Appendix

Import Lib

In [1]:

```
import os
from glob import glob
import librosa
import librosa.display
import numpy as np
import matplotlib.pyplot as plt
import ipyparallel as ipp
import time
from warnings import filterwarnings
filterwarnings('ignore')
```

Function to create a spectrogram from an audio file

In [2]:

```
def create_spectrogram(filename, name, store_path):
    plt.interactive(False)
    clip, sample_rate = librosa.load(filename, sr=None)
    fig = plt.figure(figsize=[0.72, 0.72])
    ax = fig.add_subplot(111)
    ax.axes.get_xaxis().set_visible(False)
    ax.axes.get_yaxis().set_visible(False)
    ax.set_frame_on(False)
    S = librosa.feature.melspectrogram(y=clip, sr=sample_rate)
    librosa.display.specshow(librosa.power_to_db(S, ref=np.max))
    filename = os.path.join(store_path, name + '.jpg')
    plt.savefig(filename, dpi=400, bbox_inches='tight', pad_inches=0)
    plt.close()
    fig.clf()
    plt.close(fig)
    plt.close('all')
    del filename, name, clip, sample_rate, fig, ax, S
```

Helper function to process a single audio file

In [3]:

```
def process_audio_file(file, folder_path):
    name = os.path.basename(file)
    create_spectrogram(file, name, folder_path)
```

Convert Audio files to spectrogram JPG files with ipyparallel

In [4]:

```
def Convert_Audio_File_to_jpg_file(filename, store_filepath, number_of_worker):
    """
    Convert audio files to spectrogram JPG files using ipyparallel.

    Parameters:
        filename (str): Path to the main directory containing audio files and number of worker.
    """
    # Set up ipyparallel cluster
    Execution_time = 0
    rc = ipp.Cluster(n=number_of_worker).start_and_connect_sync()  # Start and connect cluster with 4 workers
    dview = rc[:]  # Create a direct view to all workers

    dview.push({"create_spectrogram": create_spectrogram})
    dview.push({"process_audio_file": process_audio_file})
    dview.execute("import os")
    dview.execute("import librosa")
    dview.execute("import librosa.display")
    dview.execute("import numpy as np")
    dview.execute("import matplotlib.pyplot as plt")
    dview.execute("import time")

    start_time = time.time()
    # Make a dictionary for given class and their file path names
    file_list = glob(os.path.join(filename, "*"))
    file_dic = {}
    for file in file_list:
        all_files = []
        for root, dirs, files in os.walk(file):
            for file_ in files:
                # Join the root directory with the file name to get the full path
                all_files.append(os.path.join(root, file_))
        file_dic[file] = all_files

    # Create file directories to store the converted audio files into JPG
    file_path = []
    for folder in file_dic.keys():
        folder_rot = os.path.join(store_filepath, os.path.basename(folder))
        file_path.append(folder_rot)
        os.makedirs(folder_rot, exist_ok=True)

    # Delegate work to workers using dview.map_sync()
    for i, folder in enumerate(file_dic.keys()):
        music_files = file_dic[folder]
        folder_path = file_path[i]
        # Map the processing function across music files with ipyparallel
        dview.map_sync(lambda file: process_audio_file(file, folder_path), music_files)
    end_time = time.time()
    elapsed_time = end_time - start_time
    Execution_time = elapsed_time
    print(f"Execution time: {elapsed_time:.2f} seconds")
    # Shut down the cluster after processing
    rc.shutdown()

    return Execution_time
```

Convert Audio File to jpg file

In [5]:

```
Audio_Data_Path = r"C:\Users\nikhi\OneDrive\Desktop\Final Project\DATA\genres"
Spectro_jpg_Path = r"C:\Users\nikhi\OneDrive\Desktop\Final Project\DATA\Convert_Audio_File_to_jpg_file"
Convert_Audio_File_to_jpg_file(Audio_Data_Path,Spectro_jpg_Path ,number_of_worker = 4)
```

```
Starting 4 engines with <class 'ipyparallel.cluster.launcher.LocalEngineSetLauncher'>
  0%|          | 0/4 [00:00<?, ?engine/s]
Execution time: 43.02 seconds
```

Out[5]:

```
43.01757502555847
```

```
engine set stopped 1733365864: {'engines': {'2': {'exit_code': 0, 'pid': 20512, 'identifier': '2'}, '0': {'exit_code': 0, 'pid': 21332,
'identifier': '0'}, '3': {'exit_code': 0, 'pid': 10848, 'identifier': '3'}, '1': {'exit_code': 0, 'pid': 18488, 'identifier': '1'}}, 'e
xit_code': 0}
```

Experiment with Core

In [6]:

```
Execution_time = []
Cores = range(1,9)
for number_of_worker in Cores:
   Execution_time.append(Convert_Audio_File_to_jpg_file(Audio_Data_Path, Spectro_jpg_Path, number_of_worker = number_of_worker))
```

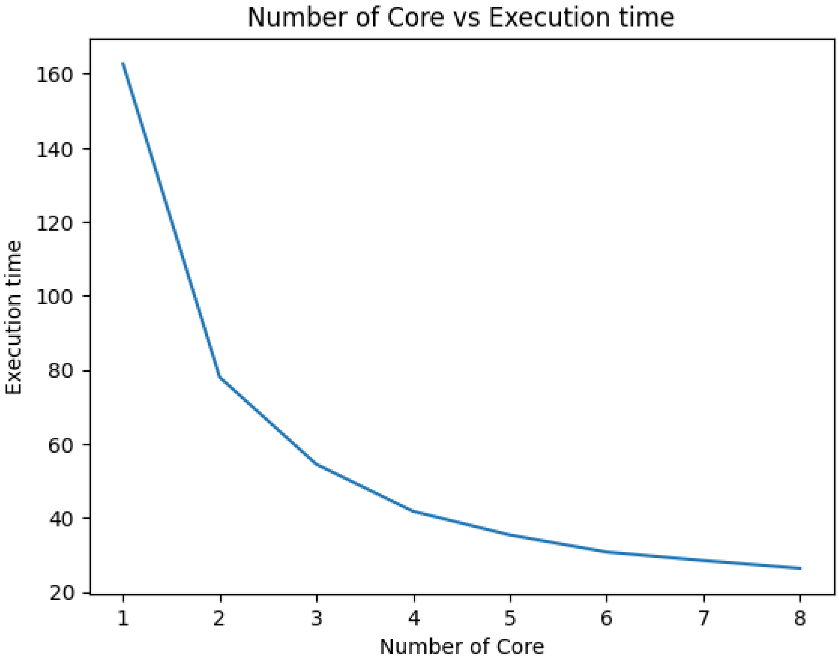
```
Starting 1 engines with <class 'ipyparallel.cluster.launcher.LocalEngineSetLauncher'>
  0%|          | 0/1 [00:00<?, ?engine/s]
Execution time: 162.66 seconds
engine set stopped 1733365915: {'engines': {'0': {'exit_code': 0, 'pid': 14416, 'identifier': '0'}}, 'exit_code': 0}
Starting 2 engines with <class 'ipyparallel.cluster.launcher.LocalEngineSetLauncher'>
  0%|          | 0/2 [00:00<?, ?engine/s]
Execution time: 78.02 seconds
Failed to remove C:\Users\nikhi\.ipython\profile_default\log\ipengine-1733366084-vstu-1733366085-1.log: [WinError 5] Access is denied:
'C:\\Users\\nikhi\\.ipython\\profile_default\\log\\ipengine-1733366084-vstu-1733366085-1.log'
engine set stopped 1733366085: {'engines': {'0': {'exit_code': 0, 'pid': 21460, 'identifier': '0'}, '1': {'exit_code': 0, 'pid': 27340,
'identifier': '1'}}, 'exit_code': 0}
Starting 3 engines with <class 'ipyparallel.cluster.launcher.LocalEngineSetLauncher'>
  0%|          | 0/3 [00:00<?, ?engine/s]
Execution time: 54.51 seconds
engine set stopped 1733366171: {'engines': {'2': {'exit_code': 0, 'pid': 25728, 'identifier': '2'}, '1': {'exit_code': 0, 'pid': 20336,
'identifier': '1'}, '0': {'exit_code': 0, 'pid': 25376, 'identifier': '0'}}, 'exit_code': 0}
Starting 4 engines with <class 'ipyparallel.cluster.launcher.LocalEngineSetLauncher'>
  0%|          | 0/4 [00:00<?, ?engine/s]
Execution time: 41.81 seconds
engine set stopped 1733366233: {'engines': {'0': {'exit_code': 0, 'pid': 18900, 'identifier': '0'}, '2': {'exit_code': 0, 'pid': 20572,
'identifier': '2'}, '3': {'exit_code': 0, 'pid': 22864, 'identifier': '3'}, '1': {'exit_code': 0, 'pid': 23844, 'identifier': '1'}}, 'e
xit_code': 0}
Starting 5 engines with <class 'ipyparallel.cluster.launcher.LocalEngineSetLauncher'>
  0%|          | 0/5 [00:00<?, ?engine/s]
```

```
Execution time: 35.43 seconds
engine set stopped 1733366282: {'engines': {'3': {'exit_code': 0, 'pid': 25924, 'identifier': '3'}, '0': {'exit_code': 0, 'pid': 24580,
'identifier': '0'}, '2': {'exit_code': 0, 'pid': 27620, 'identifier': '2'}, '1': {'exit_code': 0, 'pid': 21080, 'identifier': '1'},
'4': {'exit_code': 0, 'pid': 19184, 'identifier': '4'}}, 'exit_code': 0}
Starting 6 engines with <class 'ipyparallel.cluster.launcher.LocalEngineSetLauncher'>
  0%|          | 0/6 [00:00<?, ?engine/s]
Execution time: 30.82 seconds
engine set stopped 1733366326: {'engines': {'2': {'exit_code': 0, 'pid': 20140, 'identifier': '2'}, '0': {'exit_code': 0, 'pid': 23164,
