# universal_audio_preprocessing

January 10, 2022

## 1 AUGMENTED RAVDESS Audio preprocessing

```
[1]: pip install ipynb
```

```
Defaulting to user installation because normal site-packages is not writeable
Looking in indexes: https://pypi.org/simple, https://pypi.ngc.nvidia.com
Requirement already satisfied: ipynb in
/home/19126778/.local/lib/python3.9/site-packages (0.5.1)
Note: you may need to restart the kernel to use updated packages.
```

```
[6]: import librosa as lb
     import numpy as np
     import matplotlib.pyplot as plt
     import os, glob
     import json
     import time
     import pandas as pd

     from multiprocessing import Pool
     from joblib import Parallel, delayed

     import sys
     sys.path.insert(0, './Augmenter')
     from ipynb.fs.full.augmentation import Augmenter
```

```
---------------------------------------------------------------------------
ModuleNotFoundError                       Traceback (most recent call last)
/tmp/ipykernel_42280/1096819287.py in <module>
     12 import sys
     13 sys.path.insert(0, './Augmenter')
---> 14 from ipynb.fs.full.augmentation import Augmenter

ModuleNotFoundError: No module named 'ipynb.fs.full.augmentation'
```

```
[ ]: TRAIN_RAVDESS_AUDIO_FILES_PATH = "../../data/Train/OriginalData/RAVDESS/*.wav"
     TEST_RAVDESS_AUDIO_FILES_PATH = "../../data/Test/OriginalData/RAVDESS/*.wav"
```

```
VALIDATE_RAVDESS_AUDIO_FILES_PATH = "../../data/Validate/OriginalData/RAVDESS/*.
 ↪wav"

TRAIN_CREMA_D_AUDIO_FILES_PATH = "../../data/Train/OriginalData/CREMA-D/
 ↪AudioWAV/*.wav"
TEST_CREMA_D_AUDIO_FILES_PATH = "../../data/Test/OriginalData/CREMA-D/AudioWAV/
 ↪*.wav"
VALIDATE_CREMA_D_AUDIO_FILES_PATH = "../../data/Validate/OriginalData/CREMA-D/
 ↪AudioWAV/*.wav"

PROCESSED_DATA = {
    'features': [],
    'emotions': []
}

focused_emotion_labels = ['neutral', 'happy', 'sad', 'angry', 'fearful',␣
 ↪'disgust']

# CREMA-D Female samples
CREMA_D_female_samples =␣
 ↪[1002,1003,1004,1006,1007,1008,1009,1010,1012,1013,1018,1020,1021,1024,1025,1028,1029,1030,
         ␣
 ↪1052,1053,1054,1055,1056,1058,1060,1061,1063,1072,1073,1074,1075,1076,1078,1079,1082,1084,1

# CREMA-D
CREMA_D_emotion_labels = {
  'NEU':'neutral',
  'HAP':'happy',
  'SAD':'sad',
  'ANG':'angry',
  'FEA':'fearful',
  'DIS':'disgust'
}

# RAVDESS
RAVDESS_emotion_labels = {
  '01':'"neutral"',
  '02':'calm',
  '03':'happy',
  '04':'sad',
  '05':'angry',
  '06':'fearful',
  '07':'disgust',
  '08':'surprised'
}
```

[ ]:

```python
def load_data_in_pos_neg(path, dataset_name:str):
    data = []
    # https://www.paulekman.com/universal-emotions/what-is-surprise/
    positive = ["neutral", "happy", "calm", "suprised"]
    negative = ["sad", "angry", "fearful", "disgust", "suprised"]

    for i, file in enumerate(glob.glob(path)):
        file_path = os.path.basename(file)

        emotion = ''
        if dataset_name == "ravdess":
            emotion = RAVDESS_emotion_labels[file_path.split("-")[2]]
            #emotion = RAVDESS_emotion_labels[file_path.split("-")[3]] #turn on␣
 ↪for trimmed data
        else:
            emotion = CREMA_D_emotion_labels[file_path.split("_")[2]]


        if emotion not in focused_emotion_labels:
            continue

        if emotion in positve:
            label = "positive"
        else:
            label = "negative"

        data.append([file, label])

    end_time = time.perf_counter()

    return pd.DataFrame(data, columns=["file", "emotion"])
```

