 74.4 MB/s
Collecting fire
  Downloading fire-0.4.0.tar.gz (87 kB)
     |                  | 87 kB 7.6 MB/s
Requirement already satisfied: aiohttp!=4.0.0a0,!=4.0.0a1 in
/usr/local/lib/python3.8/dist-packages (from fsspec[http]>2021.06.0->pytorch-
```

```
lightning) (3.8.3)
Requirement already satisfied: async-timeout<5.0,>=4.0.0a3 in
/usr/local/lib/python3.8/dist-packages (from
aiohttp!=4.0.0a0,!=4.0.0a1->fsspec[http]>2021.06.0->pytorch-lightning) (4.0.2)
Requirement already satisfied: multidict<7.0,>=4.5 in
/usr/local/lib/python3.8/dist-packages (from
aiohttp!=4.0.0a0,!=4.0.0a1->fsspec[http]>2021.06.0->pytorch-lightning) (6.0.2)
Requirement already satisfied: attrs>=17.3.0 in /usr/local/lib/python3.8/dist-
packages (from aiohttp!=4.0.0a0,!=4.0.0a1->fsspec[http]>2021.06.0->pytorch-
lightning) (22.1.0)
Requirement already satisfied: charset-normalizer<3.0,>=2.0 in
/usr/local/lib/python3.8/dist-packages (from
aiohttp!=4.0.0a0,!=4.0.0a1->fsspec[http]>2021.06.0->pytorch-lightning) (2.1.1)
Requirement already satisfied: yarl<2.0,>=1.0 in /usr/local/lib/python3.8/dist-
packages (from aiohttp!=4.0.0a0,!=4.0.0a1->fsspec[http]>2021.06.0->pytorch-
lightning) (1.8.1)
Requirement already satisfied: aiosignal>=1.1.2 in
/usr/local/lib/python3.8/dist-packages (from
aiohttp!=4.0.0a0,!=4.0.0a1->fsspec[http]>2021.06.0->pytorch-lightning) (1.3.1)
Requirement already satisfied: frozenlist>=1.1.1 in
/usr/local/lib/python3.8/dist-packages (from
aiohttp!=4.0.0a0,!=4.0.0a1->fsspec[http]>2021.06.0->pytorch-lightning) (1.3.3)
Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in
/usr/local/lib/python3.8/dist-packages (from packaging>=17.0->pytorch-lightning)
(3.0.9)
Requirement already satisfied: protobuf<=3.20.1,>=3.8.0 in
/usr/local/lib/python3.8/dist-packages (from tensorboardX>=2.2->pytorch-
lightning) (3.19.6)
Requirement already satisfied: idna>=2.0 in /usr/local/lib/python3.8/dist-
packages (from
yarl<2.0,>=1.0->aiohttp!=4.0.0a0,!=4.0.0a1->fsspec[http]>2021.06.0->pytorch-
lightning) (2.10)
Requirement already satisfied: six in /usr/local/lib/python3.8/dist-packages
(from fire->lightning-utilities==0.3.*->pytorch-lightning) (1.15.0)
Requirement already satisfied: termcolor in /usr/local/lib/python3.8/dist-
packages (from fire->lightning-utilities==0.3.*->pytorch-lightning) (2.1.1)
Requirement already satisfied: chardet<4,>=3.0.2 in
/usr/local/lib/python3.8/dist-packages (from requests->torchtext==0.10.0)
(3.0.4)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in
/usr/local/lib/python3.8/dist-packages (from requests->torchtext==0.10.0)
(1.24.3)
Requirement already satisfied: certifi>=2017.4.17 in
/usr/local/lib/python3.8/dist-packages (from requests->torchtext==0.10.0)
(2022.9.24)
Building wheels for collected packages: fire
  Building wheel for fire (setup.py) … done
  Created wheel for fire: filename=fire-0.4.0-py2.py3-none-any.whl size=115943
```

```
sha256=fdfeaef00c0c93bfe05bd27f5836799e6a527e3f576921610a32688fcfed7dc2
  Stored in directory: /root/.cache/pip/wheels/1f/10/06/2a990ee4d73a8479fe292244
5e8a876d38cfbfed052284c6a1
Successfully built fire
Installing collected packages: torch, fire, torchmetrics, tensorboardX,
lightning-utilities, torchvision, torchtext, torchaudio, pytorch-lightning
  WARNING: The scripts convert-caffe2-to-onnx and convert-onnx-to-caffe2
are installed in '/root/.local/bin' which is not on PATH.
  Consider adding this directory to PATH or, if you prefer to suppress this
warning, use --no-warn-script-location.
Successfully installed fire-0.4.0 lightning-utilities-0.3.0 pytorch-
lightning-1.8.3.post1 tensorboardX-2.5.1 torch-1.9.0 torchaudio-0.9.0
torchmetrics-0.11.0 torchtext-0.10.0 torchvision-0.10.0
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-
wheels/public/simple/
Requirement already satisfied: tensorflow in /usr/local/lib/python3.8/dist-
packages (2.9.2)
Requirement already satisfied: flatbuffers<2,>=1.12 in
/usr/local/lib/python3.8/dist-packages (from tensorflow) (1.12)
Requirement already satisfied: libclang>=13.0.0 in
/usr/local/lib/python3.8/dist-packages (from tensorflow) (14.0.6)
Requirement already satisfied: protobuf<3.20,>=3.9.2 in
/usr/local/lib/python3.8/dist-packages (from tensorflow) (3.19.6)
Requirement already satisfied: six>=1.12.0 in /usr/local/lib/python3.8/dist-
packages (from tensorflow) (1.15.0)
Requirement already satisfied: gast<=0.4.0,>=0.2.1 in
/usr/local/lib/python3.8/dist-packages (from tensorflow) (0.4.0)
Requirement already satisfied: numpy>=1.20 in /usr/local/lib/python3.8/dist-
packages (from tensorflow) (1.21.6)
Requirement already satisfied: grpcio<2.0,>=1.24.3 in
/usr/local/lib/python3.8/dist-packages (from tensorflow) (1.50.0)
Requirement already satisfied: keras<2.10.0,>=2.9.0rc0 in
/usr/local/lib/python3.8/dist-packages (from tensorflow) (2.9.0)
Requirement already satisfied: google-pasta>=0.1.1 in
/usr/local/lib/python3.8/dist-packages (from tensorflow) (0.2.0)
Requirement already satisfied: packaging in /usr/local/lib/python3.8/dist-
packages (from tensorflow) (21.3)
Requirement already satisfied: astunparse>=1.6.0 in
/usr/local/lib/python3.8/dist-packages (from tensorflow) (1.6.3)
Requirement already satisfied: opt-einsum>=2.3.2 in
/usr/local/lib/python3.8/dist-packages (from tensorflow) (3.3.0)
Requirement already satisfied: tensorboard<2.10,>=2.9 in
/usr/local/lib/python3.8/dist-packages (from tensorflow) (2.9.1)
Requirement already satisfied: tensorflow-io-gcs-filesystem>=0.23.1 in
/usr/local/lib/python3.8/dist-packages (from tensorflow) (0.28.0)
Requirement already satisfied: termcolor>=1.1.0 in
```

```
/usr/local/lib/python3.8/dist-packages (from tensorflow) (2.1.1)
Requirement already satisfied: keras-preprocessing>=1.1.1 in
/usr/local/lib/python3.8/dist-packages (from tensorflow) (1.1.2)
Requirement already satisfied: h5py>=2.9.0 in /usr/local/lib/python3.8/dist-
packages (from tensorflow) (3.1.0)
Requirement already satisfied: tensorflow-estimator<2.10.0,>=2.9.0rc0 in
/usr/local/lib/python3.8/dist-packages (from tensorflow) (2.9.0)
Requirement already satisfied: setuptools in /usr/local/lib/python3.8/dist-
packages (from tensorflow) (57.4.0)
Requirement already satisfied: wrapt>=1.11.0 in /usr/local/lib/python3.8/dist-
packages (from tensorflow) (1.14.1)
Requirement already satisfied: typing-extensions>=3.6.6 in
/usr/local/lib/python3.8/dist-packages (from tensorflow) (4.1.1)
Requirement already satisfied: absl-py>=1.0.0 in /usr/local/lib/python3.8/dist-
packages (from tensorflow) (1.3.0)
Requirement already satisfied: wheel<1.0,>=0.23.0 in
/usr/local/lib/python3.8/dist-packages (from astunparse>=1.6.0->tensorflow)
(0.38.4)
Requirement already satisfied: google-auth<3,>=1.6.3 in
/usr/local/lib/python3.8/dist-packages (from tensorboard<2.10,>=2.9->tensorflow)
(2.14.1)
Requirement already satisfied: markdown>=2.6.8 in /usr/local/lib/python3.8/dist-
packages (from tensorboard<2.10,>=2.9->tensorflow) (3.4.1)
Requirement already satisfied: google-auth-oauthlib<0.5,>=0.4.1 in
/usr/local/lib/python3.8/dist-packages (from tensorboard<2.10,>=2.9->tensorflow)
(0.4.6)
Requirement already satisfied: werkzeug>=1.0.1 in /usr/local/lib/python3.8/dist-
packages (from tensorboard<2.10,>=2.9->tensorflow) (1.0.1)
Requirement already satisfied: requests<3,>=2.21.0 in
/usr/local/lib/python3.8/dist-packages (from tensorboard<2.10,>=2.9->tensorflow)
(2.23.0)
Requirement already satisfied: tensorboard-plugin-wit>=1.6.0 in
/usr/local/lib/python3.8/dist-packages (from tensorboard<2.10,>=2.9->tensorflow)
(1.8.1)
Requirement already satisfied: tensorboard-data-server<0.7.0,>=0.6.0 in
/usr/local/lib/python3.8/dist-packages (from tensorboard<2.10,>=2.9->tensorflow)
(0.6.1)
Requirement already satisfied: rsa<5,>=3.1.4 in /usr/local/lib/python3.8/dist-
packages (from google-auth<3,>=1.6.3->tensorboard<2.10,>=2.9->tensorflow) (4.9)
Requirement already satisfied: cachetools<6.0,>=2.0.0 in
/usr/local/lib/python3.8/dist-packages (from google-
auth<3,>=1.6.3->tensorboard<2.10,>=2.9->tensorflow) (5.2.0)
Requirement already satisfied: pyasn1-modules>=0.2.1 in
/usr/local/lib/python3.8/dist-packages (from google-
auth<3,>=1.6.3->tensorboard<2.10,>=2.9->tensorflow) (0.2.8)
Requirement already satisfied: requests-oauthlib>=0.7.0 in
/usr/local/lib/python3.8/dist-packages (from google-auth-
oauthlib<0.5,>=0.4.1->tensorboard<2.10,>=2.9->tensorflow) (1.3.1)
```

```
Requirement already satisfied: importlib-metadata>=4.4 in
/usr/local/lib/python3.8/dist-packages (from
markdown>=2.6.8->tensorboard<2.10,>=2.9->tensorflow) (4.13.0)
Requirement already satisfied: zipp>=0.5 in /usr/local/lib/python3.8/dist-
packages (from importlib-
metadata>=4.4->markdown>=2.6.8->tensorboard<2.10,>=2.9->tensorflow) (3.10.0)
Requirement already satisfied: pyasn1<0.5.0,>=0.4.6 in
/usr/local/lib/python3.8/dist-packages (from pyasn1-modules>=0.2.1->google-
auth<3,>=1.6.3->tensorboard<2.10,>=2.9->tensorflow) (0.4.8)
Requirement already satisfied: certifi>=2017.4.17 in
/usr/local/lib/python3.8/dist-packages (from
requests<3,>=2.21.0->tensorboard<2.10,>=2.9->tensorflow) (2022.9.24)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in
/usr/local/lib/python3.8/dist-packages (from
requests<3,>=2.21.0->tensorboard<2.10,>=2.9->tensorflow) (1.24.3)
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.8/dist-
packages (from requests<3,>=2.21.0->tensorboard<2.10,>=2.9->tensorflow) (2.10)
Requirement already satisfied: chardet<4,>=3.0.2 in
/usr/local/lib/python3.8/dist-packages (from
requests<3,>=2.21.0->tensorboard<2.10,>=2.9->tensorflow) (3.0.4)
Requirement already satisfied: oauthlib>=3.0.0 in /usr/local/lib/python3.8/dist-
packages (from requests-oauthlib>=0.7.0->google-auth-
oauthlib<0.5,>=0.4.1->tensorboard<2.10,>=2.9->tensorflow) (3.2.2)
Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in
/usr/local/lib/python3.8/dist-packages (from packaging->tensorflow) (3.0.9)
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-
wheels/public/simple/
Collecting sklearn
  Downloading sklearn-0.0.post1.tar.gz (3.6 kB)
Building wheels for collected packages: sklearn
  Building wheel for sklearn (setup.py) … done
  Created wheel for sklearn: filename=sklearn-0.0.post1-py3-none-any.whl
size=2344
sha256=380efc71b3e10f4bd82cbf0d9007e191f5b49d283d7aecbf0795ea714e19359f
  Stored in directory: /root/.cache/pip/wheels/14/25/f7/1cc0956978ae479e75140219
088deb7a36f60459df242b1a72
Successfully built sklearn
Installing collected packages: sklearn
Successfully installed sklearn-0.0.post1
Cloning into 'CLIP'…
remote: Enumerating objects: 236, done.
remote: Total 236 (delta 0), reused 0 (delta 0), pack-reused 236
Receiving objects: 100% (236/236), 8.92 MiB | 24.88 MiB/s, done.
Resolving deltas: 100% (122/122), done.
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-
wheels/public/simple/
Collecting taming-transformers
  Downloading taming_transformers-0.0.1-py3-none-any.whl (45 kB)
```