'identifier': '0'}, '1': {'exit_code': 0, 'pid': 11856, 'identifier': '1'}, '5': {'exit_code': 0, 'pid': 26344, 'identifier': '5'},
'4': {'exit_code': 0, 'pid': 2152, 'identifier': '4'}, '3': {'exit_code': 0, 'pid': 27152, 'identifier': '3'}}, 'exit_code': 0}
Starting 7 engines with <class 'ipyparallel.cluster.launcher.LocalEngineSetLauncher'>
  0%|          | 0/7 [00:00<?, ?engine/s]
Execution time: 28.52 seconds
Failed to remove C:\Users\nikhi\.ipython\profile_default\log\ipengine-1733366363-hjat-1733366364-2.log: [WinError 2] The system cannot
find the file specified: 'C:\\Users\\nikhi\\.ipython\\profile_default\\log\\ipengine-1733366363-hjat-1733366364-2.log'
Failed to remove C:\Users\nikhi\.ipython\profile_default\log\ipengine-1733366363-hjat-1733366364-3.log: [WinError 32] The process canno
t access the file because it is being used by another process: 'C:\\Users\\nikhi\\.ipython\\profile_default\\log\\ipengine-1733366363-h
jat-1733366364-3.log'
engine set stopped 1733366364: {'engines': {'1': {'exit_code': 0, 'pid': 23124, 'identifier': '1'}, '2': {'exit_code': 0, 'pid': 23284,
'identifier': '2'}, '3': {'exit_code': 0, 'pid': 10412, 'identifier': '3'}, '4': {'exit_code': 0, 'pid': 11464, 'identifier': '4'},
'5': {'exit_code': 0, 'pid': 1996, 'identifier': '5'}, '0': {'exit_code': 0, 'pid': 13308, 'identifier': '0'}, '6': {'exit_code': 0, 'p
id': 26908, 'identifier': '6'}}, 'exit_code': 0}
Starting 8 engines with <class 'ipyparallel.cluster.launcher.LocalEngineSetLauncher'>
  0%|          | 0/8 [00:00<?, ?engine/s]
Execution time: 26.41 seconds
Failed to remove C:\Users\nikhi\.ipython\profile_default\log\ipengine-1733366400-7yh5-1733366401-1.log: [WinError 2] The system cannot
find the file specified: 'C:\\Users\\nikhi\\.ipython\\profile_default\\log\\ipengine-1733366400-7yh5-1733366401-1.log'
Failed to remove C:\Users\nikhi\.ipython\profile_default\log\ipengine-1733366400-7yh5-1733366401-6.log: [WinError 32] The process canno
t access the file because it is being used by another process: 'C:\\Users\\nikhi\\.ipython\\profile_default\\log\\ipengine-1733366400-7
yh5-1733366401-6.log'
engine set stopped 1733366401: {'engines': {'0': {'exit_code': 0, 'pid': 24628, 'identifier': '0'}, '1': {'exit_code': 0, 'pid': 14832,
'identifier': '1'}, '7': {'exit_code': 0, 'pid': 15012, 'identifier': '7'}, '5': {'exit_code': 0, 'pid': 27244, 'identifier': '5'},
'3': {'exit_code': 0, 'pid': 15744, 'identifier': '3'}, '6': {'exit_code': 0, 'pid': 11516, 'identifier': '6'}, '4': {'exit_code': 0,
'pid': 4368, 'identifier': '4'}, '2': {'exit_code': 0, 'pid': 18236, 'identifier': '2'}}, 'exit_code': 0}
```

Number of Core vs Execution time

In [7]:

```python
plt.plot(Cores, Execution_time)
plt.xlabel('Number of Core')
plt.ylabel('Execution time')
plt.title('Number of Core vs Execution time')
plt.show()
```

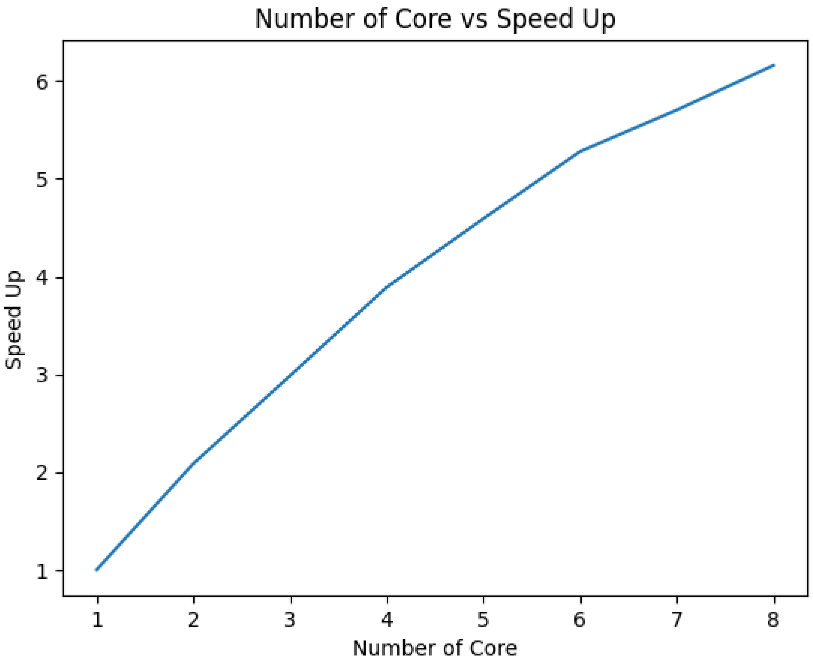


Number of Core vs Speed Up

In [8]:

```python
speed_up = []
for i in Cores:
    speed_up.append(Execution_time[0] / Execution_time[i - 1])

plt.plot(Cores, speed_up)
plt.xlabel('Number of Core')
plt.ylabel('Speed Up')
plt.title('Number of Core vs Speed Up')
plt.show()
```

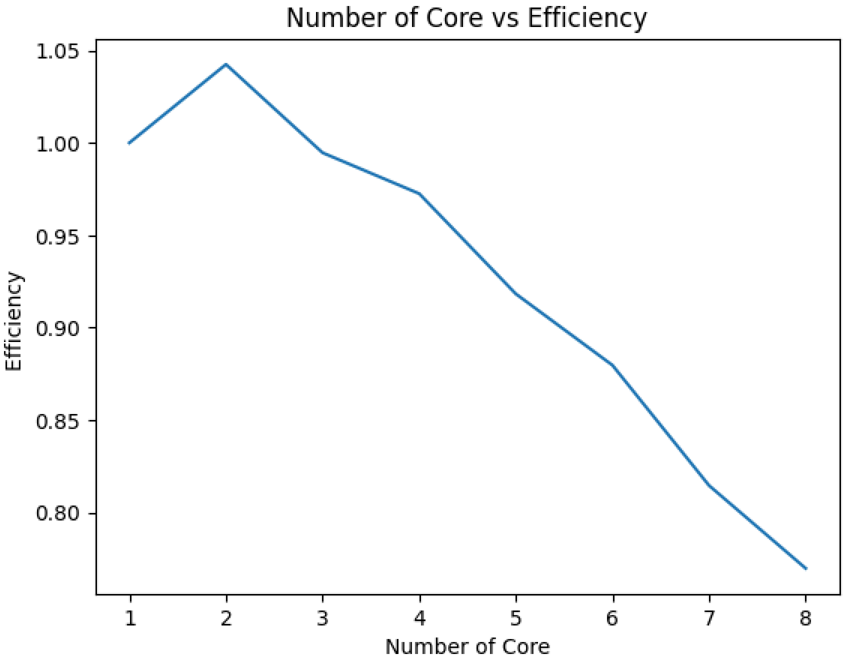


Number of Core vs Efficiency

In [9]:

```
Efficiency = []
for num_of_core in Cores:
    Efficiency.append(speed_up[num_of_core - 1] / num_of_core)

plt.plot(Cores, Efficiency)
plt.xlabel('Number of Core')
plt.ylabel('Efficiency ')
plt.title('Number of Core vs Efficiency')
plt.show()
```



In [ ]: