

3. Một số hiệu ứng trong MoviePy

01. Resize effect

Import thư viện

```
1 from moviepy.editor import *
```

Sử dụng phương thức fx với đối tượng VideoFileClip

```
1 clip.fx( vfx.resize, width= giá trị)
```

02. Thiết lập vị trí hiển thị

Để thiết lập vị trí hiển thị cho một video bất kỳ ta sử dụng lệnh clip.set_position((x, y))

```
1 clip = clip.set_position((100, 100))
```

Bài tập 1: Cho video jellyfish.mp4 thực hiện các yêu cầu bên dưới

1. Trích xuất 5s đầu tiên của video với tên biến clip1
2. Xoay clip 180 độ với tên biến là clip2
3. Hiệu chỉnh lại kích thước cho clip2 còn 1/2
4. Thiết lập thời gian bắt đầu cho clip2 từ giây thứ 3
5. Thiết lập vị trí cho clip2 với x = y = 100
6. Ghép 2 video clip1 và clip2 sử dụng phương thức composite với tên là final
7. Hiển thị video final với kích thước khung hiển thị là 480

Bài tập 2: Cho video surf.mp4 thực hiện các yêu cầu bên dưới

1. Trích xuất 5s đầu tiên của video với tên biến clip1
2. Thực hiện đối xứng (mirror) video qua trục y với tên biến là clip2 với thời gian bắt đầu (set start) là 4
3. Ghép 2 video clip1 và clip2 sử dụng phương thức composite với tên là final
4. Hiển thị video final với kích thước khung hiển thị là 480

03. Color effect

Hiệu ứng màu được sử dụng để tăng hoặc giảm cường độ màu, khi chúng ta giảm cường độ màu, màu trông tối hơn trong khi tăng cường độ, màu trông sáng hơn. Để thiết lập hiệu ứng màu sắc cho video, ta sử dụng phương thức clip.fx(vfx.colorx, m) với m là cường độ sáng/tối.

```
1 final = clip.fx( vfx.colorx, 1.5)
```

Bài tập 3: Cho video surf.mp4 thực hiện các yêu cầu bên dưới:

1. Trích xuất 5s đầu tiên của Video
2. Áp dụng hiệu ứng màu cho video (1) với cường độ sáng là 1.5
3. Hiển thị kết quả cho video (2)

Bài tập 4: Cho video flower.mp4 thực hiện các yêu cầu bên dưới:

1. Trích xuất 10s đầu tiên của Video
2. Áp dụng hiệu ứng màu cho video (1) với cường độ sáng là 0.5
3. Hiển thị kết quả cho video (2)

04. Speed effect

Tăng tốc video hoặc làm chậm video còn được gọi là hiệu ứng chuyển động nhanh/chuyển động chậm trong sản xuất video, giúp các clip video phát nhanh hơn hoặc chậm hơn tốc độ gốc. Để thiết lập hiệu ứng Speed cho video, ta sử dụng phương thức `clip.fx(vfx.speedx, m)` với `m` là tốc độ chuyển động

```
1 final = clip.fx( vfx.speedx, 0.5)
```

05. Hiệu ứng chuyển tiếp (Transition)

Để chuyển tiếp hiệu ứng giữa các Video, ta sử dụng phương thức `crossfadein`, `crossfadeout`, `slide_in` và `slide_out`

Thiết lập padding

```
1 padding = 1.5 # padding option
```

Thiết lập clip_list

```
1 clip_list=['path to video 1','path to video 2','path to video n']
```

Đọc video list

```
1 video_clips = [VideoFileClip(video) for video in clip_list]
2 video_fx_list = [video_clips[0]]
```

Triển khai hiệu ứng chuyển tiếp

```
1 idx = video_clips[0].duration - padding
2 for video in video_clips[1:]:
3     video_fx_list.append(video.set_start(idx).crossfadein(padding))
```

```
4 idx += video.duration - padding
```

Lưu ý: Cần resize các video về chung kích thước trước khi áp dụng hiệu ứng

Composite và lưu Video

```
1 final_video = CompositeVideoClip(video_fx_list)
2 final_video.write_videofile('my-outfile.mp4', fps=24)
```

Bài tập 5: Thực hiện ghép các video jellyfish.mp4, surf.mp4, bee1.mp4, flower.mp4 với các yêu cầu sau:

1. Trích xuất 5s đầu tiên cho các video trên
2. Tạo hiệu ứng chuyển tiếp cho các video là slide_in với padding là 1
3. Hiển thị video sau khi chỉnh sửa

Bài tập 6: Thực hiện ghép các video jellyfish.mp4, lava.mp4, water.mp4 với các yêu cầu sau:

1. Trích xuất 5s đầu tiên cho các video trên
2. Tạo hiệu ứng chuyển tiếp cho các video là crossfadeout với padding là 2
3. Hiển thị video sau khi chỉnh sửa

Bài tập 7: Thực hiện ghép các video jellyfish.mp4, lava.mp4, water.mp4 với các yêu cầu sau:

1. Trích xuất 5s đầu tiên cho các video trên
2. Tạo 2 hiệu ứng chuyển tiếp khác cho các video crossfadein và slide_out với padding là 1.5
3. Hiển thị video sau khi chỉnh sửa

Gợi ý: `video_fx_list.append(video.set_start(idx).fx(transfx.slide_out, padding, 'vị trí'))` //Vị trí có thể gồm: left - right - bottom - to; transfx bao gồm slide_out và slide_in

06. Chèn Video vào hình ảnh

Import thư viện

```
1 from moviepy.editor import *
2 from moviepy.video.tools.segmenting import findObjects
```

Tải hình ảnh

```
1 im = ImageClip("../ultracompositing/motif.png")
```

Tìm Vùng Để Đặt Video

```
1 regions = findObjects(im)
```

Tải Các Đoạn Video Clip Ngắn

```
1 # Load 7 clips from the US National Parks. Public Domain :D
2 clips = [VideoFileClip(n, audio=False).subclip(18,22) for n in
3     [ "../..../videos/romo_0004.mov",
4       "../..../videos/apis-0001.mov",
5       "../..../videos/romo_0001.mov",
6       "../..../videos/elma_s0003.mov",
7       "../..../videos/elma_s0002.mov",
8       "../..../videos/calio-0007.mov",
9       "../..../videos/grsm_0005.mov"]]
```

Gán Clip Vào Các Vùng Được Xác Định

```
1 comp_clips = [c.resize(r.size)
2               .set_mask(r.mask)
3               .set_pos(r.screenpos)
4               for c,r in zip(clips,regions)]
```

Tạo Video Tổng Hợp

```
1 cc = CompositeVideoClip(comp_clips, im.size)
2 cc.resize(0.6).write_videofile("../..../composition.mp4")
```

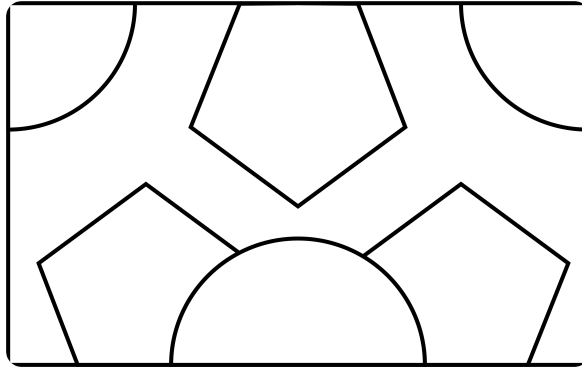
The End Example - MoviePy



Bài tập 8: Thực hiện ghép video vào hình ảnh với các yêu cầu bên dưới:

1. Thực hiện tách video [VIETNAMBEAUTY.mp4](#) thành 7 clip ngắn với thời gian mỗi clip là 10 giây

2. Thực hiện ghép các video sau khi tách vào ảnh bên dưới [motif2.png](#)



3. Hiển thị kết quả sau khi ghép

4. Lưu file kết quả lại với tên Ex5-final.mp4

Bài tập 9: Sử dụng các phương thức `moviepy.editor`, `vfx` trong `moviepy.video.fx.all` và `ndimage` trong `scipy.ndimage` để thực hiện tùy chỉnh cho video [bee1.mp4](#):

1. Đọc và hiển thị video `bee1.mp4`
2. Xây dựng hàm làm mờ ảnh gồm 2 đối số `img` và `sigma` (mặc định là 3) với bộ lọc `gaussian_filter` thuộc `ndimage`
3. Triển khai làm mờ ảnh trong video `bee1.mp4`
4. Hiển thị kết quả sau khi chỉnh sửa