```
     |                 | 45 kB 2.1 MB/s
Requirement already satisfied: torch in /root/.local/lib/python3.8/site-
packages (from taming-transformers) (1.9.0)
Requirement already satisfied: torchvision in /root/.local/lib/python3.8/site-
packages (from taming-transformers) (0.10.0)
Requirement already satisfied: numpy in /usr/local/lib/python3.8/dist-packages
(from taming-transformers) (1.21.6)
Requirement already satisfied: pytorch-lightning>=1.0.8 in
/root/.local/lib/python3.8/site-packages (from taming-transformers)
(1.8.3.post1)
Requirement already satisfied: tqdm in /usr/local/lib/python3.8/dist-packages
(from taming-transformers) (4.64.1)
Collecting omegaconf>=2.0.0
  Downloading omegaconf-2.2.3-py3-none-any.whl (79 kB)
     |                 | 79 kB 4.4 MB/s
Collecting antlr4-python3-runtime==4.9.*
  Downloading antlr4-python3-runtime-4.9.3.tar.gz (117 kB)
     |                 | 117 kB 31.0 MB/s
Requirement already satisfied: PyYAML>=5.1.0 in
/usr/local/lib/python3.8/dist-packages (from omegaconf>=2.0.0->taming-
transformers) (6.0)
Requirement already satisfied: torchmetrics>=0.7.0 in
/root/.local/lib/python3.8/site-packages (from pytorch-lightning>=1.0.8->taming-
transformers) (0.11.0)
Requirement already satisfied: tensorboardX>=2.2 in
/root/.local/lib/python3.8/site-packages (from pytorch-lightning>=1.0.8->taming-
transformers) (2.5.1)
Requirement already satisfied: lightning-utilities==0.3.* in
/root/.local/lib/python3.8/site-packages (from pytorch-lightning>=1.0.8->taming-
transformers) (0.3.0)
Requirement already satisfied: packaging>=17.0 in /usr/local/lib/python3.8/dist-
packages (from pytorch-lightning>=1.0.8->taming-transformers) (21.3)
Requirement already satisfied: typing-extensions>=4.0.0 in
/usr/local/lib/python3.8/dist-packages (from pytorch-lightning>=1.0.8->taming-
transformers) (4.1.1)
Requirement already satisfied: fsspec[http]>2021.06.0 in
/usr/local/lib/python3.8/dist-packages (from pytorch-lightning>=1.0.8->taming-
transformers) (2022.11.0)
Requirement already satisfied: fire in /root/.local/lib/python3.8/site-packages
(from lightning-utilities==0.3.*->pytorch-lightning>=1.0.8->taming-transformers)
(0.4.0)
Requirement already satisfied: aiohttp!=4.0.0a0,!=4.0.0a1 in
/usr/local/lib/python3.8/dist-packages (from fsspec[http]>2021.06.0->pytorch-
lightning>=1.0.8->taming-transformers) (3.8.3)
Requirement already satisfied: requests in /usr/local/lib/python3.8/dist-
packages (from fsspec[http]>2021.06.0->pytorch-lightning>=1.0.8->taming-
transformers) (2.23.0)
Requirement already satisfied: attrs>=17.3.0 in /usr/local/lib/python3.8/dist-
```

```
packages (from aiohttp!=4.0.0a0,!=4.0.0a1->fsspec[http]>2021.06.0->pytorch-
lightning>=1.0.8->taming-transformers) (22.1.0)
Requirement already satisfied: multidict<7.0,>=4.5 in
/usr/local/lib/python3.8/dist-packages (from
aiohttp!=4.0.0a0,!=4.0.0a1->fsspec[http]>2021.06.0->pytorch-
lightning>=1.0.8->taming-transformers) (6.0.2)
Requirement already satisfied: frozenlist>=1.1.1 in
/usr/local/lib/python3.8/dist-packages (from
aiohttp!=4.0.0a0,!=4.0.0a1->fsspec[http]>2021.06.0->pytorch-
lightning>=1.0.8->taming-transformers) (1.3.3)
Requirement already satisfied: async-timeout<5.0,>=4.0.0a3 in
/usr/local/lib/python3.8/dist-packages (from
aiohttp!=4.0.0a0,!=4.0.0a1->fsspec[http]>2021.06.0->pytorch-
lightning>=1.0.8->taming-transformers) (4.0.2)
Requirement already satisfied: charset-normalizer<3.0,>=2.0 in
/usr/local/lib/python3.8/dist-packages (from
aiohttp!=4.0.0a0,!=4.0.0a1->fsspec[http]>2021.06.0->pytorch-
lightning>=1.0.8->taming-transformers) (2.1.1)
Requirement already satisfied: yarl<2.0,>=1.0 in /usr/local/lib/python3.8/dist-
packages (from aiohttp!=4.0.0a0,!=4.0.0a1->fsspec[http]>2021.06.0->pytorch-
lightning>=1.0.8->taming-transformers) (1.8.1)
Requirement already satisfied: aiosignal>=1.1.2 in
/usr/local/lib/python3.8/dist-packages (from
aiohttp!=4.0.0a0,!=4.0.0a1->fsspec[http]>2021.06.0->pytorch-
lightning>=1.0.8->taming-transformers) (1.3.1)
Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in
/usr/local/lib/python3.8/dist-packages (from packaging>=17.0->pytorch-
lightning>=1.0.8->taming-transformers) (3.0.9)
Requirement already satisfied: protobuf<=3.20.1,>=3.8.0 in
/usr/local/lib/python3.8/dist-packages (from tensorboardX>=2.2->pytorch-
lightning>=1.0.8->taming-transformers) (3.19.6)
Requirement already satisfied: idna>=2.0 in /usr/local/lib/python3.8/dist-
packages (from
yarl<2.0,>=1.0->aiohttp!=4.0.0a0,!=4.0.0a1->fsspec[http]>2021.06.0->pytorch-
lightning>=1.0.8->taming-transformers) (2.10)
Requirement already satisfied: six in /usr/local/lib/python3.8/dist-packages
(from fire->lightning-utilities==0.3.*->pytorch-lightning>=1.0.8->taming-
transformers) (1.15.0)
Requirement already satisfied: termcolor in /usr/local/lib/python3.8/dist-
packages (from fire->lightning-utilities==0.3.*->pytorch-
lightning>=1.0.8->taming-transformers) (2.1.1)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in
/usr/local/lib/python3.8/dist-packages (from
requests->fsspec[http]>2021.06.0->pytorch-lightning>=1.0.8->taming-transformers)
(1.24.3)
Requirement already satisfied: certifi>=2017.4.17 in
/usr/local/lib/python3.8/dist-packages (from
requests->fsspec[http]>2021.06.0->pytorch-lightning>=1.0.8->taming-transformers)
```

```
(2022.9.24)
Requirement already satisfied: chardet<4,>=3.0.2 in
/usr/local/lib/python3.8/dist-packages (from
requests->fsspec[http]>2021.06.0->pytorch-lightning>=1.0.8->taming-transformers)
(3.0.4)
Requirement already satisfied: pillow>=5.3.0 in /usr/local/lib/python3.8/dist-
packages (from torchvision->taming-transformers) (7.1.2)
Building wheels for collected packages: antlr4-python3-runtime
  Building wheel for antlr4-python3-runtime (setup.py) … done
  Created wheel for antlr4-python3-runtime:
filename=antlr4_python3_runtime-4.9.3-py3-none-any.whl size=144575
sha256=7c954a1bb2576168be34b1547b83c16e276b67fc5389adc8e8c8793ffdd58134
  Stored in directory: /root/.cache/pip/wheels/b1/a3/c2/6df046c09459b73cc9bb6c44
01b0be6c47048baf9a1617c485
Successfully built antlr4-python3-runtime
Installing collected packages: antlr4-python3-runtime, omegaconf, taming-
transformers
Successfully installed antlr4-python3-runtime-4.9.3 omegaconf-2.2.3 taming-
transformers-0.0.1

Cloning into 'taming-transformers'…
remote: Enumerating objects: 1339, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 1339 (delta 0), reused 2 (delta 0), pack-reused 1335
Receiving objects: 100% (1339/1339), 409.77 MiB | 45.60 MiB/s, done.
Resolving deltas: 100% (278/278), done.
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-
wheels/public/simple/
Collecting ftfy
  Downloading ftfy-6.1.1-py3-none-any.whl (53 kB)
     |                  | 53 kB 1.7 MB/s
Requirement already satisfied: regex in /usr/local/lib/python3.8/dist-
packages (2022.6.2)
Requirement already satisfied: tqdm in /usr/local/lib/python3.8/dist-packages
(4.64.1)
Requirement already satisfied: omegaconf in /usr/local/lib/python3.8/dist-
packages (2.2.3)
Requirement already satisfied: wcwidth>=0.2.5 in /usr/local/lib/python3.8/dist-
packages (from ftfy) (0.2.5)
Requirement already satisfied: PyYAML>=5.1.0 in /usr/local/lib/python3.8/dist-
packages (from omegaconf) (6.0)
Requirement already satisfied: antlr4-python3-runtime==4.9.* in
/usr/local/lib/python3.8/dist-packages (from omegaconf) (4.9.3)
Installing collected packages: ftfy
Successfully installed ftfy-6.1.1
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-
wheels/public/simple/
```

```
Collecting kornia
  Downloading kornia-0.6.8-py2.py3-none-any.whl (551 kB)
     |                 | 551 kB 4.3 MB/s
Requirement already satisfied: torch>=1.8.1 in
/root/.local/lib/python3.8/site-packages (from kornia) (1.9.0)
Requirement already satisfied: packaging in /usr/local/lib/python3.8/dist-
packages (from kornia) (21.3)
Requirement already satisfied: typing-extensions in
/usr/local/lib/python3.8/dist-packages (from torch>=1.8.1->kornia) (4.1.1)
Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in
/usr/local/lib/python3.8/dist-packages (from packaging->kornia) (3.0.9)
Installing collected packages: kornia
Successfully installed kornia-0.6.8
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-
wheels/public/simple/
Collecting imageio-ffmpeg
  Downloading imageio_ffmpeg-0.4.7-py3-none-manylinux2010_x86_64.whl (26.9 MB)
     |                 | 26.9 MB 1.4 MB/s
Installing collected packages: imageio-ffmpeg
Successfully installed imageio-ffmpeg-0.4.7
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-
wheels/public/simple/
Collecting einops
  Downloading einops-0.6.0-py3-none-any.whl (41 kB)
     |                 | 41 kB 57 kB/s
Installing collected packages: einops
Successfully installed einops-0.6.0
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-
wheels/public/simple/
Collecting pynvml
  Downloading pynvml-11.4.1-py3-none-any.whl (46 kB)
     |                 | 46 kB 2.6 MB/s
Installing collected packages: pynvml
Successfully installed pynvml-11.4.1
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-
wheels/public/simple/
Requirement already satisfied: librosa in /usr/local/lib/python3.8/dist-packages
(0.8.1)
Requirement already satisfied: soundfile in /usr/local/lib/python3.8/dist-
packages (0.11.0)
Requirement already satisfied: numpy in /usr/local/lib/python3.8/dist-packages
(1.21.6)
Collecting SpeechRecognition
  Downloading SpeechRecognition-3.9.0-py2.py3-none-any.whl (32.8 MB)
     |                 | 32.8 MB 239 kB/s
Collecting pydub
  Downloading pydub-0.25.1-py2.py3-none-any.whl (32 kB)
Requirement already satisfied: numba>=0.43.0 in /usr/local/lib/python3.8/dist-
```

```
packages (from librosa) (0.56.4)
Requirement already satisfied: scikit-learn!=0.19.0,>=0.14.0 in
/usr/local/lib/python3.8/dist-packages (from librosa) (1.0.2)
Requirement already satisfied: pooch>=1.0 in /usr/local/lib/python3.8/dist-
packages (from librosa) (1.6.0)
Requirement already satisfied: audioread>=2.0.0 in
/usr/local/lib/python3.8/dist-packages (from librosa) (3.0.0)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.8/dist-
packages (from librosa) (21.3)
Requirement already satisfied: resampy>=0.2.2 in /usr/local/lib/python3.8/dist-
packages (from librosa) (0.4.2)
Requirement already satisfied: decorator>=3.0.0 in
/usr/local/lib/python3.8/dist-packages (from librosa) (4.4.2)
Requirement already satisfied: scipy>=1.0.0 in /usr/local/lib/python3.8/dist-
packages (from librosa) (1.7.3)
Requirement already satisfied: joblib>=0.14 in /usr/local/lib/python3.8/dist-
packages (from librosa) (1.2.0)
Requirement already satisfied: cffi>=1.0 in /usr/local/lib/python3.8/dist-
packages (from soundfile) (1.15.1)
Requirement already satisfied: pycparser in /usr/local/lib/python3.8/dist-
packages (from cffi>=1.0->soundfile) (2.21)
Requirement already satisfied: setuptools in /usr/local/lib/python3.8/dist-
packages (from numba>=0.43.0->librosa) (57.4.0)
Requirement already satisfied: importlib-metadata in
/usr/local/lib/python3.8/dist-packages (from numba>=0.43.0->librosa) (4.13.0)
Requirement already satisfied: llvmlite<0.40,>=0.39.0dev0 in
/usr/local/lib/python3.8/dist-packages (from numba>=0.43.0->librosa) (0.39.1)
Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in
/usr/local/lib/python3.8/dist-packages (from packaging>=20.0->librosa) (3.0.9)
Requirement already satisfied: appdirs>=1.3.0 in /usr/local/lib/python3.8/dist-
packages (from pooch>=1.0->librosa) (1.4.4)
Requirement already satisfied: requests>=2.19.0 in
/usr/local/lib/python3.8/dist-packages (from pooch>=1.0->librosa) (2.23.0)
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.8/dist-
packages (from requests>=2.19.0->pooch>=1.0->librosa) (2.10)
Requirement already satisfied: certifi>=2017.4.17 in
/usr/local/lib/python3.8/dist-packages (from
requests>=2.19.0->pooch>=1.0->librosa) (2022.9.24)
Requirement already satisfied: chardet<4,>=3.0.2 in
/usr/local/lib/python3.8/dist-packages (from
requests>=2.19.0->pooch>=1.0->librosa) (3.0.4)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in
/usr/local/lib/python3.8/dist-packages (from
requests>=2.19.0->pooch>=1.0->librosa) (1.24.3)
Requirement already satisfied: threadpoolctl>=2.0.0 in
/usr/local/lib/python3.8/dist-packages (from scikit-
learn!=0.19.0,>=0.14.0->librosa) (3.1.0)
Collecting requests>=2.19.0
```

```
  Downloading requests-2.28.1-py3-none-any.whl (62 kB)
     |                  | 62 kB 1.5 MB/s
Requirement already satisfied: charset-normalizer<3,>=2 in
/usr/local/lib/python3.8/dist-packages (from
requests>=2.19.0->pooch>=1.0->librosa) (2.1.1)
Requirement already satisfied: zipp>=0.5 in /usr/local/lib/python3.8/dist-
packages (from importlib-metadata->numba>=0.43.0->librosa) (3.10.0)
Installing collected packages: requests, SpeechRecognition, pydub
  Attempting uninstall: requests
    Found existing installation: requests 2.23.0
    Uninstalling requests-2.23.0:
      Successfully uninstalled requests-2.23.0
Successfully installed SpeechRecognition-3.9.0 pydub-0.25.1 requests-2.28.1
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-
wheels/public/simple/
Collecting setuptools==59.5.0
  Downloading setuptools-59.5.0-py3-none-any.whl (952 kB)
     |                  | 952 kB 4.7 MB/s
Installing collected packages: setuptools
  Attempting uninstall: setuptools
    Found existing installation: setuptools 57.4.0
    Uninstalling setuptools-57.4.0:
      Successfully uninstalled setuptools-57.4.0
ERROR: pip's dependency resolver does not currently take into account all

the packages that are installed. This behaviour is the source of the following

dependency conflicts.

ipython 7.9.0 requires jedi>=0.10, which is not installed.
Successfully installed setuptools-59.5.0
```

This cell includes code for importing the libraries required for the VQGAN+CLIP model. The VQGAN model trained on imagenet is downloaded. The Nvidia GPU drivers are loaded. ALso, the google drive is linked for fetching dataset and base images.

[3]:
```python
# Importing misc libraries
import argparse
import sys
import warnings
from google.colab import drive, output
import numpy as np
import os, glob
import matplotlib.pyplot as plt
from tqdm.notebook import tqdm

# Importing python audio, speech recognition and image libraries
from omegaconf import OmegaConf
import kornia.augmentation as K
```

```
from PIL import ImageFile, Image
import librosa
import soundfile
from pydub import AudioSegment
from IPython.display import Audio
from IPython.core.display import display
import speech_recognition as sr
from IPython.display import Javascript
from base64 import b64decode
import matplotlib.image as mpimg


#Importing GPU libraries
from pynvml.smi import nvmlInit, nvmlDeviceGetHandleByIndex,␣
 ↪nvmlDeviceGetUtilizationRates

#Importing ML libraries
import torch
import tensorflow as tf
import keras
import pickle
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, LSTM, Dropout, Conv2D, MaxPooling2D,␣
 ↪AveragePooling2D, Activation, BatchNormalization, Flatten
from sklearn import preprocessing
from torch import nn, optim
from torch.nn import functional as F
from torchvision import transforms
from torchvision.transforms import functional as TF
from CLIP import clip
from taming.models import cond_transformer, vqgan
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn.neural_network import MLPClassifier
%load_ext tensorboard



# Misc configurations
sys.path.insert(1, './taming-transformers')
ImageFile.LOAD_TRUNCATED_IMAGES = True
warnings.filterwarnings("ignore")


#Initializing GPU
try:
  nvmlInit()
```

```
  handle = nvmlDeviceGetHandleByIndex(0)
  !nvidia-smi
except Exception as err:
  print('No GPU available - ', err)



#Downloading trained VQGAN model
torch.hub.download_url_to_file('https://heibox.uni-heidelberg.de/d/
 ↪a7530b09fed84f80a887/files/?p=%2Fconfigs%2Fmodel.yaml&dl=1',
                                'vqgan_imagenet_f16_16384.yaml')
torch.hub.download_url_to_file('https://heibox.uni-heidelberg.de/d/
 ↪a7530b09fed84f80a887/files/?p=%2Fckpts%2Flast.ckpt&dl=1',
                                'vqgan_imagenet_f16_16384.ckpt')


#Mounting Google drive
drive.mount('/content/drive', force_remount=False)
os.listdir("/content/drive/MyDrive/ECS271/dataset")
```

```
Tue Dec  6 11:10:12 2022
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 460.32.03    Driver Version: 460.32.03    CUDA Version: 11.2     |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|                               |                      |               MIG M. |
|===============================+======================+======================|
|   0  Tesla T4            Off  | 00000000:00:04.0 Off |                    0 |
| N/A   64C    P0    28W /  70W |      3MiB / 15109MiB |      0%      Default |
|                               |                      |                  N/A |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                                  |
|  GPU   GI   CI        PID   Type   Process name                  GPU Memory |
|        ID   ID                                                   Usage      |
|=============================================================================|
|  No running processes found                                                 |
+-----------------------------------------------------------------------------+

/root/.local/lib/python3.8/site-
packages/pytorch_lightning/utilities/distributed.py:258:
LightningDeprecationWarning:
`pytorch_lightning.utilities.distributed.rank_zero_only` has been deprecated in
v1.8.1 and will be removed in v1.10.0. You can import it from
`pytorch_lightning.utilities` instead.
  rank_zero_deprecation(
```

```
  0%|          | 0.00/692 [00:00<?, ?B/s]
  0%|          | 0.00/935M [00:00<?, ?B/s]
Drive already mounted at /content/drive; to attempt to forcibly remount, call
drive.mount("/content/drive", force_remount=True).
```

[3]:
```
['Actor_08',
 'Actor_24',
 'Actor_09',
 'Actor_22',
 'Actor_07',
 'Actor_13',
 'Actor_23',
 'Actor_14',
 'Actor_06',
 'Actor_01',
 'Actor_03',
 'Actor_19',
 'Actor_05',
 'Actor_15',
 'Actor_17',
 'Actor_02',
 'Actor_12',
 'Actor_04',
 'Actor_10',
 'Actor_21',
 'Actor_20',
 'Actor_11',
 'Actor_18',
 'Actor_16']
```

We define methods required for training VQGAN+CLIP. Specifically, the ReplaceGrad and Clamp-Grad functions are defined.

[4]:
```
"""
Reference: https://colab.research.google.com/drive/
  ↪1go6YwMFe5MX6XM9tv-cnQiSTU50N9EeT#scrollTo=g7EDme5RYCrt
"""
"""
Overriding torch's autograd function with ReplaceGrad which masks
a gradient during backward propagation with a supplied gradient reshaped to
the original gradients shape
"""
class ReplaceGrad(torch.autograd.Function):
    @staticmethod
    def forward(ctx, x_forward, x_backward):
        ctx.shape = x_backward.shape
        return x_forward
```

```
    @staticmethod
    def backward(ctx, grad_in):
        return None, grad_in.sum_to_size(ctx.shape)
replace_grad = ReplaceGrad.apply

"""
Overriding torch's autograd function with ClampGrad which restricts a gradient
within a limit. The None terms in backward propagation represents the gradient
for minimum and maximum values which is constant
"""
class ClampWithGrad(torch.autograd.Function):
    @staticmethod
    def forward(ctx, input, min, max):
        ctx.min = min
        ctx.max = max
        ctx.save_for_backward(input)
        return input.clamp(min, max)
    @staticmethod
    def backward(ctx, grad_in):
        input, = ctx.saved_tensors
        return grad_in * (grad_in * (input - input.clamp(ctx.min, ctx.max)) >=␣
 ↪0), None, None
clamp_with_grad = ClampWithGrad.apply
```

The Prompt class is defined to encode the text inferred from the audio and calculate the similarity between the generated image and the text.

[5]:
```
"""
Reference: https://colab.research.google.com/drive/
 ↪1go6YwMFe5MX6XM9tv-cnQiSTU50N9EeT#scrollTo=g7EDme5RYCrt
"""
"""
Encodes the text prompt supplied by the user and calculate the  cosine␣
 ↪similarity
between the generated image and the text.
"""
class Prompt(nn.Module):
    def __init__(self, embed):
        super().__init__()
        self.register_buffer('embed', embed)
    def forward(self, input):
        input_normed = F.normalize(input.unsqueeze(1), dim=2)
        embed_normed = F.normalize(self.embed.unsqueeze(0), dim=2)
        dists = input_normed.sub(embed_normed).norm(dim=2).div(2).arcsin().
 ↪pow(2).mul(2)
        return dists.mean()
```

Code for making cutouts in the image to be used in the codebook.

[6]:
```
"""
Reference: https://colab.research.google.com/drive/
 ↪1go6YwMFe5MX6XM9tv-cnQiSTU50N9EeT#scrollTo=g7EDme5RYCrt
"""
"""
Model to make cutouts in the generated image to be used in the codebook by VQGAN
"""
class MakeCutouts(nn.Module):
    def __init__(self, cut_size, dim):
        super().__init__()
        self.dim = dim
        self.augs = nn.Sequential(
            K.RandomAffine(degrees=15, translate=0.1, p=0.7,␣
 ↪padding_mode='border'),
            K.RandomPerspective(0.7,p=0.7),
            K.ColorJitter(hue=0.1, saturation=0.1, p=0.7),
            K.RandomErasing((.1, .4), (.3, 1/.3), same_on_batch=True, p=0.7),
        )
        self.noise = 0.1
        self.av_pool = nn.AdaptiveAvgPool2d((cut_size, cut_size))
        self.max_pool = nn.AdaptiveMaxPool2d((cut_size, cut_size))

    def forward(self, input):
        batch = self.augs(torch.cat([(self.av_pool(input) + self.
 ↪max_pool(input))/2 for _ in range(self.dim)], dim=0))
        batch = batch + batch.new_empty([self.dim, 1, 1, 1]).uniform_(0, self.
 ↪noise) * torch.randn_like(batch)
        return batch
```

Loading VQGAN model and initliizing them.

[7]:
```
"""
Reference: https://colab.research.google.com/drive/
 ↪1go6YwMFe5MX6XM9tv-cnQiSTU50N9EeT#scrollTo=g7EDme5RYCrt
"""
"""
Function to load VQGAN model
"""
def load_vqgan_model(config_path, checkpoint_path):
    config = OmegaConf.load(config_path)
    model = vqgan.VQModel(**config.model.params)
    model.eval().requires_grad_(False)
    model.init_from_ckpt(checkpoint_path)
    del model.loss
    return model

model_name = "vqgan_imagenet_f16_16384" # Using VQGAN trained on ImageNet
```

```
# Parameters
images_interval =  50
args = argparse.Namespace(
    clip_model='ViT-B/32',
    vqgan_config=f'{model_name}.yaml',
    vqgan_checkpoint=f'{model_name}.ckpt',
    lr=0.01,
    cut_dim=64,
    display_freq=images_interval
)
# Using GPU or CPU if GPU not available
device = torch.device('cuda:0' if torch.cuda.is_available() else 'cpu')
print('Using device:', device)

# Loding VQGAN as generator and CLIP as perceptor
model = load_vqgan_model(args.vqgan_config, args.vqgan_checkpoint).to(device)
perceptor = clip.load(args.clip_model, jit=False)[0].eval().
 ↪requires_grad_(False).to(device)
```

```
Using device: cuda:0
Working with z of shape (1, 256, 16, 16) = 65536 dimensions.
loaded pretrained LPIPS loss from taming/modules/autoencoder/lpips/vgg.pth
VQLPIPSWithDiscriminator running with hinge loss.
Restored from vqgan_imagenet_f16_16384.ckpt
```

Defining the training and inference functions to be used for VQGAN+CLIP for generating images.

[8]:
```
"""
Reference: https://colab.research.google.com/drive/
 ↪1go6YwMFe5MX6XM9tv-cnQiSTU50N9EeT#scrollTo=g7EDme5RYCrt
"""
# Generateing image using VQGAN+CLIP with a text input and base image
def get_image(text,
              max_iterations,
              width,
              height,
              init_image
             ):
    size = [width, height]
    texts = text

    max_iterations = max_iterations

    texts = [phrase.strip() for phrase in texts.split("|")]
    if texts == ['']:
        texts = []
    if texts:
```

```
        print('Using texts:', texts)
    seed = torch.seed()
    torch.manual_seed(seed)
    print('Using seed:', seed)
    print('Using init image', init_image)

    # Make cutouts
    cut_size = perceptor.visual.input_resolution
    make_cutouts = MakeCutouts(cut_size, args.cut_dim)
    z_min = model.quantize.embedding.weight.min(dim=0).values[None, :, None,␣
↪None]
    z_max = model.quantize.embedding.weight.max(dim=0).values[None, :, None,␣
↪None]

    # Open base image and encoding for training
    img = Image.open(init_image)
    pil_image = img.convert('RGB')
    pil_image = pil_image.resize((size[0], size[1]), Image.LANCZOS)
    pil_tensor = TF.to_tensor(pil_image)

    z, *_ = model.encode(pil_tensor.to(device).unsqueeze(0) * 2 - 1)
    z.requires_grad_(True)
    opt = optim.Adam([z], lr=args.lr)
    normalize = transforms.Normalize(mean=[0.48145466, 0.4578275, 0.40821073],
                                     std=[0.26862954, 0.26130258, 0.27577711])
    #Encoding input texts
    pMs = [Prompt(perceptor.encode_text(clip.tokenize(prompt).to(device)).
↪float()).to(device) for prompt in texts]

    # Processing vector quantized input image
    def process(z):
        d = z.movedim(1, 3).pow(2).sum(dim=-1, keepdim=True) + model.quantize.
↪embedding.weight.pow(2).sum(dim=1) - 2 * z.movedim(1, 3) @ model.quantize.
↪embedding.weight.T
        indices = d.argmin(-1)
        x_q = F.one_hot(indices, model.quantize.embedding.weight.shape[0]).to(d.
↪dtype) @ model.quantize.embedding.weight
        z_q = replace_grad(x_q, z.movedim(1, 3)).movedim(3, 1)
        return clamp_with_grad(model.decode(z_q).add(1).div(2), 0, 1)

    #Traing VQGAN+CLIP
    def train(i):
        opt.zero_grad()
        out = process(z)
        ims = perceptor.encode_image(normalize(make_cutouts(out))).float()
        lossAll = [prompt(ims) for prompt in pMs]
```

```
        image = np.transpose(np.array(out.mul(255).clamp(0, 255)[0].cpu().
↪detach().numpy().astype(np.uint8))[:,:,:], (1, 2, 0))
        if i % args.display_freq == 0:
            tqdm.write(f'i: {i}, loss: {sum(lossAll).item():g}')
            out = process(z)

        loss = sum(lossAll)
        loss.backward()
        opt.step()
        with torch.no_grad():
            z.copy_(z.maximum(z_min).minimum(z_max))
        return image

    # Training iterations
    i = 0
    try:
        with tqdm() as pbar:
            while True:
                image = train(i)
                if i == max_iterations:
                    break
                i += 1
                pbar.update()
    except KeyboardInterrupt:
        pass
    return image
```

Code for loading and extracting features from the dataset.

[9]:
```
# Emotions in dataset
emotions={'01':'neutral', '02':'calm', '03':'happy', '04':'sad', '05':'angry',␣
↪'06':'fearful', '07':'disgust', '08':'surprised'}

# Extract mfcc, stft, mel spectrogram features from input audio
def extract_feature(file_name):
    with soundfile.SoundFile(file_name) as sound_file:
          X = sound_file.read(dtype="float32")
          sample_rate = sound_file.samplerate
          mfcc = librosa.feature.mfcc(y=X, sr=sample_rate, n_mfcc=40).T
          chroma = librosa.feature.chroma_stft(S=np.abs(librosa.stft(X)),␣
↪sr=sample_rate).T
          mel = librosa.feature.melspectrogram(X, sr=sample_rate).T
          feat = np.concatenate((mfcc, chroma, mel), axis=1)
          time_steps = 5
          q = int(feat.shape[0]/time_steps)
          res = feat[0:q*(time_steps-1),:].transpose().reshape(-1,q).mean(1).
↪reshape(feat.shape[1],-1).transpose()
```

```
            res = np.append(res, np.mean(feat[q*(time_steps-1):feat.shape[0], :],␣
 ↪axis=0).reshape(1, feat.shape[1]), axis=0)
            return res

# Loading data
def load_data(test_size=0.2):
    x,y=[],[]
    emotion_freq = np.zeros(9)
    for file in glob.glob("/content/drive/MyDrive/ECS271/dataset/Actor_*/*.
 ↪wav"):
        file_name = os.path.basename(file)
        emotion = emotions[file_name.split("-")[2]]
        emotion_freq[int(file_name.split("-")[2])] += 1
        # Required only first time to convert number of channels to 1
        # sound = AudioSegment.from_wav(file).set_channels(1)
        # sound.export(file, format="wav")
        x.append(extract_feature(file))
        y.append(emotion)
    print(emotion_freq)
    return train_test_split(np.array(x), y, test_size=test_size, random_state=9)
```

Loading dataset

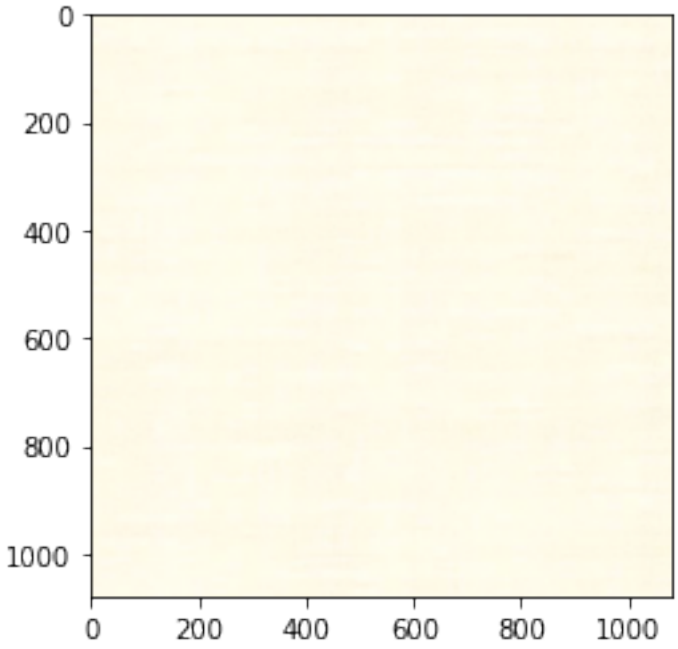
[13]:
```
# Loading training and testing dataset
x_train,x_test,y_train,y_test=load_data(test_size=0.25)
print(x_train.shape, x_test.shape, len(y_train), len(y_test))
```

```
[  0.  96. 192. 192. 192. 192. 192. 192. 192.]
(1080, 5, 180) (360, 5, 180) 1080 360
```

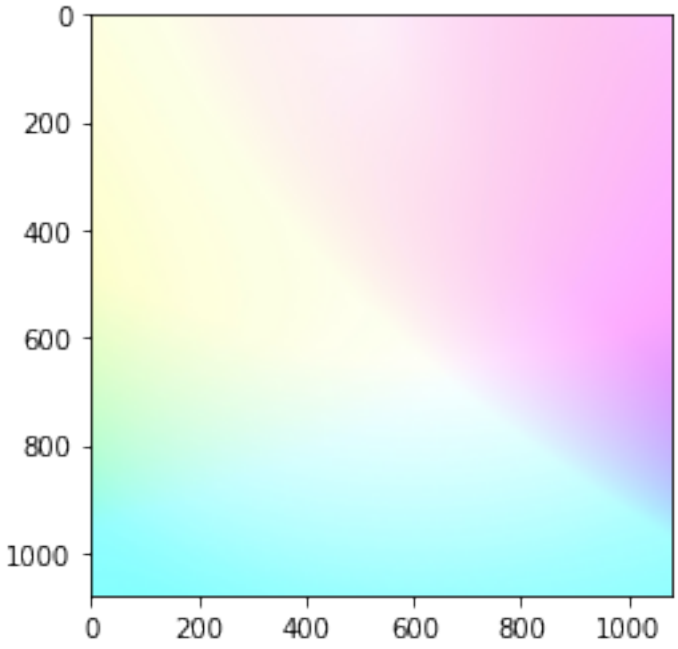
Showing the base images

[41]:
```
# Images for all emotions
for _, emotion in emotions.items():
  file = "/content/drive/MyDrive/ECS271/base_images/" + emotion + ".png"
  img = mpimg.imread(file)
  imgplot = plt.imshow(img)
  print(emotion)
  plt.show()
```

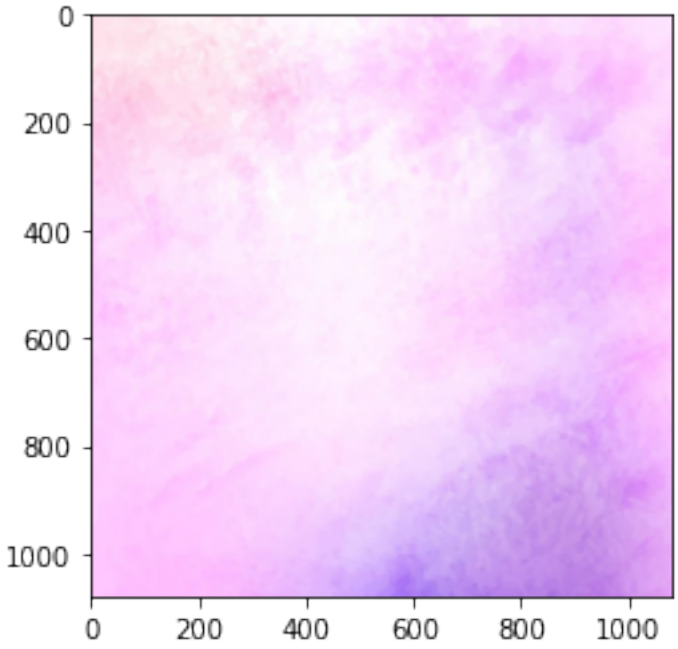
```
neutral
```


0
200
400
600
800
1000
0
200
400
600
800
1000

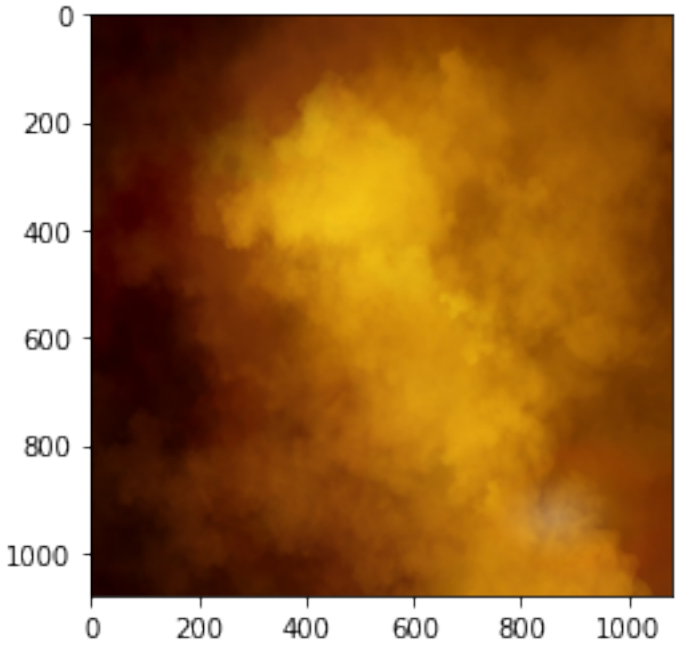
calm


0
200
400
600
800
1000
0
200
400
600
800
1000

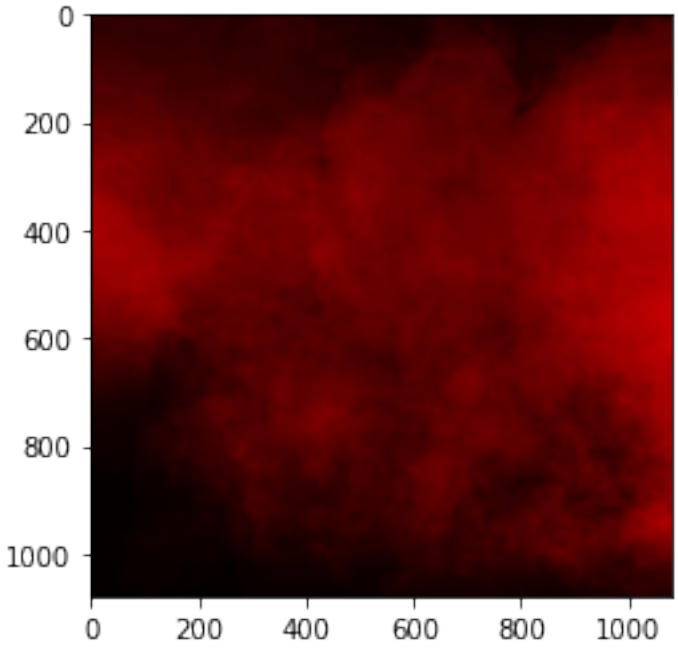
happy


0
200
400
600
800
1000
0
200
400
600
800
1000

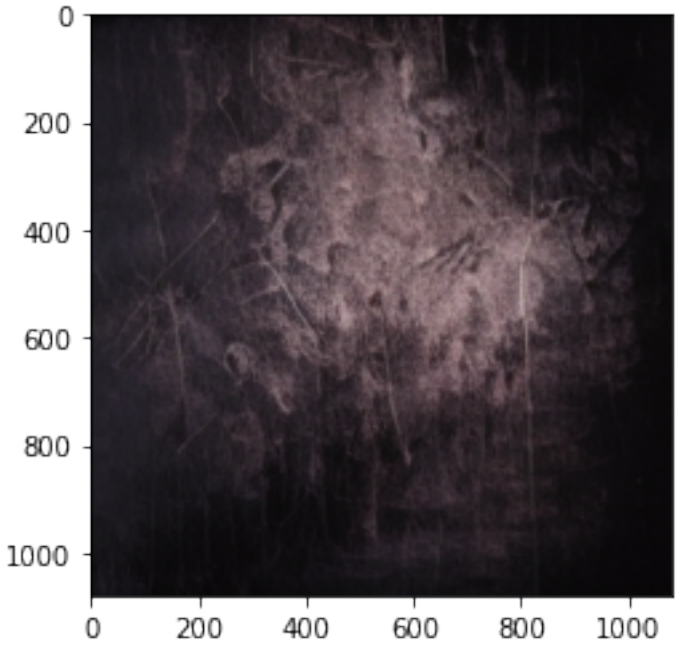
sad


0
200
400
600
800
1000
0
200
400
600
800
1000

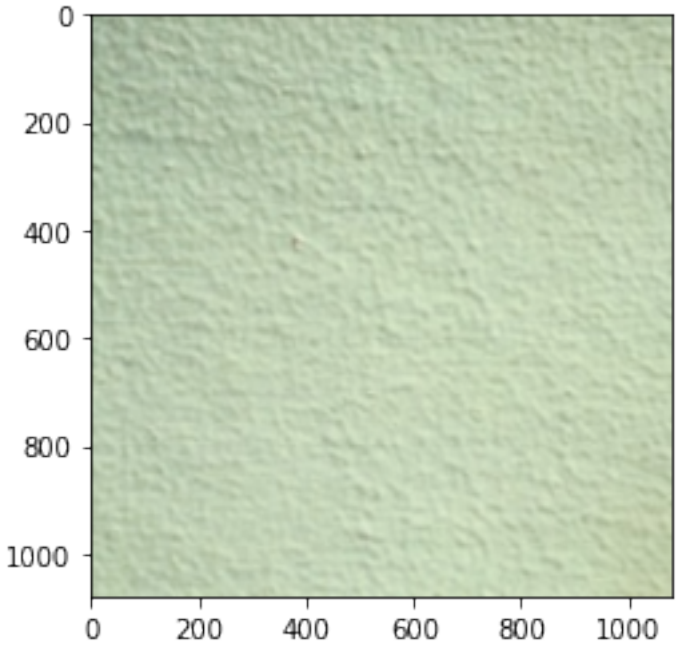
angry



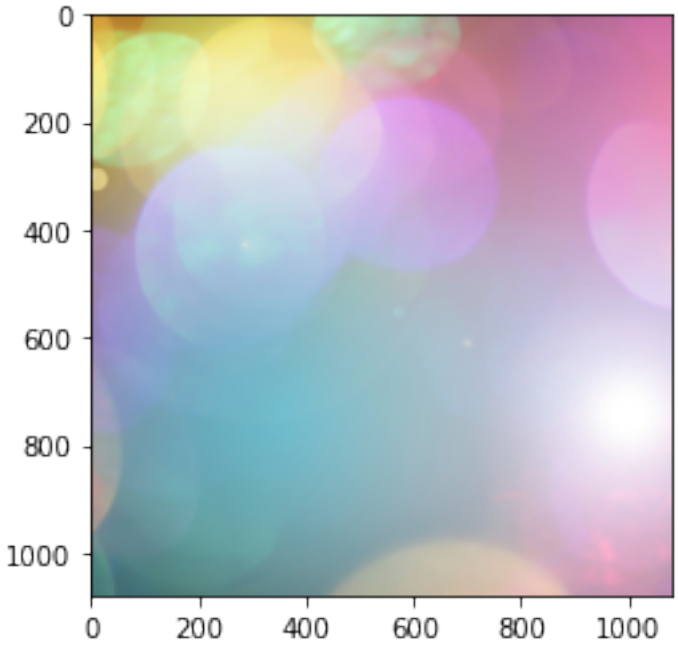
fearful



disgust




```
surprised
```



Fetching emotions and text from audio used during testing.

[18]:
```
# Getting emotion from audio file and trained model
def get_emotion(file_path):
    le = preprocessing.LabelEncoder()
    le.fit(np.array(y_train))
    feature = extract_feature(file_path)
    return le.inverse_transform(tf.keras.backend.argmax(loaded_model.predict(np.
 ↪reshape(feature, (1, feature.shape[0], feature.shape[1]))))[0]

# Get text from audio file using python libraries
def get_text(file_path):
    r = sr.Recognizer()
    with sr.AudioFile(file_path) as source:
        return r.recognize_google(r.record(source))
```

Defining LSTM model

[ ]:
```
# LSTM model
speech_model_lstm = tf.keras.Sequential()

speech_model_lstm.add(LSTM(256, input_shape=(5, 180), return_sequences=True))
speech_model_lstm.add(LSTM(128))
speech_model_lstm.add(Dense(2048, activation='relu'))
speech_model_lstm.add(Dense(2048, activation='relu'))
speech_model_lstm.add(Dropout(0.3))
speech_model_lstm.add(Dense(8, activation='softmax'))

speech_model_lstm.summary()
optimiser = tf.keras.optimizers.Adam(learning_rate=0.001)
speech_model_lstm.compile(optimizer=optimiser,
                  loss='sparse_categorical_crossentropy',
                  metrics=['accuracy'])
```

```
Model: "sequential_18"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm_16 (LSTM)              (None, 5, 256)            447488

 lstm_17 (LSTM)              (None, 128)               197120

 dense_48 (Dense)            (None, 2048)              264192

 dense_49 (Dense)            (None, 2048)              4196352

 dropout_16 (Dropout)        (None, 2048)              0

 dense_50 (Dense)            (None, 8)                 16392
```

```
=================================================================
Total params: 5,121,544
Trainable params: 5,121,544
Non-trainable params: 0
_________________________________________________________________
```

Training LSTM model. Batch size = 32, learning rate = 0.001, epochs = 100

[ ]:
```
# Training LSTM Model
le = preprocessing.LabelEncoder()
le.fit(np.array(y_train))
logdir = 'logs/lstm'
speech_model_lstm.fit(x_train,le.transform(np.array(y_train)),batch_size=32,␣
 ↪epochs=100, callbacks=[keras.callbacks.TensorBoard(log_dir=logdir)])
y_pred=le.inverse_transform(tf.keras.backend.argmax(speech_model_lstm.
 ↪predict(x_test)))

print('Accuracy with lstm - ', accuracy_score(y_true=y_test, y_pred=y_pred))

pickle.dump(speech_model_lstm, open('/content/drive/MyDrive/ECS271/
 ↪speech_emotion_lstm.sav', 'wb'))
%tensorboard --logdir logs
```

```
Epoch 1/100
34/34 [==============================] - 7s 78ms/step - loss: 1.9254 - accuracy:
0.2213
Epoch 2/100
34/34 [==============================] - 3s 77ms/step - loss: 1.6932 - accuracy:
0.3343
Epoch 3/100
34/34 [==============================] - 3s 78ms/step - loss: 1.6295 - accuracy:
0.3306
Epoch 4/100
34/34 [==============================] - 3s 78ms/step - loss: 1.5104 - accuracy:
0.4019
Epoch 5/100
34/34 [==============================] - 3s 77ms/step - loss: 1.3582 - accuracy:
0.4657
Epoch 6/100
34/34 [==============================] - 3s 77ms/step - loss: 1.3170 - accuracy:
0.4981
Epoch 7/100
34/34 [==============================] - 3s 77ms/step - loss: 1.3279 - accuracy:
0.4972
Epoch 8/100
34/34 [==============================] - 3s 76ms/step - loss: 1.1722 - accuracy:
0.5435
Epoch 9/100
```

```
34/34 [==============================] - 3s 76ms/step - loss: 1.1579 - accuracy:
0.5676
Epoch 10/100
34/34 [==============================] - 3s 78ms/step - loss: 1.1057 - accuracy:
0.5741
Epoch 11/100
34/34 [==============================] - 3s 79ms/step - loss: 1.0502 - accuracy:
0.6083
Epoch 12/100
34/34 [==============================] - 3s 78ms/step - loss: 1.0533 - accuracy:
0.6167
Epoch 13/100
34/34 [==============================] - 3s 78ms/step - loss: 0.9300 - accuracy:
0.6546
Epoch 14/100
34/34 [==============================] - 3s 77ms/step - loss: 0.8807 - accuracy:
0.6676
Epoch 15/100
34/34 [==============================] - 3s 77ms/step - loss: 0.7830 - accuracy:
0.6935
Epoch 16/100
34/34 [==============================] - 3s 78ms/step - loss: 0.6754 - accuracy:
0.7370
Epoch 17/100
34/34 [==============================] - 3s 76ms/step - loss: 0.7947 - accuracy:
0.7176
Epoch 18/100
34/34 [==============================] - 3s 76ms/step - loss: 0.6494 - accuracy:
0.7648
Epoch 19/100
34/34 [==============================] - 3s 78ms/step - loss: 0.6597 - accuracy:
0.7574
Epoch 20/100
34/34 [==============================] - 3s 77ms/step - loss: 0.6422 - accuracy:
0.7537
Epoch 21/100
34/34 [==============================] - 3s 77ms/step - loss: 0.5072 - accuracy:
0.8037
Epoch 22/100
34/34 [==============================] - 3s 77ms/step - loss: 0.5857 - accuracy:
0.7769
Epoch 23/100
34/34 [==============================] - 3s 76ms/step - loss: 0.5806 - accuracy:
0.7713
Epoch 24/100
34/34 [==============================] - 3s 76ms/step - loss: 0.5651 - accuracy:
0.7898
Epoch 25/100
```

```
34/34 [==============================] - 3s 77ms/step - loss: 0.5446 - accuracy:
0.7972
Epoch 26/100
34/34 [==============================] - 3s 76ms/step - loss: 0.6127 - accuracy:
0.7815
Epoch 27/100
34/34 [==============================] - 3s 78ms/step - loss: 0.5189 - accuracy:
0.8194
Epoch 28/100
34/34 [==============================] - 3s 77ms/step - loss: 0.4068 - accuracy:
0.8565
Epoch 29/100
34/34 [==============================] - 3s 76ms/step - loss: 0.3752 - accuracy:
0.8713
Epoch 30/100
34/34 [==============================] - 3s 78ms/step - loss: 0.2501 - accuracy:
0.9176
Epoch 31/100
34/34 [==============================] - 3s 77ms/step - loss: 0.3196 - accuracy:
0.8769
Epoch 32/100
34/34 [==============================] - 3s 77ms/step - loss: 0.3060 - accuracy:
0.9028
Epoch 33/100
34/34 [==============================] - 3s 76ms/step - loss: 0.3938 - accuracy:
0.8648
Epoch 34/100
34/34 [==============================] - 3s 77ms/step - loss: 0.3259 - accuracy:
0.8898
Epoch 35/100
34/34 [==============================] - 3s 77ms/step - loss: 0.2356 - accuracy:
0.9083
Epoch 36/100
34/34 [==============================] - 3s 77ms/step - loss: 0.3082 - accuracy:
0.8954
Epoch 37/100
34/34 [==============================] - 3s 77ms/step - loss: 0.3580 - accuracy:
0.8778
Epoch 38/100
34/34 [==============================] - 3s 77ms/step - loss: 0.3840 - accuracy:
0.8583
Epoch 39/100
34/34 [==============================] - 3s 78ms/step - loss: 0.2005 - accuracy:
0.9250
Epoch 40/100
34/34 [==============================] - 3s 79ms/step - loss: 0.1470 - accuracy:
0.9528
Epoch 41/100
```

```
34/34 [==============================] - 3s 77ms/step - loss: 0.1089 - accuracy:
0.9667
Epoch 42/100
34/34 [==============================] - 3s 76ms/step - loss: 0.1444 - accuracy:
0.9454
Epoch 43/100
34/34 [==============================] - 3s 77ms/step - loss: 0.2202 - accuracy:
0.9241
Epoch 44/100
34/34 [==============================] - 3s 76ms/step - loss: 0.2446 - accuracy:
0.9139
Epoch 45/100
34/34 [==============================] - 3s 77ms/step - loss: 0.2096 - accuracy:
0.9315
Epoch 46/100
34/34 [==============================] - 3s 76ms/step - loss: 0.1019 - accuracy:
0.9648
Epoch 47/100
34/34 [==============================] - 3s 76ms/step - loss: 0.1234 - accuracy:
0.9500
Epoch 48/100
34/34 [==============================] - 3s 78ms/step - loss: 0.2265 - accuracy:
0.9204
Epoch 49/100
34/34 [==============================] - 3s 77ms/step - loss: 0.2034 - accuracy:
0.9278
Epoch 50/100
34/34 [==============================] - 3s 78ms/step - loss: 0.1028 - accuracy:
0.9694
Epoch 51/100
34/34 [==============================] - 3s 79ms/step - loss: 0.0565 - accuracy:
0.9843
Epoch 52/100
34/34 [==============================] - 5s 143ms/step - loss: 0.1430 -
accuracy: 0.9546
Epoch 53/100
34/34 [==============================] - 3s 101ms/step - loss: 0.3351 -
accuracy: 0.8907
Epoch 54/100
34/34 [==============================] - 3s 77ms/step - loss: 0.2670 - accuracy:
0.9074
Epoch 55/100
34/34 [==============================] - 3s 77ms/step - loss: 0.1953 - accuracy:
0.9352
Epoch 56/100
34/34 [==============================] - 3s 77ms/step - loss: 0.1407 - accuracy:
0.9565
Epoch 57/100
```

```
34/34 [==============================] - 3s 78ms/step - loss: 0.1950 - accuracy:
0.9361
Epoch 58/100
34/34 [==============================] - 3s 77ms/step - loss: 0.1482 - accuracy:
0.9463
Epoch 59/100
34/34 [==============================] - 3s 77ms/step - loss: 0.0721 - accuracy:
0.9722
Epoch 60/100
34/34 [==============================] - 3s 78ms/step - loss: 0.0536 - accuracy:
0.9833
Epoch 61/100
34/34 [==============================] - 3s 78ms/step - loss: 0.0741 - accuracy:
0.9741
Epoch 62/100
34/34 [==============================] - 3s 78ms/step - loss: 0.1059 - accuracy:
0.9694
Epoch 63/100
34/34 [==============================] - 3s 77ms/step - loss: 0.0965 - accuracy:
0.9667
Epoch 64/100
34/34 [==============================] - 3s 76ms/step - loss: 0.2377 - accuracy:
0.9148
Epoch 65/100
34/34 [==============================] - 3s 76ms/step - loss: 0.1621 - accuracy:
0.9380
Epoch 66/100
34/34 [==============================] - 3s 76ms/step - loss: 0.1043 - accuracy:
0.9667
Epoch 67/100
34/34 [==============================] - 3s 78ms/step - loss: 0.1388 - accuracy:
0.9537
Epoch 68/100
34/34 [==============================] - 3s 76ms/step - loss: 0.1520 - accuracy:
0.9481
Epoch 69/100
34/34 [==============================] - 3s 78ms/step - loss: 0.1827 - accuracy:
0.9306
Epoch 70/100
34/34 [==============================] - 3s 77ms/step - loss: 0.0319 - accuracy:
0.9907
Epoch 71/100
34/34 [==============================] - 3s 78ms/step - loss: 0.0142 - accuracy:
0.9944
Epoch 72/100
34/34 [==============================] - 3s 77ms/step - loss: 0.0049 - accuracy:
0.9991
Epoch 73/100
```

```
34/34 [==============================] - 3s 79ms/step - loss: 0.0017 - accuracy:
1.0000
Epoch 74/100
34/34 [==============================] - 3s 77ms/step - loss: 8.8322e-04 -
accuracy: 1.0000
Epoch 75/100
34/34 [==============================] - 3s 76ms/step - loss: 6.9117e-04 -
accuracy: 1.0000
Epoch 76/100
34/34 [==============================] - 3s 77ms/step - loss: 6.7822e-04 -
accuracy: 1.0000
Epoch 77/100
34/34 [==============================] - 3s 78ms/step - loss: 5.0828e-04 -
accuracy: 1.0000
Epoch 78/100
34/34 [==============================] - 3s 76ms/step - loss: 4.5106e-04 -
accuracy: 1.0000
Epoch 79/100
34/34 [==============================] - 3s 77ms/step - loss: 4.0562e-04 -
accuracy: 1.0000
Epoch 80/100
34/34 [==============================] - 3s 78ms/step - loss: 4.2939e-04 -
accuracy: 1.0000
Epoch 81/100
34/34 [==============================] - 3s 78ms/step - loss: 3.0735e-04 -
accuracy: 1.0000
Epoch 82/100
34/34 [==============================] - 3s 77ms/step - loss: 2.6614e-04 -
accuracy: 1.0000
Epoch 83/100
34/34 [==============================] - 3s 77ms/step - loss: 3.0952e-04 -
accuracy: 1.0000
Epoch 84/100
34/34 [==============================] - 3s 78ms/step - loss: 2.5636e-04 -
accuracy: 1.0000
Epoch 85/100
34/34 [==============================] - 3s 78ms/step - loss: 2.3115e-04 -
accuracy: 1.0000
Epoch 86/100
34/34 [==============================] - 3s 78ms/step - loss: 2.7700e-04 -
accuracy: 1.0000
Epoch 87/100
34/34 [==============================] - 3s 78ms/step - loss: 2.1703e-04 -
accuracy: 1.0000
Epoch 88/100
34/34 [==============================] - 4s 124ms/step - loss: 2.0848e-04 -
accuracy: 1.0000
Epoch 89/100
```

```
34/34 [==============================] - 5s 138ms/step - loss: 1.9148e-04 -
accuracy: 1.0000
Epoch 90/100
34/34 [==============================] - 4s 127ms/step - loss: 1.8865e-04 -
accuracy: 1.0000
Epoch 91/100
34/34 [==============================] - 3s 97ms/step - loss: 1.5017e-04 -
accuracy: 1.0000
Epoch 92/100
34/34 [==============================] - 3s 77ms/step - loss: 1.5108e-04 -
accuracy: 1.0000
Epoch 93/100
34/34 [==============================] - 3s 77ms/step - loss: 1.4640e-04 -
accuracy: 1.0000
Epoch 94/100
34/34 [==============================] - 3s 77ms/step - loss: 1.4808e-04 -
accuracy: 1.0000
Epoch 95/100
34/34 [==============================] - 3s 78ms/step - loss: 1.1542e-04 -
accuracy: 1.0000
Epoch 96/100
34/34 [==============================] - 3s 77ms/step - loss: 1.4663e-04 -
accuracy: 1.0000
Epoch 97/100
34/34 [==============================] - 3s 77ms/step - loss: 1.0234e-04 -
accuracy: 1.0000
Epoch 98/100
34/34 [==============================] - 3s 77ms/step - loss: 9.5802e-05 -
accuracy: 1.0000
Epoch 99/100
34/34 [==============================] - 3s 78ms/step - loss: 1.1174e-04 -
accuracy: 1.0000
Epoch 100/100
34/34 [==============================] - 3s 77ms/step - loss: 1.0138e-04 -
accuracy: 1.0000
12/12 [==============================] - 1s 19ms/step
Accuracy with lstm -  0.7527777777777778

WARNING:absl:Found untraced functions such as lstm_cell_16_layer_call_fn,
lstm_cell_16_layer_call_and_return_conditional_losses,
lstm_cell_17_layer_call_fn,
lstm_cell_17_layer_call_and_return_conditional_losses while saving (showing 4 of
4). These functions will not be directly callable after loading.

Reusing TensorBoard on port 6006 (pid 2792), started 0:05:18 ago. (Use '!kill␣
 ↪2792' to kill it.)

