Creative Digital Media Design
Video Editing via Python

Name: SHOAIB Muhammad **Student ID:** 18079999D

What I want to achieve:
I am a dance enthusiast. I will make a video where the singers are shown to be singing the lyrics, and in perfect synchrony, the dancers are dancing to the lyrics. The video will begin with the title of the song. Then, side by side, the singers from the official music video will be shown who'll be lisping (and singing the song). Alongside, the dancers will be dancing to the song. Then, to focus more on the dance part, I will make the singers from the official music video disappear and let the dancers dance the rest of the part. I end the song when a post-chorus part completes (at 18s). If you liked the song, and forgot the title, I will add the title again but with a vivid and colorful twist. Watch the video to learn about the twist that I learned to code in the lab.

Code:

```
from moviepy.editor import VideoFileClip, concatenate_videoclips,
clips_array, CompositeVideoClip, vfx, afx
import moviepy.video.fx.all as vfx
from math import sin, pi
import numpy as np

video = VideoFileClip("paul_dance.mp4")
video2 = VideoFileClip("title_song.mp4")
video3 = VideoFileClip("julia.mp4")

out_vid = video
out_vid = vfx.mirror_x(out_vid)          #effect 1
out_vid = vfx.blackwhite(out_vid)        #effect 2
out_vid = vfx.fadein(out_vid, 2.0)       #effect 3

def myblink(get_frame, t):  #bonus: defining my own function
  frame = get_frame(t)
  scale = (0.7 + 2*sin(t*pi))
  frame = frame*scale
  frame = frame.astype(np.int8)
  return frame

out_vid = concatenate_videoclips([out_vid, video2.fl(myblink)])
out_vid = clips_array([[out_vid, video2], [video2, out_vid]])
out_vid = out_vid.resize(0.5)             #saving disk space
out_vid = CompositeVideoClip([out_vid, video3.resize(0.5)])
out_vid = afx.audio_fadein(out_vid, 1).   #bonus: audio_fadein
out_vid = afx.audio_fadeout(out_vid, 17.5)#bonus: audio_fadeout

out_vid = vfx.freeze(out_vid, 18, 0.5)    #effect 4

out_vid.write_videofile("18079999_video.mp4")
```

Effects Added:

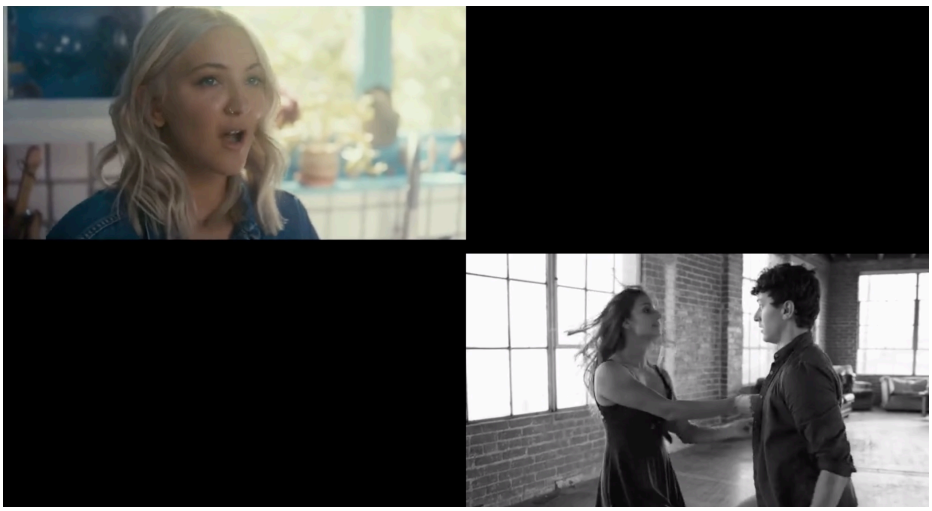
- blackwhite
- fadein
- freeze
- mirror_x

Bonus effect provided by MoviePy:

- audio_fadein (audio smoothly fades in at 1 second rather than starting abruptly)
- audio_fadeout (audio smoothly fades out at 17.5 seconds rather than being abruptly halted)
- custom function defined: myblink

Explanation with Screenshots:

When the video begins (at second 1), all three clips used for making the video were put together in a grid using ***clips_array***.



Since the duration of video2 is shorter, its grid turns black at 3 seconds. The grids that this video was in were the top right and the bottom left as they have turned black at this instant.

At 8 seconds, ***CompositeVideoClip*** was used so as to make the *video3* be replaced with the *video* (that was originally shown in the bottom right), so the *video3* is replaced by *video* here, as shown in the screenshot above.

Special effects were added to the video. Throughout the whole video, four different pre-defined effects were added which are blackwhite to make the colours turn to black and white. Fadein was used for *video* to make it fade in at the start. Mirror_x was used to mirror the video across x-axis. At 18 seconds, freeze was used to stop the video at that particular time frame for half a second. For the bonus task, audio_fadein and audio_fadeout effect was used so that the audio smoothly comes in at 1 second when the song starts and fades out at 17.5 seconds when the song stops. Another function that I defend myself for the bonus task is called myblink, which makes the *video2* (just at the end) blink at an ever-changing colour and contrast, from the time 18s to 20s.

(continues on next page)

Fadein Effect:

Freeze Effect:

Mirror_x and Blackwhite Effects:

Original (non-mirrored and non-blackwhite) With mirror_x and blackwhite effects

Myblink: