Task One: Classify

A Dataset and Taxonomy for Urban Sound Research

Justin Salamon[1,2], Christopher Jacoby[1], Juan Pablo Bello[1]
[1]Music and Audio Research Laboratory, New York University
[2]Center for Urban Science and Progress, New York University
{justin.salamon, cbj238, jpbello}@nyu.edu

ABSTRACT

Automatic urban sound classification is a growing area of research with applications in multimedia retrieval and urban informatics. In this paper we identify two main barriers to research in this area – the lack of a common taxonomy and the scarceness of large, real-world, annotated data. To address these issues we present a taxonomy of urban sounds and a new dataset, *UrbanSound*, containing 27 hours of audio with 18.5 hours of annotated sound event occurrences across 10 sound classes. The challenges presented by the new dataset are studied through a series of experiments using a baseline classification system.

Categories and Subject Descriptors

H.3.1 [**Information Systems**]: Content Analysis and Indexing; H.5.5 [**Information Systems**]: Sound and Music Computing

Keywords

Urban sound; dataset; taxonomy; classification

1. INTRODUCTION

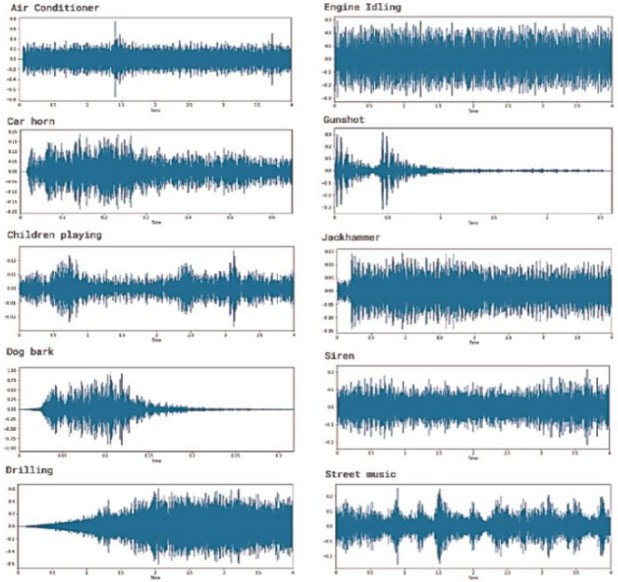
The automatic classification of environmental sound is a growing research field with multiple applications to large-scale, content-based multimedia indexing and retrieval [e.g. [13, 16, 10]]. In particular, the sonic analysis of urban environments is the subject of increased interest, partly enabled by multimedia sensor networks [18], as well as by large quantities of online multimedia content depicting urban scenes. However, while there is a large body of research in related areas such as speech, music and bioacoustics, work on the analysis of urban acoustics environments is relatively scarce. Furthermore, when existent, it mostly focuses on the classification of auditory scene type, e.g. street, park, as opposed to the identification of sound sources in those scenes, e.g. car horn, engine idling, bird tweet. See [5] for an example.

2. URBAN SOUND TAXONOMY

The taxonomical categorization of environmental sounds, a common first step in sound classification, has been extensively studied in the context of perceptual soundscape research [14]. Specific efforts to describe urban sounds have often been limited to subsets of broader taxonomies of acoustic environments [2], and thus only partially address the needs of systematic urban sound analysis. For an exhaustive review of previous work the reader is referred to [12].

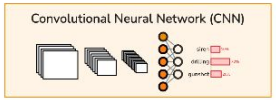
In our view, an urban sound taxonomy should satisfy the following three requirements: (1) it should factor in previous research and proposed taxonomies, (2) it should aim to be as detailed as possible, going down to low-level sound sources such as "car horn" (versus "transportation") and "jackhammer" (versus "construction"), (3) it should, in its first iteration, focus on sounds that are of specific relevance to urban sound research, such as sounds that contribute to urban noise pollution. To address (1), we decided to base our taxonomy on the subset of [2] dedicated to the urban acoustic environment. We define 4 top level groups: human, nature, mechanical and music, which are common to most previ-

Salamon et al. 2014

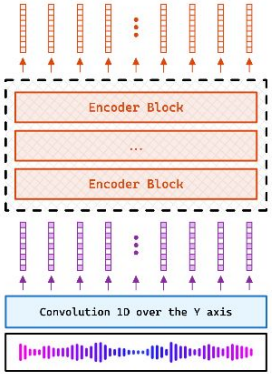


Clearly images! :)

Classify using a CNN



Classify using a Transformer



FOUNDERS AND CODERS

Convs

Machine Learning Institute

```python

class AudioEncoder(nn.Module):
    def __init__(self, n_mels: int, n_ctx: int, n_state: int, n_head: int, n_layer: int):
        super().__init__()
        self.conv1 = Conv1d(n_mels, n_state, kernel_size=3, padding=1)
        self.conv2 = Conv1d(n_state, n_state, kernel_size=3, stride=2, padding=1)
        self.register_buffer("positional_embedding", sinusoids(n_ctx, n_state))
        self.blocks = nn.ModuleList([ResidualAttentionBlock(n_state, n_head) for _ in range(n_layer)])
        self.ln_post = LayerNorm(n_state)

    def forward(self, x: Tensor):
        """
        x : torch.Tensor, shape = (batch_size, n_mels, n_ctx)
            the mel spectrogram of the audio
        """
        print('00:', x.shape) # torch.Size([1, 80, 3000])
        x = F.gelu(self.conv1(x))
        print('01:', x.shape) # torch.Size([1, 384, 3000])
        x = F.gelu(self.conv2(x))
        print('02:', x.shape) # torch.Size([1, 384, 1500])
        x = x.permute(0, 2, 1)
        print('03:', x.shape) # torch.Size([1, 1500, 384])
        assert x.shape[1:] == self.positional_embedding.shape, "incorrect audio shape"
        x = (x + self.positional_embedding).to(x.dtype)
        for block in self.blocks: x = block(x)
        x = self.ln_post(x)
        return x

```

https://github.com/openai/whisper/blob/main/whisper/model.py#L174

Language Prediction

```python
#
#
import whisper

#
#
model = whisper.load_model('tiny')
result = model.transcribe('./sample.wav')
print(result['text'])

```

```
(xld) root@Ubuntu-2404-noble-amd64-base ~/whisper # python workbench.py
100%|████████████████████████████████████████| 72.1M/72.1M [00:12<00:00, 5.91MiB/s]
00: torch.Size([1, 80, 3000])
01: torch.Size([1, 384, 3000])
02: torch.Size([1, 384, 1500])
03: torch.Size([1, 1500, 384])
00: torch.Size([1, 80, 3000])
01: torch.Size([1, 384, 3000])
02: torch.Size([1, 384, 1500])
03: torch.Size([1, 1500, 384])
 He is quite content to die.
```