<IPython.core.display.Javascript object>
```

Defining MLP model.

[ ]:
```
# MLP model
speech_model_mlp = tf.keras.Sequential()

speech_model_mlp.add(tf.keras.Input(shape=(5, 180)))
speech_model_mlp.add(Flatten())
speech_model_mlp.add(Dense(256, activation='relu'))
speech_model_mlp.add(Dense(128, activation='relu'))
speech_model_mlp.add(Dropout(0.3))
speech_model_mlp.add(Dense(8, activation='softmax'))

speech_model_mlp.summary()
optimiser = tf.keras.optimizers.Adam(learning_rate=0.001)
speech_model_mlp.compile(optimizer=optimiser,
                  loss='sparse_categorical_crossentropy',
                  metrics=['accuracy'])
```

```
Model: "sequential_19"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 flatten_8 (Flatten)         (None, 900)               0

 dense_51 (Dense)            (None, 256)               230656

 dense_52 (Dense)            (None, 128)               32896

 dropout_17 (Dropout)        (None, 128)               0

 dense_53 (Dense)            (None, 8)                 1032

=================================================================
Total params: 264,584
Trainable params: 264,584
Non-trainable params: 0
_________________________________________________________________
```

Training MLP model with Batch size = 16, learning rate = 0.001, epochs = 200

[ ]:
```
# # Training MLP model
le = preprocessing.LabelEncoder()
le.fit(np.array(y_train))
logdir = 'logs/mlp'
speech_model_mlp.fit(x_train,le.transform(np.array(y_train)), batch_size=16,␣
 ↪epochs=200, callbacks=[keras.callbacks.TensorBoard(log_dir=logdir)])
y_pred = le.inverse_transform(tf.keras.backend.argmax(speech_model_mlp.
 ↪predict(x_test)))
```

```
print('Accuracy with mlp - ', accuracy_score(y_true=y_test, y_pred=y_pred))

pickle.dump(speech_model_mlp, open('/content/drive/MyDrive/ECS271/
  ↪speech_emotion_mlp.sav', 'wb'))
%tensorboard --logdir logs
```

```
Epoch 1/200
68/68 [==============================] - 1s 3ms/step - loss: 15.9536 - accuracy:
0.1528
Epoch 2/200
68/68 [==============================] - 0s 3ms/step - loss: 2.0502 - accuracy:
0.1454
Epoch 3/200
68/68 [==============================] - 0s 3ms/step - loss: 2.0103 - accuracy:
0.1509
Epoch 4/200
68/68 [==============================] - 0s 3ms/step - loss: 2.0067 - accuracy:
0.1926
Epoch 5/200
68/68 [==============================] - 0s 3ms/step - loss: 2.0085 - accuracy:
0.1778
Epoch 6/200
68/68 [==============================] - 0s 3ms/step - loss: 1.9751 - accuracy:
0.1954
Epoch 7/200
68/68 [==============================] - 0s 3ms/step - loss: 1.9574 - accuracy:
0.2037
Epoch 8/200
68/68 [==============================] - 0s 3ms/step - loss: 1.9437 - accuracy:
0.1833
Epoch 9/200
68/68 [==============================] - 0s 3ms/step - loss: 1.9519 - accuracy:
0.2065
Epoch 10/200
68/68 [==============================] - 0s 3ms/step - loss: 1.9076 - accuracy:
0.2111
Epoch 11/200
68/68 [==============================] - 0s 3ms/step - loss: 1.9390 - accuracy:
0.2315
Epoch 12/200
68/68 [==============================] - 0s 3ms/step - loss: 1.9168 - accuracy:
0.2296
Epoch 13/200
68/68 [==============================] - 0s 3ms/step - loss: 1.8789 - accuracy:
0.2472
Epoch 14/200
68/68 [==============================] - 0s 3ms/step - loss: 1.8947 - accuracy:
0.2444
```

```
Epoch 15/200
68/68 [==============================] - 0s 3ms/step - loss: 1.8455 - accuracy:
0.2685
Epoch 16/200
68/68 [==============================] - 0s 3ms/step - loss: 1.8943 - accuracy:
0.2370
Epoch 17/200
68/68 [==============================] - 0s 3ms/step - loss: 1.8606 - accuracy:
0.2491
Epoch 18/200
68/68 [==============================] - 0s 3ms/step - loss: 1.8558 - accuracy:
0.2528
Epoch 19/200
68/68 [==============================] - 0s 3ms/step - loss: 1.8719 - accuracy:
0.2454
Epoch 20/200
68/68 [==============================] - 0s 3ms/step - loss: 1.8815 - accuracy:
0.2407
Epoch 21/200
68/68 [==============================] - 0s 3ms/step - loss: 1.8420 - accuracy:
0.2481
Epoch 22/200
68/68 [==============================] - 0s 3ms/step - loss: 1.8518 - accuracy:
0.2574
Epoch 23/200
68/68 [==============================] - 0s 3ms/step - loss: 1.8138 - accuracy:
0.2731
Epoch 24/200
68/68 [==============================] - 0s 3ms/step - loss: 1.8386 - accuracy:
0.2556
Epoch 25/200
68/68 [==============================] - 0s 3ms/step - loss: 1.8418 - accuracy:
0.2611
Epoch 26/200
68/68 [==============================] - 0s 3ms/step - loss: 1.7789 - accuracy:
0.2824
Epoch 27/200
68/68 [==============================] - 0s 3ms/step - loss: 1.8418 - accuracy:
0.2704
Epoch 28/200
68/68 [==============================] - 0s 3ms/step - loss: 1.8405 - accuracy:
0.2472
Epoch 29/200
68/68 [==============================] - 0s 3ms/step - loss: 1.7736 - accuracy:
0.3083
Epoch 30/200
68/68 [==============================] - 0s 3ms/step - loss: 1.8101 - accuracy:
0.2630
```

```
Epoch 31/200
68/68 [==============================] - 0s 3ms/step - loss: 1.7626 - accuracy:
0.2954
Epoch 32/200
68/68 [==============================] - 0s 3ms/step - loss: 1.7344 - accuracy:
0.3046
Epoch 33/200
68/68 [==============================] - 0s 3ms/step - loss: 1.7277 - accuracy:
0.3222
Epoch 34/200
68/68 [==============================] - 0s 3ms/step - loss: 1.7216 - accuracy:
0.3176
Epoch 35/200
68/68 [==============================] - 0s 3ms/step - loss: 1.7033 - accuracy:
0.3213
Epoch 36/200
68/68 [==============================] - 0s 3ms/step - loss: 1.7356 - accuracy:
0.3009
Epoch 37/200
68/68 [==============================] - 0s 3ms/step - loss: 1.7183 - accuracy:
0.3213
Epoch 38/200
68/68 [==============================] - 0s 3ms/step - loss: 1.7247 - accuracy:
0.3102
Epoch 39/200
68/68 [==============================] - 0s 3ms/step - loss: 1.7447 - accuracy:
0.2954
Epoch 40/200
68/68 [==============================] - 0s 3ms/step - loss: 1.7076 - accuracy:
0.3037
Epoch 41/200
68/68 [==============================] - 0s 3ms/step - loss: 1.6782 - accuracy:
0.3398
Epoch 42/200
68/68 [==============================] - 0s 3ms/step - loss: 1.6621 - accuracy:
0.3454
Epoch 43/200
68/68 [==============================] - 0s 3ms/step - loss: 1.6312 - accuracy:
0.3639
Epoch 44/200
68/68 [==============================] - 0s 3ms/step - loss: 1.6671 - accuracy:
0.3407
Epoch 45/200
68/68 [==============================] - 0s 3ms/step - loss: 1.6317 - accuracy:
0.3657
Epoch 46/200
68/68 [==============================] - 0s 3ms/step - loss: 1.6616 - accuracy:
0.3426
```

```
Epoch 47/200
68/68 [==============================] - 0s 3ms/step - loss: 1.6575 - accuracy:
0.3435
Epoch 48/200
68/68 [==============================] - 0s 3ms/step - loss: 1.6810 - accuracy:
0.3389
Epoch 49/200
68/68 [==============================] - 0s 3ms/step - loss: 1.6837 - accuracy:
0.3250
Epoch 50/200
68/68 [==============================] - 0s 3ms/step - loss: 1.6499 - accuracy:
0.3426
Epoch 51/200
68/68 [==============================] - 0s 3ms/step - loss: 1.6646 - accuracy:
0.3472
Epoch 52/200
68/68 [==============================] - 0s 3ms/step - loss: 1.7020 - accuracy:
0.3519
Epoch 53/200
68/68 [==============================] - 0s 3ms/step - loss: 1.7451 - accuracy:
0.3491
Epoch 54/200
68/68 [==============================] - 0s 4ms/step - loss: 1.6469 - accuracy:
0.3537
Epoch 55/200
68/68 [==============================] - 0s 3ms/step - loss: 1.5600 - accuracy:
0.3963
Epoch 56/200
68/68 [==============================] - 0s 3ms/step - loss: 1.7017 - accuracy:
0.3278
Epoch 57/200
68/68 [==============================] - 0s 3ms/step - loss: 1.7375 - accuracy:
0.3194
Epoch 58/200
68/68 [==============================] - 0s 3ms/step - loss: 1.6629 - accuracy:
0.3287
Epoch 59/200
68/68 [==============================] - 0s 3ms/step - loss: 1.6536 - accuracy:
0.3583
Epoch 60/200
68/68 [==============================] - 0s 3ms/step - loss: 1.6516 - accuracy:
0.3389
Epoch 61/200
68/68 [==============================] - 0s 3ms/step - loss: 1.5925 - accuracy:
0.3796
Epoch 62/200
68/68 [==============================] - 0s 3ms/step - loss: 1.5801 - accuracy:
0.3778
```

```
Epoch 63/200
68/68 [==============================] - 0s 3ms/step - loss: 1.5638 - accuracy:
0.3731
Epoch 64/200
68/68 [==============================] - 0s 3ms/step - loss: 1.6287 - accuracy:
0.3491
Epoch 65/200
68/68 [==============================] - 0s 3ms/step - loss: 1.6500 - accuracy:
0.3574
Epoch 66/200
68/68 [==============================] - 0s 3ms/step - loss: 1.6719 - accuracy:
0.3176
Epoch 67/200
68/68 [==============================] - 0s 3ms/step - loss: 1.6018 - accuracy:
0.3620
Epoch 68/200
68/68 [==============================] - 0s 3ms/step - loss: 1.5334 - accuracy:
0.4009
Epoch 69/200
68/68 [==============================] - 0s 3ms/step - loss: 1.5402 - accuracy:
0.4213
Epoch 70/200
68/68 [==============================] - 0s 3ms/step - loss: 1.5147 - accuracy:
0.4019
Epoch 71/200
68/68 [==============================] - 0s 3ms/step - loss: 1.4815 - accuracy:
0.4231
Epoch 72/200
68/68 [==============================] - 0s 3ms/step - loss: 1.5058 - accuracy:
0.4130
Epoch 73/200
68/68 [==============================] - 0s 3ms/step - loss: 1.4545 - accuracy:
0.4213
Epoch 74/200
68/68 [==============================] - 0s 3ms/step - loss: 1.4603 - accuracy:
0.4213
Epoch 75/200
68/68 [==============================] - 0s 3ms/step - loss: 1.4470 - accuracy:
0.4287
Epoch 76/200
68/68 [==============================] - 0s 3ms/step - loss: 1.4643 - accuracy:
0.4352
Epoch 77/200
68/68 [==============================] - 0s 3ms/step - loss: 1.4549 - accuracy:
0.4398
Epoch 78/200
68/68 [==============================] - 0s 3ms/step - loss: 1.4109 - accuracy:
0.4509
```

```
Epoch 79/200
68/68 [==============================] - 0s 3ms/step - loss: 1.4542 - accuracy:
0.4333
Epoch 80/200
68/68 [==============================] - 0s 3ms/step - loss: 1.4671 - accuracy:
0.4287
Epoch 81/200
68/68 [==============================] - 0s 3ms/step - loss: 1.5144 - accuracy:
0.4093
Epoch 82/200
68/68 [==============================] - 0s 3ms/step - loss: 1.4639 - accuracy:
0.4444
Epoch 83/200
68/68 [==============================] - 0s 3ms/step - loss: 1.4603 - accuracy:
0.4278
Epoch 84/200
68/68 [==============================] - 0s 3ms/step - loss: 1.4011 - accuracy:
0.4537
Epoch 85/200
68/68 [==============================] - 0s 3ms/step - loss: 1.3967 - accuracy:
0.4648
Epoch 86/200
68/68 [==============================] - 0s 3ms/step - loss: 1.3804 - accuracy:
0.4602
Epoch 87/200
68/68 [==============================] - 0s 3ms/step - loss: 1.3450 - accuracy:
0.4833
Epoch 88/200
68/68 [==============================] - 0s 3ms/step - loss: 1.3659 - accuracy:
0.4741
Epoch 89/200
68/68 [==============================] - 0s 4ms/step - loss: 1.2911 - accuracy:
0.4954
Epoch 90/200
68/68 [==============================] - 0s 3ms/step - loss: 1.3624 - accuracy:
0.4787
Epoch 91/200
68/68 [==============================] - 0s 3ms/step - loss: 1.3029 - accuracy:
0.5028
Epoch 92/200
68/68 [==============================] - 0s 3ms/step - loss: 1.3032 - accuracy:
0.5065
Epoch 93/200
68/68 [==============================] - 0s 3ms/step - loss: 1.2665 - accuracy:
0.5213
Epoch 94/200
68/68 [==============================] - 0s 3ms/step - loss: 1.2573 - accuracy:
0.5222
```

```
Epoch 95/200
68/68 [==============================] - 0s 3ms/step - loss: 1.2057 - accuracy:
0.5306
Epoch 96/200
68/68 [==============================] - 0s 3ms/step - loss: 1.2122 - accuracy:
0.5509
Epoch 97/200
68/68 [==============================] - 0s 3ms/step - loss: 1.1832 - accuracy:
0.5500
Epoch 98/200
68/68 [==============================] - 0s 3ms/step - loss: 1.3444 - accuracy:
0.5046
Epoch 99/200
68/68 [==============================] - 0s 4ms/step - loss: 1.2802 - accuracy:
0.5130
Epoch 100/200
68/68 [==============================] - 0s 3ms/step - loss: 1.1929 - accuracy:
0.5398
Epoch 101/200
68/68 [==============================] - 0s 3ms/step - loss: 1.2104 - accuracy:
0.5380
Epoch 102/200
68/68 [==============================] - 0s 3ms/step - loss: 1.1909 - accuracy:
0.5491
Epoch 103/200
68/68 [==============================] - 0s 3ms/step - loss: 1.1346 - accuracy:
0.5685
Epoch 104/200
68/68 [==============================] - 0s 3ms/step - loss: 1.1562 - accuracy:
0.5870
Epoch 105/200
68/68 [==============================] - 0s 3ms/step - loss: 1.1842 - accuracy:
0.5556
Epoch 106/200
68/68 [==============================] - 0s 3ms/step - loss: 1.1944 - accuracy:
0.5602
Epoch 107/200
68/68 [==============================] - 0s 3ms/step - loss: 1.0693 - accuracy:
0.5963
Epoch 108/200
68/68 [==============================] - 0s 3ms/step - loss: 1.0312 - accuracy:
0.6157
Epoch 109/200
68/68 [==============================] - 0s 3ms/step - loss: 1.0803 - accuracy:
0.6065
Epoch 110/200
68/68 [==============================] - 0s 3ms/step - loss: 1.1416 - accuracy:
0.5759
```

```
Epoch 111/200
68/68 [==============================] - 0s 3ms/step - loss: 1.0179 - accuracy:
0.6306
Epoch 112/200
68/68 [==============================] - 0s 3ms/step - loss: 1.0277 - accuracy:
0.6019
Epoch 113/200
68/68 [==============================] - 0s 3ms/step - loss: 0.9741 - accuracy:
0.6370
Epoch 114/200
68/68 [==============================] - 0s 3ms/step - loss: 0.9659 - accuracy:
0.6472
Epoch 115/200
68/68 [==============================] - 0s 3ms/step - loss: 0.9808 - accuracy:
0.6315
Epoch 116/200
68/68 [==============================] - 0s 3ms/step - loss: 0.9321 - accuracy:
0.6537
Epoch 117/200
68/68 [==============================] - 0s 3ms/step - loss: 0.9532 - accuracy:
0.6630
Epoch 118/200
68/68 [==============================] - 0s 3ms/step - loss: 0.9531 - accuracy:
0.6370
Epoch 119/200
68/68 [==============================] - 0s 3ms/step - loss: 0.9613 - accuracy:
0.6491
Epoch 120/200
68/68 [==============================] - 0s 3ms/step - loss: 0.9453 - accuracy:
0.6583
Epoch 121/200
68/68 [==============================] - 0s 3ms/step - loss: 0.8609 - accuracy:
0.6889
Epoch 122/200
68/68 [==============================] - 0s 3ms/step - loss: 0.8117 - accuracy:
0.6926
Epoch 123/200
68/68 [==============================] - 0s 3ms/step - loss: 0.8652 - accuracy:
0.6759
Epoch 124/200
68/68 [==============================] - 0s 3ms/step - loss: 0.7882 - accuracy:
0.6972
Epoch 125/200
68/68 [==============================] - 0s 3ms/step - loss: 0.8297 - accuracy:
0.7083
Epoch 126/200
68/68 [==============================] - 0s 3ms/step - loss: 0.8158 - accuracy:
0.6972
```

```
Epoch 127/200
68/68 [==============================] - 0s 3ms/step - loss: 0.7376 - accuracy:
0.7213
Epoch 128/200
68/68 [==============================] - 0s 3ms/step - loss: 0.8882 - accuracy:
0.6843
Epoch 129/200
68/68 [==============================] - 0s 3ms/step - loss: 0.8576 - accuracy:
0.6713
Epoch 130/200
68/68 [==============================] - 0s 3ms/step - loss: 0.7566 - accuracy:
0.7269
Epoch 131/200
68/68 [==============================] - 0s 3ms/step - loss: 0.6914 - accuracy:
0.7426
Epoch 132/200
68/68 [==============================] - 0s 3ms/step - loss: 0.7252 - accuracy:
0.7380
Epoch 133/200
68/68 [==============================] - 0s 3ms/step - loss: 0.7625 - accuracy:
0.7194
Epoch 134/200
68/68 [==============================] - 0s 3ms/step - loss: 0.7011 - accuracy:
0.7454
Epoch 135/200
68/68 [==============================] - 0s 3ms/step - loss: 0.7135 - accuracy:
0.7222
Epoch 136/200
68/68 [==============================] - 0s 3ms/step - loss: 0.7954 - accuracy:
0.7157
Epoch 137/200
68/68 [==============================] - 0s 3ms/step - loss: 0.6924 - accuracy:
0.7417
Epoch 138/200
68/68 [==============================] - 0s 3ms/step - loss: 0.7451 - accuracy:
0.7556
Epoch 139/200
68/68 [==============================] - 0s 3ms/step - loss: 0.7364 - accuracy:
0.7333
Epoch 140/200
68/68 [==============================] - 0s 3ms/step - loss: 0.6704 - accuracy:
0.7546
Epoch 141/200
68/68 [==============================] - 0s 3ms/step - loss: 0.6327 - accuracy:
0.7556
Epoch 142/200
68/68 [==============================] - 0s 3ms/step - loss: 0.6257 - accuracy:
0.7639
```

```
Epoch 143/200
68/68 [==============================] - 0s 3ms/step - loss: 0.5981 - accuracy:
0.7722
Epoch 144/200
68/68 [==============================] - 0s 3ms/step - loss: 0.5918 - accuracy:
0.7731
Epoch 145/200
68/68 [==============================] - 0s 4ms/step - loss: 0.5699 - accuracy:
0.7954
Epoch 146/200
68/68 [==============================] - 0s 3ms/step - loss: 0.5556 - accuracy:
0.7861
Epoch 147/200
68/68 [==============================] - 0s 3ms/step - loss: 0.5567 - accuracy:
0.7981
Epoch 148/200
68/68 [==============================] - 0s 3ms/step - loss: 0.4987 - accuracy:
0.7954
Epoch 149/200
68/68 [==============================] - 0s 3ms/step - loss: 0.5365 - accuracy:
0.7852
Epoch 150/200
68/68 [==============================] - 0s 3ms/step - loss: 0.4937 - accuracy:
0.8231
Epoch 151/200
68/68 [==============================] - 0s 3ms/step - loss: 0.5062 - accuracy:
0.8083
Epoch 152/200
68/68 [==============================] - 0s 3ms/step - loss: 0.5084 - accuracy:
0.8111
Epoch 153/200
68/68 [==============================] - 0s 3ms/step - loss: 0.4635 - accuracy:
0.8204
Epoch 154/200
68/68 [==============================] - 0s 3ms/step - loss: 0.5236 - accuracy:
0.8019
Epoch 155/200
68/68 [==============================] - 0s 3ms/step - loss: 0.4786 - accuracy:
0.8176
Epoch 156/200
68/68 [==============================] - 0s 3ms/step - loss: 0.4757 - accuracy:
0.8250
Epoch 157/200
68/68 [==============================] - 0s 3ms/step - loss: 0.4228 - accuracy:
0.8500
Epoch 158/200
68/68 [==============================] - 0s 3ms/step - loss: 0.5710 - accuracy:
0.7972
```

```
Epoch 159/200
68/68 [==============================] - 0s 3ms/step - loss: 0.5450 - accuracy:
0.7898
Epoch 160/200
68/68 [==============================] - 0s 3ms/step - loss: 0.4268 - accuracy:
0.8370
Epoch 161/200
68/68 [==============================] - 0s 3ms/step - loss: 0.4002 - accuracy:
0.8389
Epoch 162/200
68/68 [==============================] - 0s 3ms/step - loss: 0.4382 - accuracy:
0.8389
Epoch 163/200
68/68 [==============================] - 0s 3ms/step - loss: 0.4888 - accuracy:
0.8176
Epoch 164/200
68/68 [==============================] - 0s 3ms/step - loss: 0.4207 - accuracy:
0.8519
Epoch 165/200
68/68 [==============================] - 0s 3ms/step - loss: 0.3680 - accuracy:
0.8611
Epoch 166/200
68/68 [==============================] - 0s 3ms/step - loss: 0.3723 - accuracy:
0.8593
Epoch 167/200
68/68 [==============================] - 0s 3ms/step - loss: 0.3052 - accuracy:
0.8843
Epoch 168/200
68/68 [==============================] - 0s 3ms/step - loss: 0.3264 - accuracy:
0.8694
Epoch 169/200
68/68 [==============================] - 0s 3ms/step - loss: 0.4350 - accuracy:
0.8435
Epoch 170/200
68/68 [==============================] - 0s 3ms/step - loss: 0.4456 - accuracy:
0.8333
Epoch 171/200
68/68 [==============================] - 0s 3ms/step - loss: 0.3894 - accuracy:
0.8611
Epoch 172/200
68/68 [==============================] - 0s 3ms/step - loss: 0.3319 - accuracy:
0.8713
Epoch 173/200
68/68 [==============================] - 0s 3ms/step - loss: 0.3801 - accuracy:
0.8546
Epoch 174/200
68/68 [==============================] - 0s 3ms/step - loss: 0.4995 - accuracy:
0.8130
```

```
Epoch 175/200
68/68 [==============================] - 0s 3ms/step - loss: 0.3691 - accuracy:
0.8611
Epoch 176/200
68/68 [==============================] - 0s 3ms/step - loss: 0.3165 - accuracy:
0.8833
Epoch 177/200
68/68 [==============================] - 0s 3ms/step - loss: 0.2912 - accuracy:
0.8898
Epoch 178/200
68/68 [==============================] - 0s 3ms/step - loss: 0.3079 - accuracy:
0.8815
Epoch 179/200
68/68 [==============================] - 0s 3ms/step - loss: 0.3723 - accuracy:
0.8602
Epoch 180/200
68/68 [==============================] - 0s 3ms/step - loss: 0.3276 - accuracy:
0.8787
Epoch 181/200
68/68 [==============================] - 0s 3ms/step - loss: 0.3370 - accuracy:
0.8806
Epoch 182/200
68/68 [==============================] - 0s 3ms/step - loss: 0.3043 - accuracy:
0.8954
Epoch 183/200
68/68 [==============================] - 0s 3ms/step - loss: 0.3423 - accuracy:
0.8731
Epoch 184/200
68/68 [==============================] - 0s 3ms/step - loss: 0.2929 - accuracy:
0.8926
Epoch 185/200
68/68 [==============================] - 0s 3ms/step - loss: 0.2695 - accuracy:
0.8898
Epoch 186/200
68/68 [==============================] - 0s 3ms/step - loss: 0.2450 - accuracy:
0.9083
Epoch 187/200
68/68 [==============================] - 0s 3ms/step - loss: 0.3324 - accuracy:
0.8741
Epoch 188/200
68/68 [==============================] - 0s 3ms/step - loss: 0.2382 - accuracy:
0.9102
Epoch 189/200
68/68 [==============================] - 0s 3ms/step - loss: 0.2320 - accuracy:
0.9093
Epoch 190/200
68/68 [==============================] - 0s 4ms/step - loss: 0.2641 - accuracy:
0.8981
```

```
Epoch 191/200
68/68 [==============================] - 0s 3ms/step - loss: 0.2041 - accuracy:
0.9185
Epoch 192/200
68/68 [==============================] - 0s 3ms/step - loss: 0.2145 - accuracy:
0.9194
Epoch 193/200
68/68 [==============================] - 0s 3ms/step - loss: 0.2435 - accuracy:
0.9019
Epoch 194/200
68/68 [==============================] - 0s 3ms/step - loss: 0.2405 - accuracy:
0.9093
Epoch 195/200
68/68 [==============================] - 0s 3ms/step - loss: 0.3503 - accuracy:
0.8713
Epoch 196/200
68/68 [==============================] - 0s 3ms/step - loss: 0.2303 - accuracy:
0.9185
Epoch 197/200
68/68 [==============================] - 0s 3ms/step - loss: 0.1749 - accuracy:
0.9389
Epoch 198/200
68/68 [==============================] - 0s 3ms/step - loss: 0.2464 - accuracy:
0.8981
Epoch 199/200
68/68 [==============================] - 0s 3ms/step - loss: 0.2304 - accuracy:
0.9111
Epoch 200/200
68/68 [==============================] - 0s 3ms/step - loss: 0.3625 - accuracy:
0.8685
12/12 [==============================] - 0s 2ms/step
Accuracy with mlp -  0.6277777777777778

Reusing TensorBoard on port 6006 (pid 2792), started 0:07:27 ago. (Use '!kill␣
 ↪2792' to kill it.)

<IPython.core.display.Javascript object>
```

Defining CNN model

[ ]:
```
# CNN model
speech_model_cnn = tf.keras.Sequential()

speech_model_cnn.add(tf.keras.Input(shape=(5, 180, 1)))
speech_model_cnn.add(Conv2D(2, 4,  padding='same'))
speech_model_cnn.add(Activation('relu'))
speech_model_cnn.add(AveragePooling2D(pool_size=(2)))
speech_model_cnn.add(Conv2D(2, 2, padding='same'))
speech_model_cnn.add(AveragePooling2D(pool_size=(2)))
```

```
speech_model_cnn.add(Flatten())
speech_model_cnn.add(Dense(2048, activation='relu'))
speech_model_cnn.add(Dense(1024, activation='relu'))
speech_model_cnn.add(Dropout(0.3))
speech_model_cnn.add(Dense(8, activation='softmax'))

speech_model_cnn.summary()
optimiser = tf.keras.optimizers.Adam(learning_rate=0.001)
speech_model_cnn.compile(optimizer=optimiser,
                  loss='sparse_categorical_crossentropy',
                  metrics=['accuracy'])
```

```
Model: "sequential_17"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_12 (Conv2D)          (None, 5, 180, 2)         34

 activation_8 (Activation)   (None, 5, 180, 2)         0

 average_pooling2d_10 (Avera  (None, 2, 90, 2)         0
 gePooling2D)

 conv2d_13 (Conv2D)          (None, 2, 90, 2)          18

 average_pooling2d_11 (Avera  (None, 1, 45, 2)         0
 gePooling2D)

 flatten_7 (Flatten)         (None, 90)                0

 dense_45 (Dense)            (None, 2048)              186368

 dense_46 (Dense)            (None, 1024)              2098176

 dropout_15 (Dropout)        (None, 1024)              0

 dense_47 (Dense)            (None, 8)                 8200

=================================================================
Total params: 2,292,796
Trainable params: 2,292,796
Non-trainable params: 0
_________________________________________________________________
```

Training CNN model with Batch size = 16, learning rate = 0.001, epochs = 100

[ ]:
```
# Training CNN model
le = preprocessing.LabelEncoder()
```

```
le.fit(np.array(y_train))
logdir = 'logs/cnn'
tensorboard_callback = keras.callbacks.TensorBoard(log_dir=logdir)
speech_model_cnn.fit(np.reshape(x_train, (x_train.shape[0], x_train.shape[1],␣
 ↪x_train.shape[2], 1)),le.transform(np.array(y_train)), batch_size=16,␣
 ↪epochs=100, callbacks=[tensorboard_callback])
y_pred = le.inverse_transform(tf.keras.backend.argmax(speech_model_cnn.
 ↪predict(np.reshape(x_test, (x_test.shape[0], x_test.shape[1], x_test.
 ↪shape[2], 1)))))
print('Accuracy with cnn - ', accuracy_score(y_true=y_test, y_pred=y_pred))

pickle.dump(speech_model_cnn, open('/content/drive/MyDrive/ECS271/
 ↪speech_emotion_cnn.sav', 'wb'))
%tensorboard --logdir logs
```

```
Epoch 1/100
68/68 [==============================] - 3s 30ms/step - loss: 3.2028 - accuracy:
0.1880
Epoch 2/100
68/68 [==============================] - 2s 30ms/step - loss: 1.8148 - accuracy:
0.2713
Epoch 3/100
68/68 [==============================] - 2s 31ms/step - loss: 1.7179 - accuracy:
0.3250
Epoch 4/100
68/68 [==============================] - 2s 31ms/step - loss: 1.6838 - accuracy:
0.3380
Epoch 5/100
68/68 [==============================] - 3s 39ms/step - loss: 1.6116 - accuracy:
0.3741
Epoch 6/100
68/68 [==============================] - 4s 55ms/step - loss: 1.6008 - accuracy:
0.3824
Epoch 7/100
68/68 [==============================] - 2s 31ms/step - loss: 1.5656 - accuracy:
0.3880
Epoch 8/100
68/68 [==============================] - 2s 31ms/step - loss: 1.5075 - accuracy:
0.4102
Epoch 9/100
68/68 [==============================] - 2s 31ms/step - loss: 1.4786 - accuracy:
0.4231
Epoch 10/100
68/68 [==============================] - 2s 31ms/step - loss: 1.4434 - accuracy:
0.4528
Epoch 11/100
68/68 [==============================] - 2s 31ms/step - loss: 1.4035 - accuracy:
```

```
0.4593
Epoch 12/100
68/68 [==============================] - 2s 31ms/step - loss: 1.3597 - accuracy:
0.4796
Epoch 13/100
68/68 [==============================] - 2s 30ms/step - loss: 1.3086 - accuracy:
0.5130
Epoch 14/100
68/68 [==============================] - 2s 31ms/step - loss: 1.2985 - accuracy:
0.5028
Epoch 15/100
68/68 [==============================] - 2s 30ms/step - loss: 1.2703 - accuracy:
0.5157
Epoch 16/100
68/68 [==============================] - 2s 31ms/step - loss: 1.2863 - accuracy:
0.5130
Epoch 17/100
68/68 [==============================] - 2s 31ms/step - loss: 1.1913 - accuracy:
0.5241
Epoch 18/100
68/68 [==============================] - 2s 31ms/step - loss: 1.1825 - accuracy:
0.5352
Epoch 19/100
68/68 [==============================] - 2s 31ms/step - loss: 1.2006 - accuracy:
0.5806
Epoch 20/100
68/68 [==============================] - 2s 31ms/step - loss: 1.1297 - accuracy:
0.5870
Epoch 21/100
68/68 [==============================] - 2s 30ms/step - loss: 1.1943 - accuracy:
0.5611
Epoch 22/100
68/68 [==============================] - 2s 31ms/step - loss: 1.0285 - accuracy:
0.6000
Epoch 23/100
68/68 [==============================] - 2s 30ms/step - loss: 1.0222 - accuracy:
0.6194
Epoch 24/100
68/68 [==============================] - 2s 30ms/step - loss: 0.9289 - accuracy:
0.6481
Epoch 25/100
68/68 [==============================] - 2s 30ms/step - loss: 0.8968 - accuracy:
0.6556
Epoch 26/100
68/68 [==============================] - 2s 30ms/step - loss: 0.9012 - accuracy:
0.6620
Epoch 27/100
68/68 [==============================] - 2s 30ms/step - loss: 0.8387 - accuracy:
```

```
0.6935
Epoch 28/100
68/68 [==============================] - 2s 30ms/step - loss: 0.7994 - accuracy:
0.7000
Epoch 29/100
68/68 [==============================] - 2s 30ms/step - loss: 0.7332 - accuracy:
0.7194
Epoch 30/100
68/68 [==============================] - 2s 29ms/step - loss: 0.7104 - accuracy:
0.7204
Epoch 31/100
68/68 [==============================] - 2s 30ms/step - loss: 0.7188 - accuracy:
0.7185
Epoch 32/100
68/68 [==============================] - 2s 30ms/step - loss: 0.6913 - accuracy:
0.7546
Epoch 33/100
68/68 [==============================] - 2s 30ms/step - loss: 0.6777 - accuracy:
0.7778
Epoch 34/100
68/68 [==============================] - 2s 30ms/step - loss: 0.6055 - accuracy:
0.7741
Epoch 35/100
68/68 [==============================] - 2s 29ms/step - loss: 0.5319 - accuracy:
0.8139
Epoch 36/100
68/68 [==============================] - 2s 30ms/step - loss: 0.5374 - accuracy:
0.7954
Epoch 37/100
68/68 [==============================] - 2s 30ms/step - loss: 0.4625 - accuracy:
0.8324
Epoch 38/100
68/68 [==============================] - 2s 29ms/step - loss: 0.4336 - accuracy:
0.8491
Epoch 39/100
68/68 [==============================] - 2s 30ms/step - loss: 0.4225 - accuracy:
0.8417
Epoch 40/100
68/68 [==============================] - 2s 30ms/step - loss: 0.3727 - accuracy:
0.8648
Epoch 41/100
68/68 [==============================] - 2s 30ms/step - loss: 0.3500 - accuracy:
0.8685
Epoch 42/100
68/68 [==============================] - 2s 30ms/step - loss: 0.3409 - accuracy:
0.8917
Epoch 43/100
68/68 [==============================] - 2s 29ms/step - loss: 0.3284 - accuracy:
```

```
0.8991
Epoch 44/100
68/68 [==============================] - 2s 30ms/step - loss: 0.6738 - accuracy:
0.8537
Epoch 45/100
68/68 [==============================] - 2s 30ms/step - loss: 0.3857 - accuracy:
0.8815
Epoch 46/100
68/68 [==============================] - 2s 29ms/step - loss: 0.3010 - accuracy:
0.8972
Epoch 47/100
68/68 [==============================] - 2s 29ms/step - loss: 0.2151 - accuracy:
0.9389
Epoch 48/100
68/68 [==============================] - 2s 30ms/step - loss: 0.2137 - accuracy:
0.9315
Epoch 49/100
68/68 [==============================] - 2s 30ms/step - loss: 0.2228 - accuracy:
0.9269
Epoch 50/100
68/68 [==============================] - 2s 30ms/step - loss: 0.2050 - accuracy:
0.9352
Epoch 51/100
68/68 [==============================] - 2s 30ms/step - loss: 0.1656 - accuracy:
0.9528
Epoch 52/100
68/68 [==============================] - 2s 30ms/step - loss: 0.1822 - accuracy:
0.9389
Epoch 53/100
68/68 [==============================] - 2s 30ms/step - loss: 0.1351 - accuracy:
0.9565
Epoch 54/100
68/68 [==============================] - 2s 30ms/step - loss: 0.1462 - accuracy:
0.9565
Epoch 55/100
68/68 [==============================] - 2s 29ms/step - loss: 0.1287 - accuracy:
0.9630
Epoch 56/100
68/68 [==============================] - 2s 30ms/step - loss: 0.1154 - accuracy:
0.9685
Epoch 57/100
68/68 [==============================] - 2s 30ms/step - loss: 0.1320 - accuracy:
0.9565
Epoch 58/100
68/68 [==============================] - 2s 29ms/step - loss: 0.0940 - accuracy:
0.9750
Epoch 59/100
68/68 [==============================] - 2s 30ms/step - loss: 0.0915 - accuracy:
```

```
0.9657
Epoch 60/100
68/68 [==============================] - 2s 29ms/step - loss: 0.1204 - accuracy:
0.9583
Epoch 61/100
68/68 [==============================] - 2s 30ms/step - loss: 0.0997 - accuracy:
0.9722
Epoch 62/100
68/68 [==============================] - 4s 54ms/step - loss: 0.0715 - accuracy:
0.9843
Epoch 63/100
68/68 [==============================] - 3s 40ms/step - loss: 0.0633 - accuracy:
0.9796
Epoch 64/100
68/68 [==============================] - 2s 30ms/step - loss: 0.0699 - accuracy:
0.9806
Epoch 65/100
68/68 [==============================] - 2s 29ms/step - loss: 0.0719 - accuracy:
0.9815
Epoch 66/100
68/68 [==============================] - 2s 30ms/step - loss: 0.1378 - accuracy:
0.9519
Epoch 67/100
68/68 [==============================] - 2s 30ms/step - loss: 0.1090 - accuracy:
0.9667
Epoch 68/100
68/68 [==============================] - 2s 30ms/step - loss: 0.0774 - accuracy:
0.9759
Epoch 69/100
68/68 [==============================] - 2s 30ms/step - loss: 0.1259 - accuracy:
0.9657
Epoch 70/100
68/68 [==============================] - 2s 30ms/step - loss: 0.1110 - accuracy:
0.9741
Epoch 71/100
68/68 [==============================] - 2s 30ms/step - loss: 0.1236 - accuracy:
0.9639
Epoch 72/100
68/68 [==============================] - 2s 30ms/step - loss: 0.0899 - accuracy:
0.9722
Epoch 73/100
68/68 [==============================] - 2s 30ms/step - loss: 0.0719 - accuracy:
0.9824
Epoch 74/100
68/68 [==============================] - 2s 29ms/step - loss: 0.0504 - accuracy:
0.9898
Epoch 75/100
68/68 [==============================] - 2s 29ms/step - loss: 0.0640 - accuracy:
```

```
0.9898
Epoch 76/100
68/68 [==============================] - 2s 30ms/step - loss: 0.0371 - accuracy:
0.9917
Epoch 77/100
68/68 [==============================] - 2s 30ms/step - loss: 0.0939 - accuracy:
0.9769
Epoch 78/100
68/68 [==============================] - 2s 30ms/step - loss: 0.1214 - accuracy:
0.9667
Epoch 79/100
68/68 [==============================] - 2s 30ms/step - loss: 0.1288 - accuracy:
0.9620
Epoch 80/100
68/68 [==============================] - 2s 30ms/step - loss: 0.0668 - accuracy:
0.9750
Epoch 81/100
68/68 [==============================] - 2s 30ms/step - loss: 0.0249 - accuracy:
0.9963
Epoch 82/100
68/68 [==============================] - 2s 30ms/step - loss: 0.0142 - accuracy:
0.9991
Epoch 83/100
68/68 [==============================] - 2s 30ms/step - loss: 0.0175 - accuracy:
0.9954
Epoch 84/100
68/68 [==============================] - 2s 30ms/step - loss: 0.0124 - accuracy:
0.9981
Epoch 85/100
68/68 [==============================] - 2s 30ms/step - loss: 0.0142 - accuracy:
0.9981
Epoch 86/100
68/68 [==============================] - 2s 30ms/step - loss: 0.0084 - accuracy:
1.0000
Epoch 87/100
68/68 [==============================] - 2s 30ms/step - loss: 0.0365 - accuracy:
0.9926
Epoch 88/100
68/68 [==============================] - 2s 29ms/step - loss: 0.0597 - accuracy:
0.9796
Epoch 89/100
68/68 [==============================] - 2s 30ms/step - loss: 0.3170 - accuracy:
0.9093
Epoch 90/100
68/68 [==============================] - 2s 30ms/step - loss: 0.3049 - accuracy:
0.9380
Epoch 91/100
68/68 [==============================] - 2s 30ms/step - loss: 0.0965 - accuracy:
```

```
0.9750
Epoch 92/100
68/68 [==============================] - 2s 30ms/step - loss: 0.1015 - accuracy:
0.9806
Epoch 93/100
68/68 [==============================] - 2s 30ms/step - loss: 0.4987 - accuracy:
0.9694
Epoch 94/100
68/68 [==============================] - 2s 30ms/step - loss: 0.2431 - accuracy:
0.9694
Epoch 95/100
68/68 [==============================] - 2s 30ms/step - loss: 0.1876 - accuracy:
0.9657
Epoch 96/100
68/68 [==============================] - 2s 30ms/step - loss: 0.1386 - accuracy:
0.9741
Epoch 97/100
68/68 [==============================] - 2s 30ms/step - loss: 0.0592 - accuracy:
0.9824
Epoch 98/100
68/68 [==============================] - 2s 30ms/step - loss: 0.0184 - accuracy:
0.9963
Epoch 99/100
68/68 [==============================] - 2s 31ms/step - loss: 0.0118 - accuracy:
0.9981
Epoch 100/100
68/68 [==============================] - 2s 30ms/step - loss: 0.0212 - accuracy:
0.9935
12/12 [==============================] - 0s 11ms/step
Accuracy with cnn -  0.5583333333333333

WARNING:absl:Found untraced functions such as _jit_compiled_convolution_op,
_jit_compiled_convolution_op while saving (showing 2 of 2). These functions will
not be directly callable after loading.

<IPython.core.display.Javascript object>
```

Code for recording audio using javascript

[11]:
```
"""
Recording audio
Reference: https://gist.github.com/korakot/c21c3476c024ad6d56d5f48b0bca92be
"""

RECORD = """
const sleep = time => new Promise(resolve => setTimeout(resolve, time))
const b2text = blob => new Promise(resolve => {
    const reader = new FileReader()
    reader.onloadend = e => resolve(e.srcElement.result)
```

```
    reader.readAsDataURL(blob)
})
var record = time => new Promise(async resolve => {
    stream = await navigator.mediaDevices.getUserMedia({
        audio: true
    })
    recorder = new MediaRecorder(stream)
    chunks = []
    recorder.ondataavailable = e => chunks.push(e.data)
    recorder.start()
    await sleep(time)
    recorder.onstop = async () => {
        blob = new Blob(chunks)
        text = await b2text(blob)
        resolve(text)
    }
    recorder.stop()
})
"""

def record_audio(sec=3):
  display(Javascript(RECORD))
  audio_js = output.eval_js('record(%d)' % (sec*1000))
  base64encoded = b64decode(audio_js.split(',')[1])
  with open('recording.wav','wb') as f:
    f.write(base64encoded)
  x,_ = librosa.load('recording.wav', sr=48000)
  soundfile.write('recording.wav', x, 48000)
```

Final testing. Speak after pressing ENTER key. The image will be generated after 300 iterations.

[16]:
```
"""
Recording and audio and sending to LSTM model to fetch emotions and text. Then␣
 ↪iterating
over VQGAN+CLIP to generate image
"""
torch.cuda.empty_cache()
loaded_model = pickle.load(open('/content/drive/MyDrive/ECS271/
 ↪speech_emotion_lstm.sav', 'rb'))
input('Press Enter to start recording audio')
record_audio(3)
file_path='recording.wav'
display(Audio(file_path))
text = get_text(file_path)
init_image = "/content/drive/MyDrive/ECS271/base_images/" +␣
 ↪get_emotion(file_path) + ".png"
img = get_image(
```

```
    text = text,
    max_iterations = 300,
    width = 480,
    height = 480,
    init_image = init_image
)
plt.figure(figsize=(9,9))
plt.imshow(img)
plt.axis('off')
```

```
Press Enter to start recording audio
```

```
<IPython.core.display.Javascript object>
```

```
<IPython.lib.display.Audio object>
```

```
result2:
{   'alternative': [   {   'confidence': 0.73696381,
                           'transcript': 'dark in Bossier City'},
                       {'transcript': 'dark Rainbow City'},
                       {'transcript': 'dark in what city'},
                       {'transcript': 'Dark Angel City'},
                       {'transcript': 'start in Bossier City'}],
    'final': True}
1/1 [==============================] - 1s 718ms/step
Using texts: ['dark in Bossier City']
Using seed: 15942641373436208158
Using init image /content/drive/MyDrive/ECS271/base_images/fearful.png
```

```
0it [00:00, ?it/s]
```

```
i: 0, loss: 0.893858
i: 50, loss: 0.87155
i: 100, loss: 0.80785
i: 150, loss: 0.795381
i: 200, loss: 0.803648
i: 250, loss: 0.773702
i: 300, loss: 0.767927
```

[16]: (-0.5, 479.5, 479.5, -0.5)