[ ]:

```python
def load_files_in_df(path, dataset_name:str):
    data = []
    start_time = time.perf_counter()

    for i, file in enumerate(glob.glob(path)):
        file_path = os.path.basename(file)

        emotion = ''
        if dataset_name == "ravdess":
            emotion = RAVDESS_emotion_labels[file_path.split("-")[2]]
            #emotion = RAVDESS_emotion_labels[file_path.split("-")[3]] #turn on␣
 ↪for trimmed data
        else:
            emotion = CREMA_D_emotion_labels[file_path.split("_")[2]]
        if emotion not in focused_emotion_labels:
            continue

        data.append([file, emotion])

    end_time = time.perf_counter()

    return pd.DataFrame(data, columns=["file", "emotion"])
```

[ ]:

```
def load_sex_splitted_files(path:str, dataset_name:str):
    """
        Splits the male and female data into seperate datasets
    """
    female_data = []
    male_data = []

    for i, file in enumerate(glob.glob(path)):
        file_path = os.path.basename(file)

        if dataset_name == "ravdess":
            # Splits RAVDESS sex samples
            parts = file_path.replace('.','-').split("-")
            #emotion = RAVDESS_emotion_labels[parts[2]]
            emotion = RAVDESS_emotion_labels[parts[3]] #turn on for trimmed data

            if emotion not in focused_emotion_labels:
                continue

            if int(parts[7])%2 == 0: #turn on for trimmed data
            #if int(parts[6])%2 == 0:
                # Female sample
                female_data.append([file, emotion])
            else:
                # Male sample
                male_data.append([file, emotion])
        else:
            # Splits CREMA-D sex samples
            parts = file_path.replace('.','_').split("_")
            emotion = CREMA_D_emotion_labels[parts[2]]

            if emotion not in focused_emotion_labels:
                continue

            if int(parts[0].split('-')[1]) in CREMA_D_female_samples: # turn on␣
 ↪for trimmed data
            #if int(parts[0]) in CREMA_D_female_samples:
                # Female sample
                female_data.append([file, emotion])
            else:
                # Male sample
                male_data.append([file, emotion])


    female_df = pd.DataFrame(female_data, columns=["file", "emotion"])
    male_df = pd.DataFrame(male_data, columns=["file", "emotion"])

    return female_df, male_df
```

```
[ ]: def augment_audio(row, is_augmented: bool = False):
         audio, sr = lb.load(row["file"], sr=22050)

         extracted_features = []
         augmented_audios = [audio]

         if is_augmented:
             # Change pitch down
             augmented_audios.append(Augmenter.change_pitch(audio=audio, sr=sr))

             # Change pitch up
             augmented_audios.append(Augmenter.change_pitch(audio=audio, sr=sr,␣
      ↪pitch_type="up"))

             # Change speed slow
             augmented_audios.append(Augmenter.change_speed(audio=audio))

             # Change speed fast
             augmented_audios.append(Augmenter.change_speed(audio=audio,␣
      ↪speed_change="high"))

             # Change speed & pitch down
             augmented_audios.append(Augmenter.change_speed_and_pitch(audio=audio,␣
      ↪sr=sr))

             # Change speed & pitch up
             augmented_audios.append(Augmenter.change_speed_and_pitch(audio=audio,␣
      ↪sr=sr, pitch_type="up"))

             # Add distribution noise
             augmented_audios.append(Augmenter.add_distribution_noise(audio=audio))


         for a in augmented_audios:
             extracted_features.append(extract_feature(a, sr, row["emotion"]))
```

```
    return extracted_features
```

