YAMNet

YAMNet is a pretrained deep net that predicts 521 audio event classes based on the AudioSet-YouTube corpus, and employing the Mobilenet_v1 depthwise-separable convolution architecture.

This directory contains the Keras code to construct the model, and example code for applying the model to input sound files.

Installation

YAMNet depends on the following Python packages:

- `numpy`
- `resampy`
- `tensorflow`
- `pysoundfile`

These are all easily installable via, e.g., `pip install numpy` (as in the example command sequence below). Any reasonably recent version of these packages should work.

YAMNet also requires downloading the following data file:

- YAMNet model weights in Keras saved weights in HDF5 format.

After downloading this file into the same directory as this README, the installation can be tested by running `python yamnet_test.py` which runs some synthetic signals through the model and checks the outputs.

Here's a sample installation and test session:

```
# Upgrade pip first. Also make sure wheel is installed.
python -m pip install --upgrade pip wheel

# Install dependences.
pip install numpy resampy tensorflow soundfile

# Clone TensorFlow models repo into a 'models' directory.
git clone https://github.com/tensorflow/models.git
cd models/research/audioset/yamnet
# Download data file into same directory as code.
curl -O https://storage.googleapis.com/audioset/yamnet.h5

# Installation ready, let's test it.
python yamnet_test.py
# If we see "Ran 4 tests ... OK ...", then we're all set.
```

Usage

You can run the model over existing soundfiles using inference.py:

```
python inference.py input_sound.wav
```

The code will report the top-5 highest-scoring classes averaged over all the frames of the input. You can access greater detail by modifying the example code in inference.py.

See the jupyter notebook `yamnet_visualization.ipynb` for an example of displaying the per-frame model output scores.

About the Model

The YAMNet code layout is as follows:

- `yamnet.py` : Model definition in Keras.
- `params.py` : Hyperparameters. You can usefully modify PATCH_HOP_SECONDS.
- `features.py` : Audio feature extraction helpers.
- `inference.py` : Example code to classify input wav files.
- `yamnet_test.py` : Simple test of YAMNet installation

Input: Audio Features

See `features.py` .

As with our previous release VGGish, YAMNet was trained with audio features computed as follows:

- All audio is resampled to 16 kHz mono.
- A spectrogram is computed using magnitudes of the Short-Time Fourier Transform with a window size of 25 ms, a window hop of 10 ms, and a periodic Hann window.
- A mel spectrogram is computed by mapping the spectrogram to 64 mel bins covering the range 125-7500 Hz.
- A stabilized log mel spectrogram is computed by applying log(mel-spectrum + 0.001) where the offset is used to avoid taking a logarithm of zero.
- These features are then framed into 50%-overlapping examples of 0.96 seconds, where each example covers 64 mel bands and 96 frames of 10 ms each.

These 96x64 patches are then fed into the Mobilenet_v1 model to yield a 3x2 array of activations for 1024 kernels at the top of the convolution. These are averaged to give a 1024-dimension embedding, then put through a single logistic layer to get the 521 per-class output scores corresponding to the 960 ms input waveform segment. (Because of the window framing, you need at least 975 ms of input waveform to get the first frame of output scores.)

Class vocabulary

The file `yamnet_class_map.csv` describes the audio event classes associated with each of the 521 outputs of the network. Its format is:

```
index,mid,display_name
```

where `index` is the model output index (0..520), `mid` is the machine identifier for that class (e.g. `/m/09x0r`), and display_name is a human-readable description of the class (e.g. `Speech`).

The original Audioset data release had 527 classes. This model drops six of them on the recommendation of our Fairness reviewers to avoid potentially offensive mislabelings. We dropped the gendered versions (Male/Female) of Speech and Singing. We also dropped Battle cry and Funny music.

Performance

On the 20,366-segment AudioSet eval set, over the 521 included classes, the balanced average d-prime is 2.318, balanced mAP is 0.306, and the balanced average lwlrap is 0.393.

According to our calculations, the classifier has 3.7M weights and performs 69.2M multiplies for each 960ms input frame.

Contact information

This model repository is maintained by Manoj Plakal and Dan Ellis.