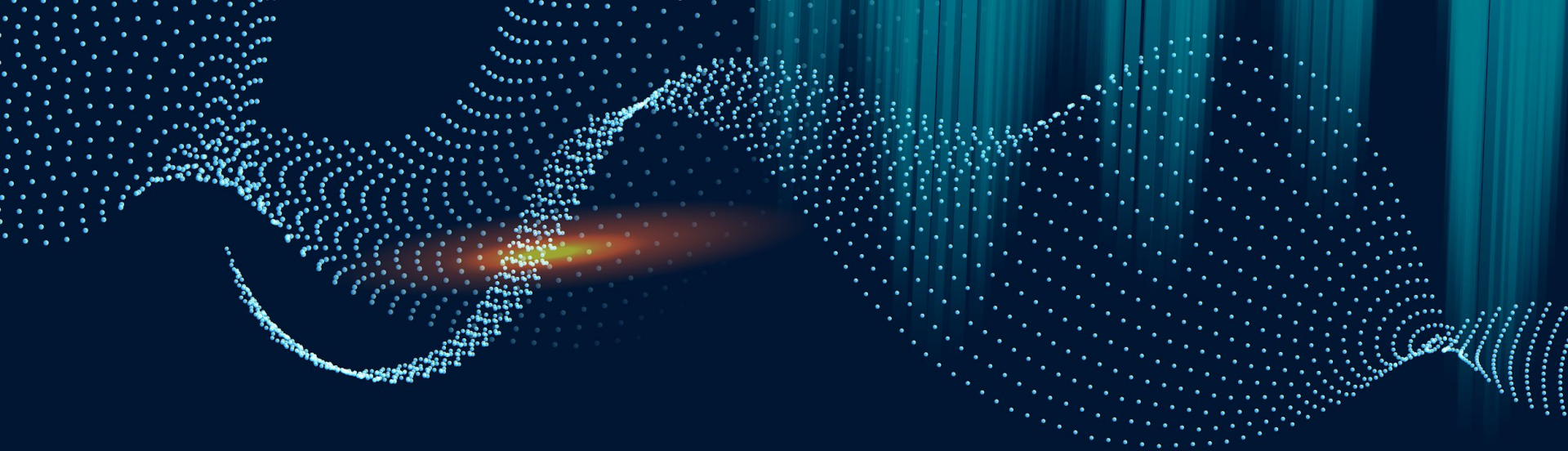# MUSIC RESAMPLER

Team 03

Yesin Soufi

Osman Kaplan

Jakob Schaal

Niklas Öxle

Elena Müller

Sascha Lehmann

01

Challenge 1

Create Samples

Create Samples

- Es wurden Songs ohne Gesang genutzt
    - Ca. 30 Songs aus einem Genre
    - (https://drive.google.com/drive/folders/1y46
      vfqsB-ZrLwG3UFxaXqDNHz14qxKXh)

- Die Songs wurden zunächst in gleich lange
  Samples geschnitten

- Darauf aufbauend wurden verschiedene
  Sample Trainings Datensätze erzeugt

Create Samples (3 Python files)

createSamples.py

createDataSet.py

createDataSet.py

https://bit.ly/3KSxwCF

https://bit.ly/3sKHtMz

02 | # Challenge 2

Rebuild song from samples

Convolutional Neural Network

buildModel.py

```python
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv1D, MaxPooling1D, Dense,Dropout,Activation,Flatten
from tensorflow.keras.optimizers import Adam

def firstModel(num_labels):
    model=Sequential()
    ###first layer
    model.add(Dense(100,input_shape=(176400,)))
    model.add(Activation('relu'))
    model.add(Dropout(0.5))
    ###second layer
    model.add(Dense(200))
    model.add(Activation('relu'))
    model.add(Dropout(0.5))
    ###third layer
    model.add(Dense(100))
    model.add(Activation('relu'))
    model.add(Dropout(0.5))
    ###fourth layer
    model.add(Dense(200))
    model.add(Activation('relu'))
    model.add(Dropout(0.5))
    ###fifth layer
    model.add(Dense(100))
    model.add(Activation('relu'))
    model.add(Dropout(0.5))

    ###final layer
    model.add(Dense(num_labels))
    model.add(Activation('softmax'))

    model.compile(loss='categorical_crossentropy',metrics=['accuracy'],optimizer='adam')

    return model

def cnnModel():
    model = Sequential()
    model.add(Conv1D(64, 3, activation='relu', input_shape=(176400,1)))
    model.add(MaxPooling1D(2))
    model.add(Conv1D(64, 3, activation='relu'))
    model.add(MaxPooling1D(2))
    model.add(Flatten())
    model.add(Dense(64, activation='relu'))
    model.add(Dense(24, activation='softmax'))

    model.compile(Adam(lr=.0001), loss='categorical_crossentropy', metrics=['accuracy'])

    return model
```

https://bit.ly/3M38OkG

tensorflow_classification.py

```python
22  #%%
23  # Label Parameter
24  N_CLASSES = 25 # CHANGE HERE, total number of classes
25
26  #Files containing the path to images and the labels [path/to/images label]
27  train_file = 'data/csv/train_Samples.csv'
28  val_file = 'data/csv/vali_Samples.csv'
29
30  #Lists where to store the paths and labels will be stored
31  files = []
32  labels = []
33
34  df_test = pd.read_csv(train_file, index_col=0)
35  df_test
36  #%%
37  #load waveform from samples with filePath
38  extracted_waveform=[]
39  for index_num,row in df_test.iterrows():
40      file_name = row['filePath']
41      file_name = file_name.replace('Sascha', 'sasch')
42      class_label = row["label"]
43      data, sr = librosa.load(file_name, mono=True)
44      extracted_waveform.append([data,class_label])
45
46  extracted_waveform
47
48  #%%
49  #create df from extracted waveform and label
50  extracted_df=pd.DataFrame(extracted_waveform,columns=['waveform','class'])
51  extracted_df.head(10)
52  extracted_df = extracted_df.drop(extracted_df[extracted_df['waveform'].map(len) < 176400].index)
53
54  extracted_df
55
56  #%%
57  #split into waveform and label
58  X=np.array(extracted_df['waveform'].tolist(), dtype='float32')
59  #X=np.array(X).astype("float32")
60  y=np.array(extracted_df['class'].tolist())
61
```

https://bit.ly/3KZ4Msf

Vorgehensweise Challenge 2

- Setzen von Label Parametern
- Für die Klassifikation haben wir 2 csv Dateien verwendet (train_file und val_file)
- Im Trainingsdatensatz wurden die Samples mit einer nach der Reihenfolge gesetzten ID des Songs gelabelt
- Anschließend wurde ein neuer Song genommen der nicht im Trainingsdatensatz enthalten war und wird in 8 sek Stücke geschnitten und sollte für jedes Sample das zugehörige Label prognostizieren
- Im Letzten Schritt werden die Samples nach den prognostizierten Samples sortiert und exportiert

Ergebnisse

Original Song

Trained Song