[ ]:
```
def extract_feature(audio, sr, emotion, mfcc=True, chroma=True, mel=True):
    result = np.array([])

    if mfcc:
        # Gets the mean of the MFCC
        # Change to 21 bins instead of 40
        mfccs = np.mean(lb.feature.mfcc(y=audio, sr=sr, n_mfcc=40).T, axis=0)
        result = np.hstack((result, mfccs))
    if chroma:
        # Gets the mean of the chromagram
        stft = np.abs(lb.stft(audio))
        chroma = np.mean(lb.feature.chroma_stft(S=stft, sr=sr).T, axis=0)
        result = np.hstack((result, chroma))
    if mel:
        # Gets the mean of the Mel-frequency spectrogram
        mel = np.mean(lb.feature.melspectrogram(audio, sr=sr).T, axis=0)
        result = np.hstack((result, mel))

    return (result.tolist(), emotion)
```

[ ]:
```
def create_json(df, dataset_type:str, dataset_name:str, is_augmented:bool, sex:
 ↪str = "", augmentation_type:str=""):
    start_time = time.perf_counter()
    n_proc = 35
    data = Parallel(n_jobs=n_proc)(delayed(augment_audio)(row, is_augmented)␣
 ↪for _, row in df.iterrows())

    formatted_data = {
        "features": [],
        "emotions": []
    }

    for d in data:
        for audio in d:
            formatted_data["features"].append(audio[0])
            formatted_data["emotions"].append(audio[1])

    is_sex = "" if not sex else f"{sex}_"
    is_agumented = "augmented_" if is_augmented else ""

    json_path = f'../data/{dataset_type}/{dataset_name}/
 ↪{dataset_type}_{is_sex}{is_agumented}{augmentation_type}_{dataset_name}_preprocessed_data.
 ↪json'
    print(json_path)
    with open(json_path, 'w') as outfile:
        json.dump(formatted_data, outfile, indent=4)

    end_time = time.perf_counter()
    print(f"{dataset_type} set | DONE PROCESSING DATA!!! | Duration: {end_time␣
↪- start_time:04f}")

    print(f'Size: {len(formatted_data["features"])}\n')
```

```
[ ]: def store_data(datasets, dataset_name:str, is_augmented:bool = False, split_sex:
↪bool = False, augmentation_type:str="", split_pos_neg=False):

    for key in datasets:
        if split_sex:
            print("Sex splitted")
            female_df, male_df = load_sex_splitted_files(datasets[key],␣
↪dataset_name)

            # Create female JSON file
            #create_json(df=female_df, dataset_type=key,␣
↪dataset_name=dataset_name, is_augmented=is_augmented,␣
↪sex='female',augmentation_type=augmentation_type)

            # Create male JSON file
            #create_json(df=male_df, dataset_type=key,␣
↪dataset_name=dataset_name, is_augmented=is_augmented,␣
↪sex='male',augmentation_type=augmentation_type)
        if split_pos_neg:
            print("Emotions classified")
            pos_df, neg_df = load_data_in_pos_neg(datasets[key], dataset_name)
            # maak pos neg df met load
        else:
            print("Default")
            df = load_files_in_df(datasets[key], dataset_name)

            #create_json(df=df, dataset_type=key,␣
↪dataset_name=dataset_name,is_augmented=is_augmented,␣
↪augmentation_type=augmentation_type)
```

## 1.1 Combined preprocessing

```
[ ]: def store_data_combined(datasets, is_augmented:bool = False, split_sex:bool =␣
↪False, augmentation_type:str=""):
    datasets = datasets
    dataset_name='ravdess'
    df_train_r = load_files_in_df(datasets['train-r'], dataset_name)
    df_test_r = load_files_in_df(datasets['test-r'], dataset_name)
    df_valid_r = load_files_in_df(datasets['validate-r'], dataset_name)
    print(len(df_train_r))
    print(len(df_test_r))
    print(len(df_valid_r))
    print()

    dataset_name='crema-d'
    df_train_c = load_files_in_df(datasets['train-c'], dataset_name)
    df_test_c = load_files_in_df(datasets['test-c'], dataset_name)
    df_valid_c = load_files_in_df(datasets['validate-c'], dataset_name)
    print(len(df_train_c))
    print(len(df_test_c))
    print(len(df_valid_c))
    print()

    df_train = pd.concat([df_train_r,df_train_c])
    df_test = pd.concat([df_test_r,df_test_c])
    df_valid = pd.concat([df_valid_r,df_valid_c])
    print(len(df_train))
    print(len(df_test))
    print(len(df_valid))
    print()

    create_json(df_train, 'train','combined',␣
 ↪is_augmented,split_sex,augmentation_type)
    create_json(df_test, 'test','combined',␣
 ↪is_augmented,split_sex,augmentation_type)
    create_json(df_test, 'validate','combined',␣
 ↪is_augmented,split_sex,augmentation_type)
```

[ ]:

```
datasets = {
        'train-r': TRAIN_RAVDESS_AUDIO_FILES_PATH,
        'test-r': TEST_RAVDESS_AUDIO_FILES_PATH,
        'validate-r': VALIDATE_RAVDESS_AUDIO_FILES_PATH,
        'train-c': TRAIN_CREMA_D_AUDIO_FILES_PATH,
        'test-c': TEST_CREMA_D_AUDIO_FILES_PATH,
        'validate-c': VALIDATE_CREMA_D_AUDIO_FILES_PATH,
    }
datasets_trimmed = {
        'train-r': "../../data/Train/Trimmed/RAVDESS/*.wav",
        'test-r': "../../data/Test/Trimmed/RAVDESS/*.wav",
        'validate-r': "../../data/Validate/Trimmed/RAVDESS/*.wav",
        'train-c': "../../data/Train/Trimmed/CREMA-D/AudioWAV/*.wav",
        'test-c': "../../data/Test/Trimmed/CREMA-D/AudioWAV/*.wav",
        'validate-c': "../../data/Validate/Trimmed/CREMA-D/AudioWAV/*.wav",
    }

store_data_combined(datasets=datasets_trimmed,
          is_augmented=False,
          split_sex=False,
          augmentation_type='Trimmed')
```

### 1.1.1 RAVDESS preprocessing

### 1.1.2 Original data - Ravdess

[ ]:

```
datasets = {
    'train': TRAIN_RAVDESS_AUDIO_FILES_PATH,
    'test': TEST_RAVDESS_AUDIO_FILES_PATH,
    'validate': VALIDATE_RAVDESS_AUDIO_FILES_PATH,
}

store_data(datasets=datasets,
           dataset_name='ravdess',
           is_augmented=False,
          augmentation_type='Original')
```

### 1.1.3 Original data - Crema-d

[ ]:

```
datasets = {
    'train': TRAIN_CREMA_D_AUDIO_FILES_PATH,
    'test': TEST_CREMA_D_AUDIO_FILES_PATH,
    'validate': VALIDATE_CREMA_D_AUDIO_FILES_PATH,
}

store_data(datasets=datasets,
           dataset_name='crema-d',
           is_augmented=False,
          augmentation_type='Original')
```

### 1.1.4 Trimmed data - Ravdess

**(turn on commented line in load_files_in_df())**

[ ]:

```
datasets = {
    'train': "../../data/Train/Trimmed/RAVDESS/*.wav",
    'test': "../../data/Test/Trimmed/RAVDESS/*.wav",
    'validate': "../../data/Validate/Trimmed/RAVDESS/*.wav",
}

store_data(datasets=datasets,
           dataset_name='ravdess',
           is_augmented=False,
          augmentation_type='Trimmed')
```

### 1.1.5 Trimmed data - Crema-d

**(turn on commented line in load_files_in_df())**

[ ]:

```
datasets = {
    'train': "../../data/Train/Trimmed/CREMA-D/AudioWAV/*.wav",
    'test': "../../data/Test/Trimmed/CREMA-D/AudioWAV/*.wav",
    'validate': "../../data/Validate/Trimmed/CREMA-D/AudioWAV/*.wav",
}

store_data(datasets=datasets,
           dataset_name='crema-d',
           is_augmented=False,
          augmentation_type='Trimmed')
```

### 1.1.6 Augmented - pitch up - RAVDESS

[ ]:

```
datasets = {
    'train': TRAIN_RAVDESS_AUDIO_FILES_PATH,
    'test': TEST_RAVDESS_AUDIO_FILES_PATH,
    'validate': VALIDATE_RAVDESS_AUDIO_FILES_PATH,
}

store_data(datasets=datasets,
           dataset_name='ravdess',
           is_augmented=True,
          augmentation_type='PitchUp')
```

### 1.1.7 Augmented - pitch up - CREMA-D

[ ]:

```
datasets = {
    'train': TRAIN_CREMA_D_AUDIO_FILES_PATH,
    'test': TEST_CREMA_D_AUDIO_FILES_PATH,
    'validate': VALIDATE_CREMA_D_AUDIO_FILES_PATH,
}

store_data(datasets=datasets,
           dataset_name='crema-d',
           is_augmented=True,
          augmentation_type='PitchUp')
```

### 1.1.8 Augmented - pitch down - RAVDESS

[ ]:

```
datasets = {
    'train': TRAIN_RAVDESS_AUDIO_FILES_PATH,
    'test': TEST_RAVDESS_AUDIO_FILES_PATH,
    'validate': VALIDATE_RAVDESS_AUDIO_FILES_PATH,
}

store_data(datasets=datasets,
           dataset_name='ravdess',
           is_augmented=True,
          augmentation_type='PitchDown')
```

### 1.1.9 Augmented - pitch down - CREMA-D

[ ]:

```
datasets = {
    'train': TRAIN_CREMA_D_AUDIO_FILES_PATH,
    'test': TEST_CREMA_D_AUDIO_FILES_PATH,
    'validate': VALIDATE_CREMA_D_AUDIO_FILES_PATH,
}

store_data(datasets=datasets,
           dataset_name='crema-d',
           is_augmented=True,
          augmentation_type='PitchDown')
```

### 1.1.10 Augmented - lower speed - RAVDESS

[ ]:

```
datasets = {
    'train': TRAIN_RAVDESS_AUDIO_FILES_PATH,
    'test': TEST_RAVDESS_AUDIO_FILES_PATH,
    'validate': VALIDATE_RAVDESS_AUDIO_FILES_PATH,
}

store_data(datasets=datasets,
           dataset_name='ravdess',
           is_augmented=True,
          augmentation_type='LowerSpeed')
```

### 1.1.11 Augmented - lower speed - CREMA-D

[ ]:

```
datasets = {
    'train': TRAIN_CREMA_D_AUDIO_FILES_PATH,
    'test': TEST_CREMA_D_AUDIO_FILES_PATH,
    'validate': VALIDATE_CREMA_D_AUDIO_FILES_PATH,
}

store_data(datasets=datasets,
           dataset_name='crema-d',
           is_augmented=True,
          augmentation_type='LowerSpeed')
```

### 1.1.12 Augmented - higher speed - RAVDESS

[ ]:

```
datasets = {
    'train': TRAIN_RAVDESS_AUDIO_FILES_PATH,
    'test': TEST_RAVDESS_AUDIO_FILES_PATH,
    'validate': VALIDATE_RAVDESS_AUDIO_FILES_PATH,
}

store_data(datasets=datasets,
           dataset_name='ravdess',
           is_augmented=True,
          augmentation_type='HigherSpeed')
```

### 1.1.13 Augmented - higher speed - CREMA-D

[ ]:

```
datasets = {
    'train': TRAIN_CREMA_D_AUDIO_FILES_PATH,
    'test': TEST_CREMA_D_AUDIO_FILES_PATH,
    'validate': VALIDATE_CREMA_D_AUDIO_FILES_PATH,
}

store_data(datasets=datasets,
           dataset_name='crema-d',
           is_augmented=True,
          augmentation_type='HigherSpeed')
```

### 1.1.14 Augmented - distribution noise - RAVDESS

[ ]:

```
datasets = {
    'train': TRAIN_RAVDESS_AUDIO_FILES_PATH,
    'test': TEST_RAVDESS_AUDIO_FILES_PATH,
    'validate': VALIDATE_RAVDESS_AUDIO_FILES_PATH,
}

store_data(datasets=datasets,
           dataset_name='ravdess',
           is_augmented=True,
          augmentation_type='DistributionNoise')
```

### 1.1.15 Augmented - distribution noise - CREMA-D

[ ]:

```
datasets = {
    'train': TRAIN_CREMA_D_AUDIO_FILES_PATH,
    'test': TEST_CREMA_D_AUDIO_FILES_PATH,
    'validate': VALIDATE_CREMA_D_AUDIO_FILES_PATH,
}

store_data(datasets=datasets,
           dataset_name='crema-d',
           is_augmented=True,
          augmentation_type='DistributionNoise')
```

### 1.1.16 Augmented - speed & pitch up - RAVDESS

[ ]:

```
datasets = {
    'train': TRAIN_RAVDESS_AUDIO_FILES_PATH,
    'test': TEST_RAVDESS_AUDIO_FILES_PATH,
    'validate': VALIDATE_RAVDESS_AUDIO_FILES_PATH,
}

store_data(datasets=datasets,
           dataset_name='ravdess',
           is_augmented=True,
          augmentation_type='SpeedAndPitchUp')
```

### 1.1.17 Augmented - speed & pitch up- CREMA-D

[ ]:

```
datasets = {
    'train': TRAIN_CREMA_D_AUDIO_FILES_PATH,
    'test': TEST_CREMA_D_AUDIO_FILES_PATH,
    'validate': VALIDATE_CREMA_D_AUDIO_FILES_PATH,
}

store_data(datasets=datasets,
           dataset_name='crema-d',
           is_augmented=True,
          augmentation_type='SpeedAndPitchUp')
```

### 1.1.18 Augmented - speed & pitch down - RAVDESS

[ ]:

```
datasets = {
    'train': TRAIN_RAVDESS_AUDIO_FILES_PATH,
    'test': TEST_RAVDESS_AUDIO_FILES_PATH,
    'validate': VALIDATE_RAVDESS_AUDIO_FILES_PATH,
}

store_data(datasets=datasets,
           dataset_name='ravdess',
           is_augmented=True,
          augmentation_type='SpeedAndPitchDown')
```

### 1.1.19 Augmented - speed & pitch down- CREMA-D

[ ]:

```
datasets = {
    'train': TRAIN_CREMA_D_AUDIO_FILES_PATH,
    'test': TEST_CREMA_D_AUDIO_FILES_PATH,
    'validate': VALIDATE_CREMA_D_AUDIO_FILES_PATH,
}

store_data(datasets=datasets,
           dataset_name='crema-d',
           is_augmented=True,
          augmentation_type='SpeedAndPitchDown')
```

### 1.1.20 Augmented - all - RAVDESS

[ ]:

```
datasets = {
    'train': TRAIN_RAVDESS_AUDIO_FILES_PATH,
    'test': TEST_RAVDESS_AUDIO_FILES_PATH,
    'validate': VALIDATE_RAVDESS_AUDIO_FILES_PATH,
}

store_data(datasets=datasets,
           dataset_name='ravdess',
           is_augmented=True,
          augmentation_type='AllAugmentations')
```

### 1.1.21 Augmented - all - CREMA-D

[ ]:

```
datasets = {
    'train': TRAIN_CREMA_D_AUDIO_FILES_PATH,
    'test': TEST_CREMA_D_AUDIO_FILES_PATH,
    'validate': VALIDATE_CREMA_D_AUDIO_FILES_PATH,
}

store_data(datasets=datasets,
           dataset_name='crema-d',
           is_augmented=True,
          augmentation_type='AllAugmentations')
```

### 1.1.22 Augmented - all + trimmed - RAVDESS

**(turn on commented line in load_files_in_df())**

[ ]:

```
datasets = {
    'train': "../../data/Train/Trimmed/RAVDESS/*.wav",
    'test': "../../data/Test/Trimmed/RAVDESS/*.wav",
    'validate': "../../data/Validate/Trimmed/RAVDESS/*.wav",
}

store_data(datasets=datasets,
           dataset_name='ravdess',
           is_augmented=True,
          augmentation_type='AllAugmentationsTrimmed')
```

### 1.1.23 Augmented - all + trimmed - CREMA-D

**(turn on commented line in load_files_in_df())**

[ ]:

```
datasets = {
    'train': "../../data/Train/Trimmed/CREMA-D/AudioWAV/*.wav",
    'test': "../../data/Test/Trimmed/CREMA-D/AudioWAV/*.wav",
    'validate': "../../data/Validate/Trimmed/CREMA-D/AudioWAV/*.wav",
}

store_data(datasets=datasets,
           dataset_name='crema-d',
           is_augmented=True,
          augmentation_type='AllAugmentationsTrimmed')
```

### 1.1.24 Split Gender No Aug Original - RAVDESS

[ ]:

```
datasets = {
    'train': TRAIN_RAVDESS_AUDIO_FILES_PATH,
    'test': TEST_RAVDESS_AUDIO_FILES_PATH,
    'validate': VALIDATE_RAVDESS_AUDIO_FILES_PATH,
}

store_data(datasets=datasets,
           dataset_name='ravdess',
           is_augmented=False,
           split_sex=True,
          augmentation_type='SplitGenderNoAugOriginal')
```

### 1.1.25 Split Gender No Aug Original - CREMA-D

[ ]:

```
datasets = {
    'train': TRAIN_CREMA_D_AUDIO_FILES_PATH,
    'test': TEST_CREMA_D_AUDIO_FILES_PATH,
    'validate': VALIDATE_CREMA_D_AUDIO_FILES_PATH,
}

store_data(datasets=datasets,
           dataset_name='crema-d',
           is_augmented=False,
           split_sex=True,
          augmentation_type='SplitGenderNoAugOriginal')
```

### 1.1.26 Split Gender No Aug Trimmed - RAVDESS

**(turn on commented line in load_sex_splitted_files())**

[ ]:

```
datasets = {
    'train': "../../data/Train/Trimmed/RAVDESS/*.wav",
    'test': "../../data/Test/Trimmed/RAVDESS/*.wav",
    'validate': "../../data/Validate/Trimmed/RAVDESS/*.wav",
}
store_data(datasets=datasets,
           dataset_name='ravdess',
           is_augmented=False,
           split_sex=True,
          augmentation_type='SplitGenderNoAugTrimmed')
```

### 1.1.27 Split Gender No Aug Trimmed - CREMA-D

#### (turn on commented line in load_sex_splitted_files())

[ ]:

```
datasets = {
    'train': "../../data/Train/Trimmed/CREMA-D/AudioWAV/*.wav",
    'test': "../../data/Test/Trimmed/CREMA-D/AudioWAV/*.wav",
    'validate': "../../data/Validate/Trimmed/CREMA-D/AudioWAV/*.wav",
}

store_data(datasets=datasets,
           dataset_name='crema-d',
           is_augmented=False,
           split_sex=True,
          augmentation_type='SplitGenderNoAugTrimmed')
```

[ ]:

```

```