

```
# PREP: options for preparing dataset
[PREP]
# dataset_type: corresponds to the model family such as "frame
classification" or "parametric umap"
dataset_type = "frame classification"
# input_type: input to model, either audio ("audio") or
spectrogram ("spect")
input_type = "audio"
# data_dir: directory with data to use when preparing dataset
data_dir = "/User/Library/CloudStorage/OneDrive/Tweetynet/
HAV_TN_Training/training_data"
# output_dir: directory where dataset will be created (as a sub-
directory within output_dir)
output_dir = "/User/Library/CloudStorage/OneDrive/Tweetynet/
HAV_TN_Training/train"
# audio_format: format of audio, either wav or cbin
audio_format = "wav"
# annot_format: format of annotations
annot_format = "raven"
# labelset: string or array with unique set of labels used in
annotations
labelset = "['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j',
'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's', 't', 'u', 'v', 'w',
'x', 'y', 'z', 'a1', 'b1', 'c1', 'd1', 'e1', 'f1', 'g1', 'h1',
'i1', 'j1', 'k1', 'l1', 'm1', 'n1', 'o1', 'p1', 'q1', 'r1',
's1', 't1', 'u1', 'v1', 'w1', 'x1', 'y1', 'z1', 'a2', 'b2',
'c2', 'd2', 'e2', 'f2', 'g2', 'h2', 'i2', 'j2', 'k2', 'l2',
'm2', 'n2', 'o2', 'p2', 'q2', 'r2', 's2', 't2', 'u2', 'z3',
'bz1', 'bz2', 'bz3']"
# train_dur: duration of training split in dataset, in seconds
train_dur = 50
# val_dur: duration of validation split in dataset, in seconds
val_dur = 15
# test_dur: duration of test split in dataset, in seconds
test_dur = 30

# SPECT_PARAMS: parameters for computing spectrograms
[SPECT_PARAMS]
# fft_size: size of window used for Fast Fourier Transform, in
number of samples
fft_size = 512
# step_size: size of step to take when computing spectra with
FFT for spectrogram
# also known as hop size
step_size = 64

# TRAIN: options for training model
```

```
[TRAIN]
# model: the string name of the model. must be a name within
`vak.models` or added e.g. with `vak.model.decorators.model`
model = "TweetyNet"
# root_results_dir: directory where results should be saved, as
a sub-directory within `root_results_dir`
root_results_dir = "/PATH/TO/DATA/vak/results/train"
# batch_size: number of samples from dataset per batch fed into
network
batch_size = 8
# num_epochs: number of training epochs, where an epoch is one
iteration through all samples in training split
num_epochs = 2
# normalize_spectrograms: if true, normalize spectrograms per
frequency bin, so mean of each is 0.0 and std is 1.0
# across the entire training split
normalize_spectrograms = true
# val_step: step number on which to compute metrics with
validation set, every time step % val_step == 0
# (a step is one batch fed through the network)
# saves a checkpoint if the monitored evaluation metric improves
(which is model specific)
val_step = 400
# ckpt_step: step number on which to save a checkpoint (as a
backup, regardless of validation metrics)
ckpt_step = 200
# patience: number of validation steps to wait before stopping
training early
# if the monitored evaluation metrics does not improve after
`patience` validation steps,
# then we stop training
patience = 4
# num_workers: number of workers to use when loading data with
multiprocessing
num_workers = 4
# device: name of device to run model on, one of "cuda", "cpu"
device = "cuda"
# dataset_path : path to dataset created by prep. This will be
added when you run `vak prep`, you don't have to add it

# train_dataset_params: parameters used when loading training
dataset
# for a frame classification model, we use a WindowDataset with
a specific `window_size`
[TRAIN.train_dataset_params]
window_size = 176
```

```
# val_transform_params: parameters used when transforming
validation data
# for a frame classification model, we use FrameDataset with the
eval_item_transform,
# that reshapes batches into consecutive adjacent windows with a
specific `window_size`
[TRAIN.val_transform_params]
window_size = 176

# TweetyNet.optimizer: we specify options for the model's
optimizer in this table
[TweetyNet.optimizer]
# lr: the learning rate
lr = 0.001

# TweetyNet.network: we specify options for the model's network
in this table
[TweetyNet.network]
# hidden_size: the number of elements in the hidden state in the
recurrent layer of the network
hidden_size = 256
```