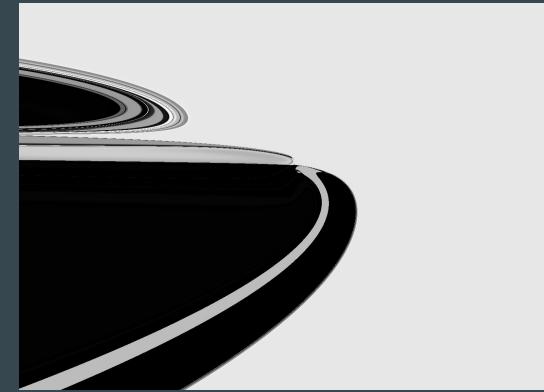
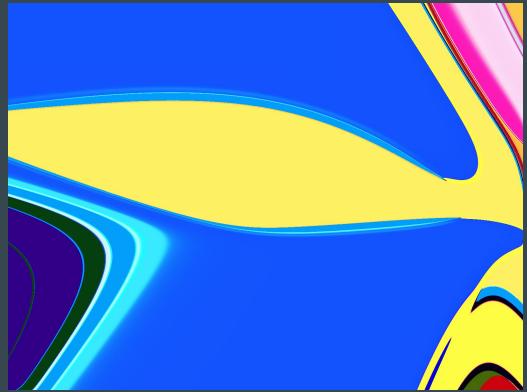


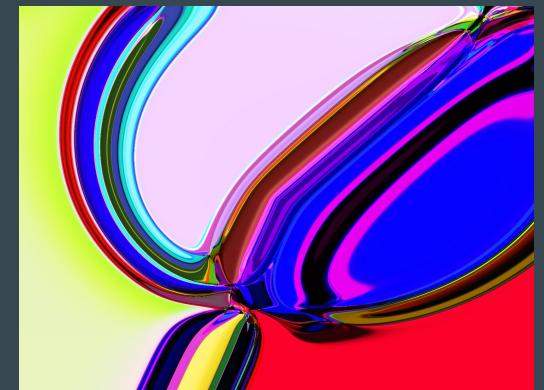
# Soul Snapshot

• • •

Nam Tran



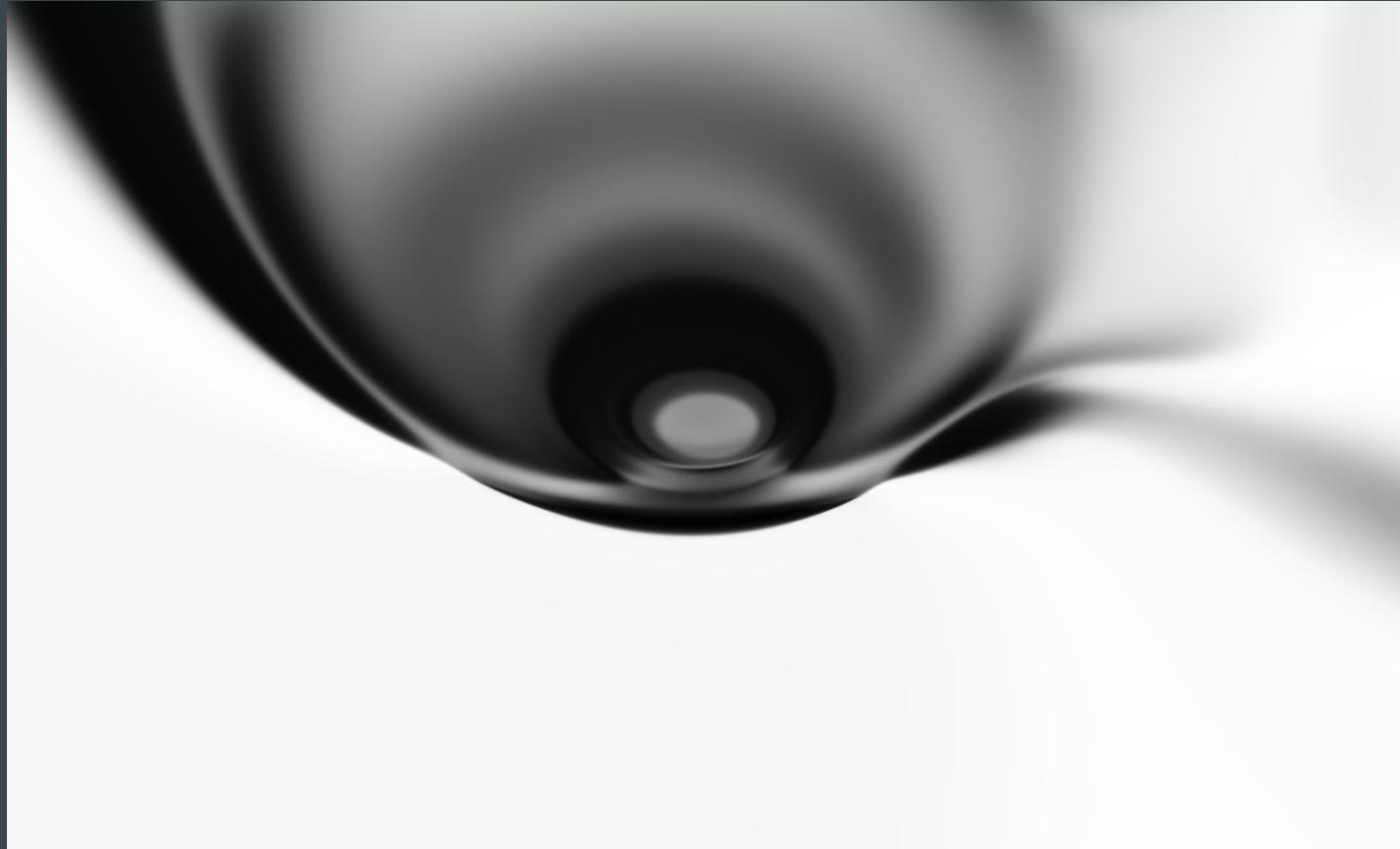
# A Twitter bot for modern/neural art fans

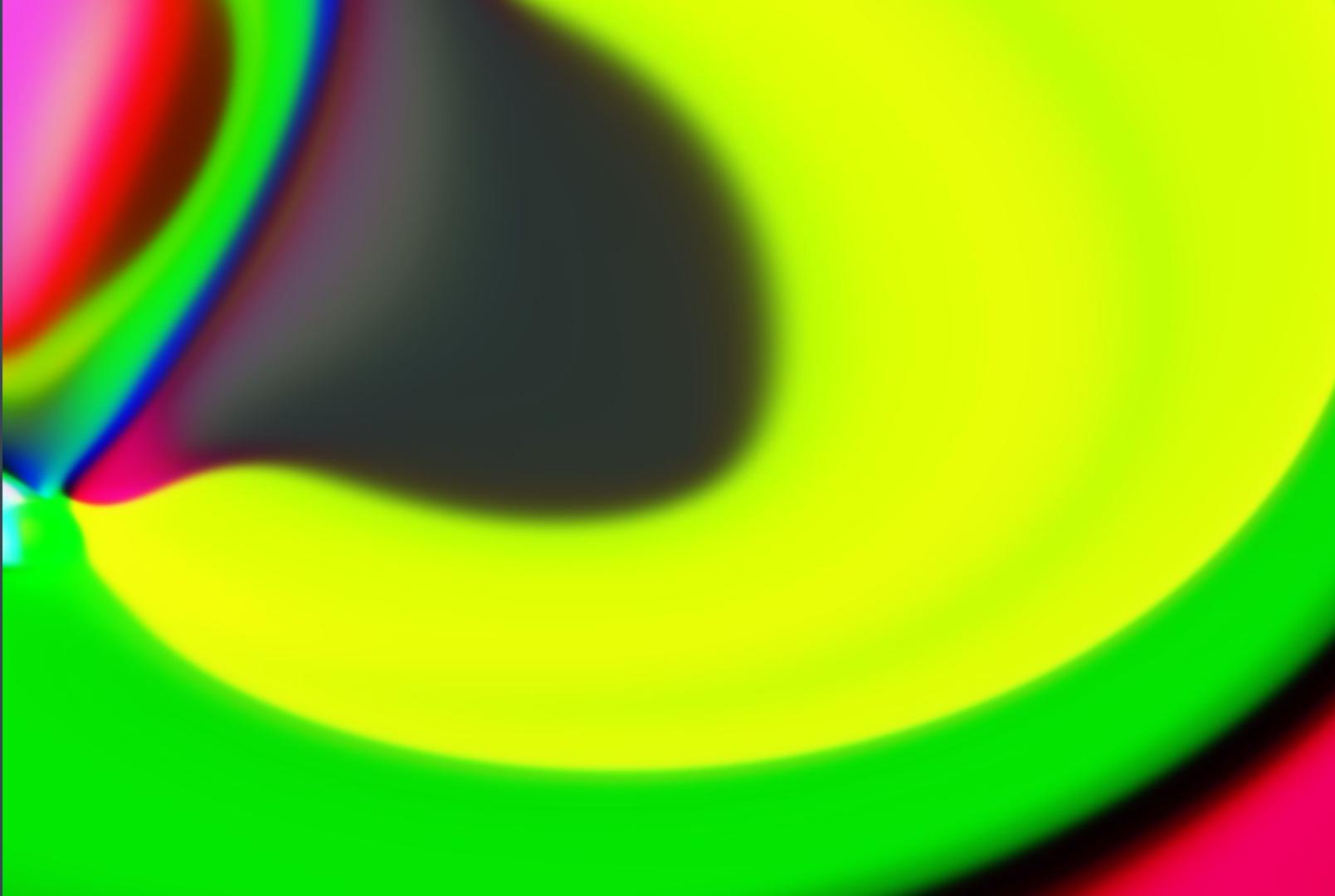


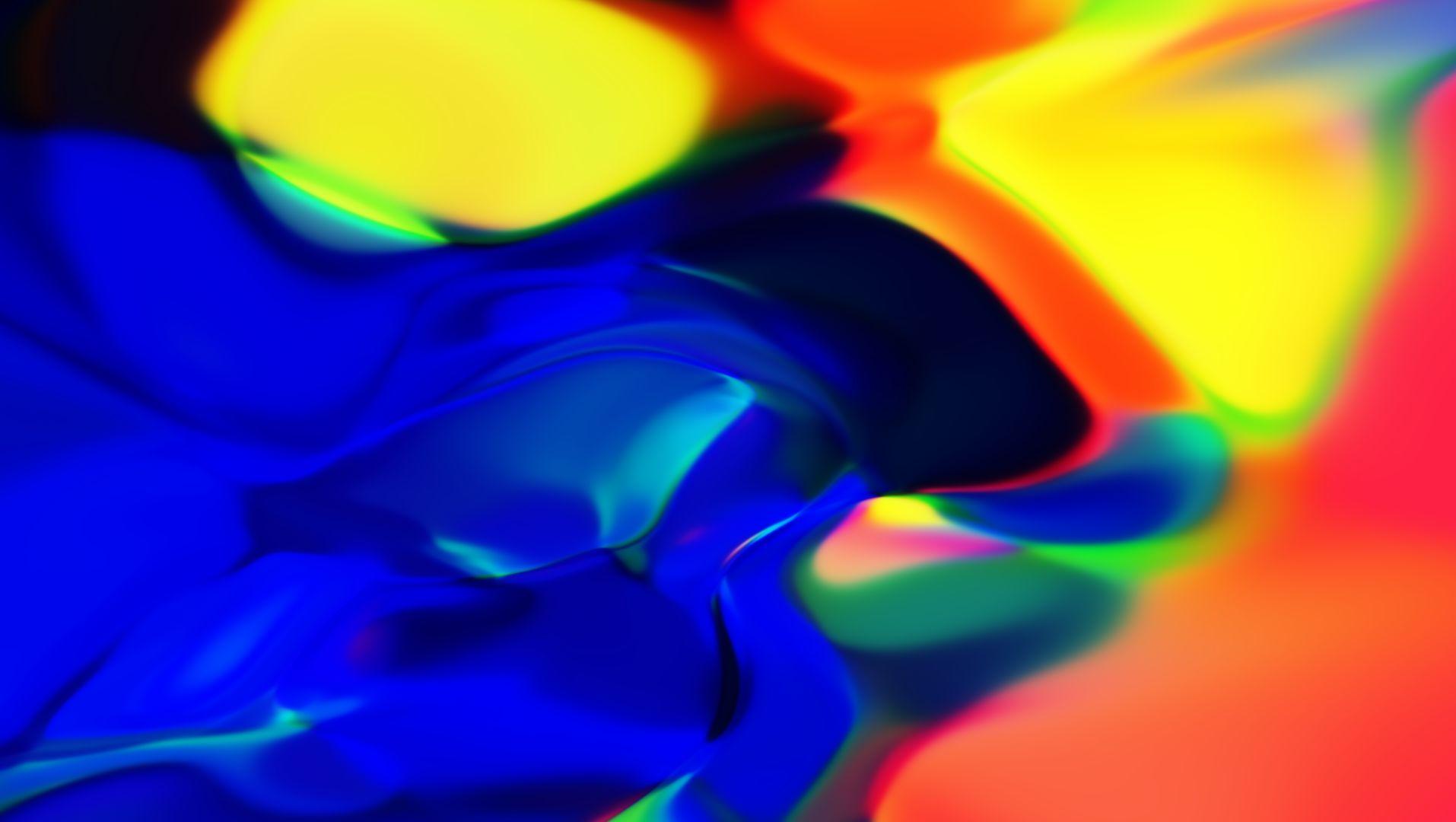
# Features and Libraries:

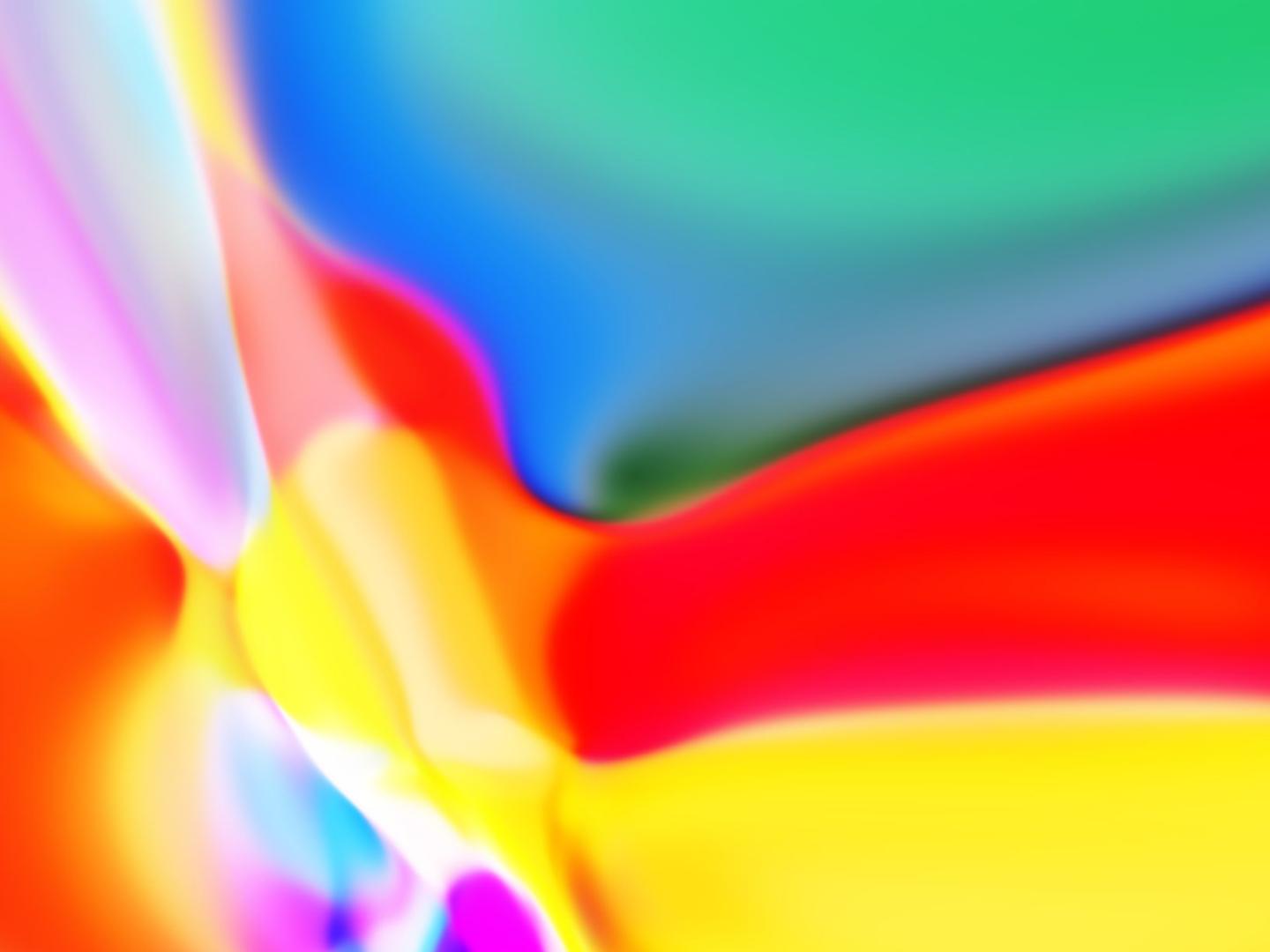
- Neural network for artwork generation: **neuralart**
- Sentiment analysis: **textblob**
- Image processing and analysis: **PIL**
- Markov chain for text generation: **markovify**
- Multi-threading: **threading**
- Twitter API access: **tweepy**
- Miscellaneous for data retrieval: **wget, time, datetime, numpy**

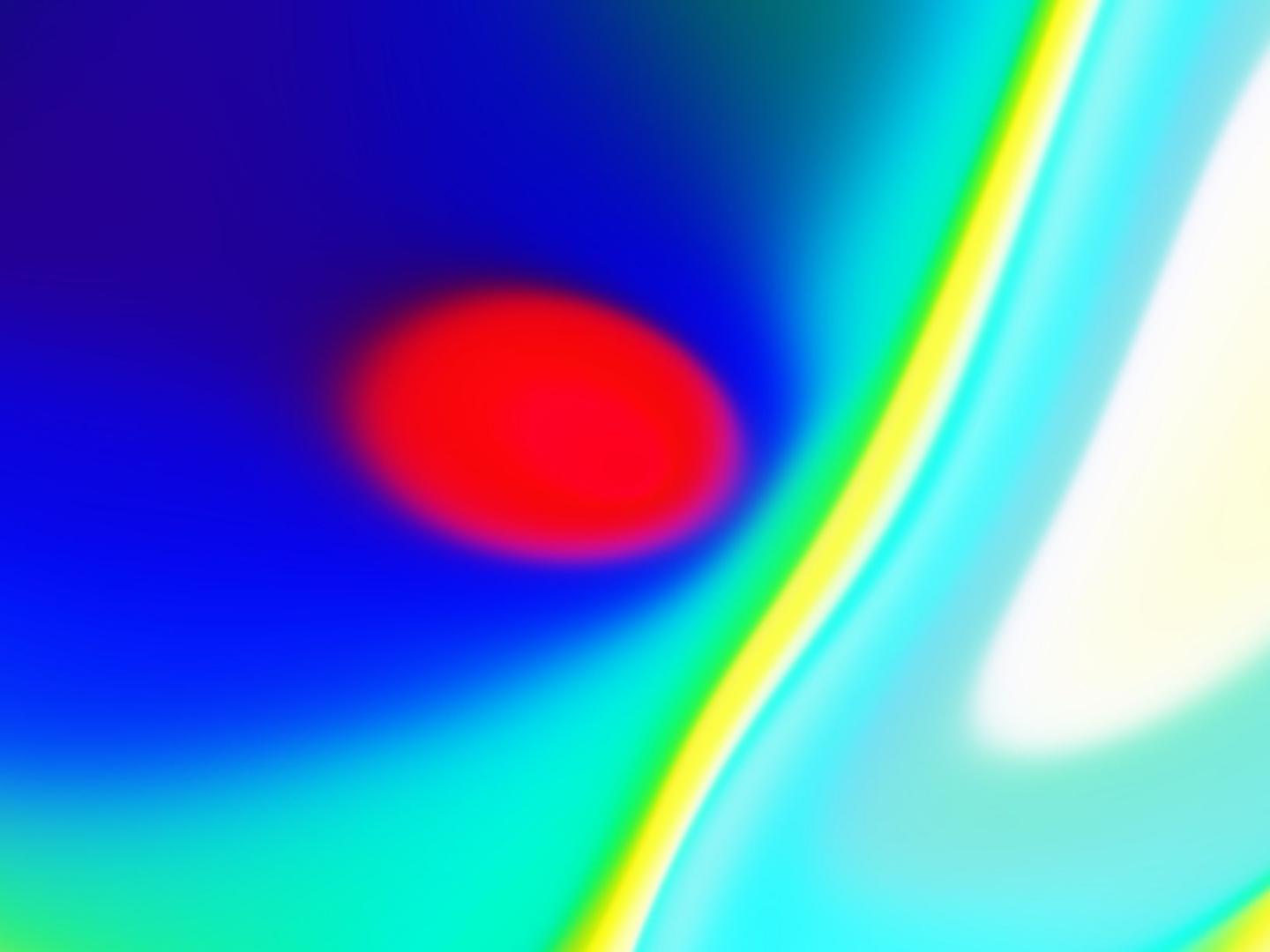
# Neural Network-generated Artworks

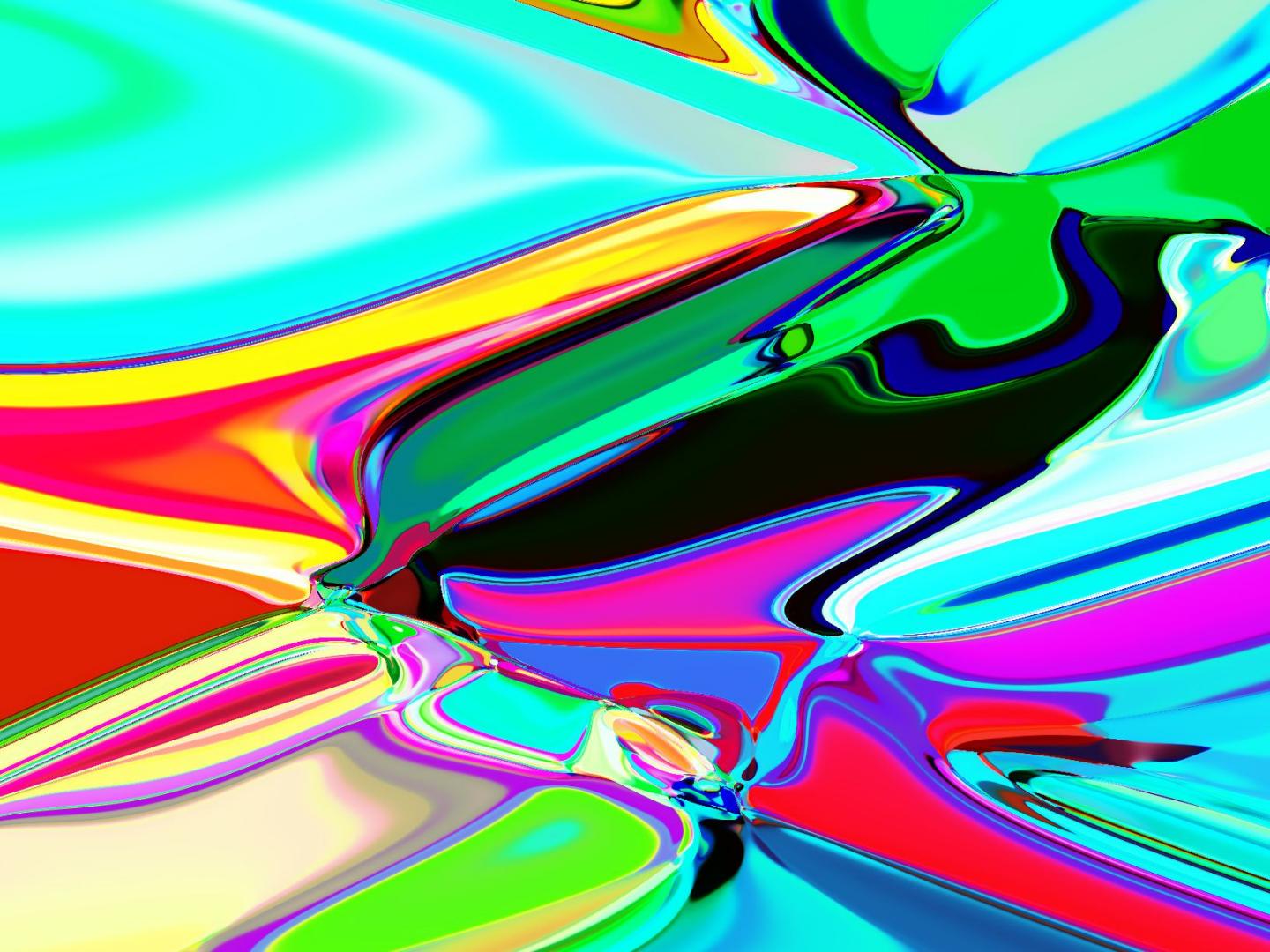












# Text Generation using Markovify Learning Models

```
33 def generate_text(tweet):
34     #generates text using Markov chains
35
36     with open("./the_picture_of_dorian_gray.txt", encoding="utf8") as f:
37         the_picture_of_dorian_gray = f.read()
38     with open("./declaration_of_independence.txt", encoding="utf8") as f2:
39         declaration_of_independence = f2.read()
40     with open("./jane_eyre.txt", encoding = "utf8") as f3:
41         declaration_of_independence = f3.read()
42     with open("./war_and_peace.txt", encoding="utf8") as f4:
43         war_and_peace = f4.read()
44     with open("./jane_eyre.txt", encoding="utf8") as f5:
45         jane_eyre = f5.read()
46     with open("./jokes.txt", encoding="utf8") as f6:
47         jokes = f6.read()
48     with open("./moby_dick.txt", encoding="utf8") as f7:
49         moby_dick = f7.read()
50     with open("./pride_and_prejudice.txt", encoding="utf8") as f8:
51         pride_and_prejudice = f8.read()
52
53 model_a = markovify.Text(the_picture_of_dorian_gray, state_size = 4)
54 model_b = markovify.Text(declaration_of_independence, state_size = 4)
55 model_c = markovify.Text(jane_eyre, state_size = 4)
56 model_d = markovify.Text(war_and_peace, state_size = 4)
57 model_e = markovify.Text(jokes, state_size = 4)
58 model_f = markovify.Text(pride_and_prejudice, state_size = 4)
59 model_g = markovify.Text(moby_dick, state_size = 4)
60 model_tweet = markovify.Text(tweet.full_text, state_size = 4)
61
62 model_combo = markovify.combine([model_a, model_b, model_c, model_d, model_e, model_tweet], [1.4, 1.25, 1.25, 1.4, 1.25, 1.0])
63
64 return model_combo.make_short_sentence(280)
```

# Sentiment Extraction during Tweet Analysis

```
171 def analyze_tweet(tweet):
172     #analyzes a tweet to see if it contains photos or not
173
174     contains_photos = False
175     try:
176         print(True in [medium['type'] == 'photo' for medium in tweet.entities['media']])
177         contains_photos = True
178     except:
179         print("No picture in this tweet")
180
181     testimonial = TextBlob(tweet.full_text)
182     polarity = testimonial.sentiment.polarity
183     subjectivity = testimonial.sentiment.subjectivity
184     print("Polarity value: {}. Possible range: [-1,1]".format(polarity))
185     print("Subjectivity value: {}. Possible range: [0,1]".format(subjectivity))
186     print("Value of contains_photos: {}".format(contains_photos))
187
188     return contains_photos, polarity, subjectivity
189
```

# Polarity vs Subjectivity

Polarity ranges from -1 to 1 (negative versus positive).

Subjectivity ranges from 0 to 1 (objective versus subjective).

These values, among with other characteristics extracted from a tweet, are feeded into the neural network to generate artworks.

Uses numpy's interp function to map polarity and subjectivity values to ranges of [3, 13] and [3, 40] for DEPTH and RENDER\_SEED variables respectively.

# Image Generation using Neural Network, Sentiment Analysis and Image Processing & Analysis

```
65
66     def generate_image(tweet, contains_photos, polarity, subjectivity):
67         #generates image by feeding sentiment analysis results into a neural network
68
69         #modes = {"I": 1, "L": 8, "P": 8, "RGB": 24, "RGBA": 32, "CMYK": 32, "YCbcCr": 24, "LAB": 24, "HSV": 24, "I": 32, "F": 32}
70
71         if contains_photos is True: #when a user's tweet contains photos
72             image_files = []
73             if 'media' in tweet.entities:
74                 for image in tweet.entities['media']:
75                     media_url = str(image['media_url'])
76                     image_files.append(media_url)
77
78             for image in image_files:
79                 wget.download(image, "download_image.png")
80
81             seed_path = "download_image.png"
82             seed_image = Image.open(seed_path)
83             im_obj = ImageStat.Stat(seed_image)
84
85             UNITS = int(subjectivity*2+10)
86             DEPTH = int(interp(polarity, [-1,1],[3,13]))
87
88             if seed_image.mode == "RGB":
89                 CHANNELS = 3
90                 RENDER_SEED = int(interp(subjectivity, [0,1],[3,40]))
91                 Z_DIMS = int((im_obj.stddev[0] + im_obj.stddev[1] + im_obj.stddev[2])/3)
92             else:
93                 CHANNELS = 1
94                 RENDER_SEED = int(interp(subjectivity, [0,1],[3,200]))
95                 min_max = seed_image.getextrema()
96                 Z_DIMS = random.randint(1, min_max[1] - min_max[0])
97                 os.remove(seed_path) #remove the downloaded photo
98
99             else: #when a user's tweet contains text only
100                 UNITS = int(subjectivity*2+10)
101                 DEPTH = int(interp(polarity, [-1,1],[3,13]))
102                 CHANNELS = 1
103                 RENDER_SEED = int(interp(subjectivity, [0,1],[3,40]))
104                 Z_DIMS = int(interp(subjectivity, [0,1],[1,40]))
105
106             ITERATIONS = 10
107             #full-hd resolution
108             WIDTH = 1920
109             HEIGHT = 1080
110             HIDDEN_STD = float(random.randrange(100, 500))/100
111             OUTPUT_STD = 1.0
112             RADIUS = True
113             BIAS = True
114             zfill = len(str(ITERATIONS - 1))
115             z = [-1.0] * Z_DIMS
116             step_size = 1/ITERATIONS #the higher the step size, the quicker the pattern transformation
```

# Continued

```
116
117     print("")
118     print("-----RENDERING PARAMETERS-----")
119     print("SEED: {}".format(RENDER_SEED))
120     print("DEPTH: {}".format(DEPTH))
121     print("WIDTH: {}".format(WIDTH))
122     print("HEIGHT: {}".format(HEIGHT))
123     print("CHANNELS: {}".format(CHANNELS))
124     print("Z-DIMENSIONS: {}".format(Z_DIMS))
125     print("STEP SIZE: {}".format(step_size))
126     print("UNITS: {}".format(UNITS))
127     print("HIDDEN STD: {}".format(HIDDEN_STD))
128     print("OUTPUT STD: {}".format(OUTPUT_STD))
129     print("RADIUS: {}".format(RADIUS))
130     print("BIAS: {}".format(BIAS))
131     print("ITERATIONS: {}".format(ITERATIONS))
132     print("")
133
134
135 generated_images = []
136 for x in range(ITERATIONS):
137     result = neuralart.render(
138         xres = WIDTH,
139         yres = HEIGHT,
140         seed = RENDER_SEED,
141         channels = CHANNELS,
142         hidden_std = HIDDEN_STD,
143         output_std = OUTPUT_STD,
144         units = UNITS,
145         depth = DEPTH,
146         device = 'cpu',
147         radius = RADIUS,
148         bias = BIAS,
149         z=z
150     )
151     im = Image.fromarray(result.squeeze())
152     generated_images.append(im)
153     print("Completed iteration {}".format(x+1))
154
155 buckets = []
156 probability_distribution = []
157 for i in range(0, len(generated_images)):
158     buckets.append(i)
159
160 for i, value in enumerate(generated_images):
161     if i > len(generated_images)/2:
162         probability_distribution.append(i/len(generated_images))
163     else:
164         probability_distribution.append(0)
165
166 chosen_image_id = int(choice(buckets, 1, probability_distribution))
```

# UTC to Local Time

Tweepy's returned timestamp is given in UTC time zone. For multi-threaded monitoring purposes, this has to be converted to local time zone.

```
199
200 def utc_to_local(utc_datetime):
201     #converts utc time to local time
202
203     now = time.time()
204     offset = datetime.fromtimestamp(now) - datetime.utcfromtimestamp(now)
205     return utc_datetime + offset
206
```

# Main - Autonomous Monitoring

Spins threads to handle new tweets. Sleep time to comply with Twitter's rate limit.

```
206
207 def main():
208     #uses multi-threading to listen on the entire Twitter user space for mentions
209
210     time_now = datetime.now()
211     tweets = []
212     cold_start = True
213     wait_time = 300
214
215     while True:
216         if cold_start == True:
217             for tweet in tweepy.Cursor(api.search, q = "@Soul_Snapshot", tweet_mode = 'extended').items(10):
218                 tweets.append(tweet)
219
220         else:
221             for tweet in tweepy.Cursor(api.search, q = "@Soul_Snapshot", tweet_mode = 'extended').items(10):
222                 if tweet not in tweets:
223                     tweets.append(tweet)
224
225         cold_start = False
226         count = 0
227         for tweet in tweets:
228             post = tweet
229             post_time = utc_to_local(post.created_at) #tweet.created_at returns UTC timezone
230             #if post_time > time_now:
231             print("Now responding to user {}, who tweeted: {}, at {}".format(tweet.user.screen_name, tweet.full_text, post_time))
232             t = threading.Thread(target = respond, args = (post,))
233             t.start()
234             t.join()
235             count+=1
236
237         if count == 0:
238             print("No new interaction detected. Sleep for {} seconds and check again".format(wait_time))
239         else:
240             print("Responded to all new tweets. Sleep for {} seconds and check again".format(wait_time))
241             time_now = datetime.now()
242             sleep(wait_time)
243
244 main()
```

# Demo

```
PS C:\Users\Nam\Desktop\COEN 166\Twitter Bot> python soul_art.py
No new interaction detected. Sleep for 900 seconds and check again
No new interaction detected. Sleep for 900 seconds and check again
No new interaction detected. Sleep for 900 seconds and check again
Now responding to user negativethegod, who tweeted: @Soul_Snapshot we are back my friend https://t.co/OGNfbmxGU
T, at 2019-06-03 19:04:39
True
Generated text: Now what do you call a little burro?
Polarity value: 0.0. Possible range: [-1,1]
Subjectivity value: 0.0. Possible range: [0,1]
value of contains_photos: True
100% [.....] 99158 / 99158
-----RENDERING PARAMETERS-----
SEED: 1
DEPTH: 8
WIDTH: 1920
HEIGHT: 1080
CHANNELS: 3
Z-DIMENSIONS: 64
STEP SIZE: 0.1
UNITS: 10
HIDDEN STD: 4.38
OUTPUT STD: 1.0
RADIUS: True
BIAS: True
ITERATIONS: 10

Completed iteration 1
Completed iteration 2
Completed iteration 3
Completed iteration 4
Completed iteration 5
Completed iteration 6
Completed iteration 7
```

Completed iteration 1  
Completed iteration 2  
Completed iteration 3  
Completed iteration 4  
Completed iteration 5  
Completed iteration 6  
Completed iteration 7  
Completed iteration 8  
Completed iteration 9  
Completed iteration 10  
Response tweeted  
Now responding to user negativethegod, who tweeted: @Soul\_Snapshot Hello Corleone, at 2019-06-03 19:03:25  
No picture in this tweet  
Generated text: "General Barclay de Tolly risked his life everywhere at the head of the countermovement from east to west.  
Polarity value: 0.0. Possible range: [-1,1]  
Subjectivity value: 0.0. Possible range: [0,1]  
Value of contains\_photos: False

-----RENDERING PARAMETERS-----

SEED: 1  
DEPTH: 8  
WIDTH: 1920  
HEIGHT: 1080  
CHANNELS: 1  
Z-DIMENSIONS: 1  
STEP SIZE: 0.1  
UNITS: 10  
HIDDEN STD: 4.45  
OUTPUT STD: 1.0  
RADIUS: True  
BIAS: True  
ITERATIONS: 10

```
Completed iteration 1
Completed iteration 2
Completed iteration 3
Completed iteration 4
Completed iteration 5
Completed iteration 6
Completed iteration 7
Completed iteration 8
Completed iteration 9
Completed iteration 10
Response tweeted
Now responding to user negativethegod, who tweeted: @Soul_Snapshot we are back my friend https://t.co/OGNfbmxGU
T, at 2019-06-03 19:04:39
True
Generated text: what lies on the bottom of the long room, on the hearth; for there was a fire there too, but no
candle, and no Mrs. Fairfax.
Polarity value: 0.0. Possible range: [-1,1]
Subjectivity value: 0.0. Possible range: [0,1]
value of contains_photos: True
100% [.....] 99158 / 99158
-----RENDERING PARAMETERS-----
SEED: 1
DEPTH: 8
WIDTH: 1920
HEIGHT: 1080
CHANNELS: 3
Z-DIMENSIONS: 64
STEP SIZE: 0.1
UNITS: 10
HIDDEN STD: 1.6
OUTPUT STD: 1.0
RADIUS: True
BIAS: True
ITERATIONS: 10
```

[Home](#)[Moments](#)[Notifications](#) 4[Messages](#)

Search Twitter



Tweet



## Soul Snapshot

@Soul\_Snapshot

Tell me something, using words and a

Tweets  
**20**Following  
**2**Followers  
**4**Lists  
**0**Moments  
**0**[Edit profile](#)[Tweets](#)[Tweets & replies](#)[Media](#)**Soul Snapshot** @Soul\_Snapshot · Jun 2

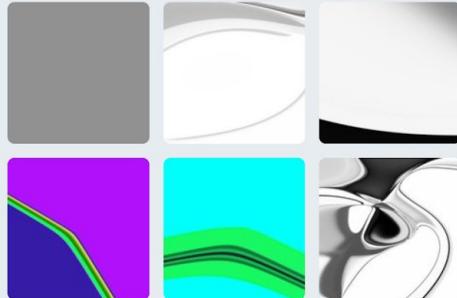
I discovered, too, that a knight is always stronger than one, while in war a battalion is sometimes stronger than a married man's should be for his friend's.

photo, and I will reveal to you an artistic snapshot of your soul. It takes time and effort to understand what I say.

Joined May 2019

Born September 8, 1997

20 Photos and videos



## Trends for you · Change

#THISNIGHTbyJIN

BTS star Jin releases new solo music

#NationalCheeseDay

13.1K Tweets

fiancée.



Soul Snapshot @Soul\_Snapshot · Jun 2

He gave her a shilling: she put it into an old man's penetration he understood all that he himself was killed, but he quickly roused himself with a sinking heart—watching the bridge and stopped the infantry.

### #THISNIGHTbyJIN

BTS star Jin releases new solo music

### #NationalCheeseDay

13.1K Tweets

### #이밤ByJin

332K Tweets

### #ReasonsYouStayOnTwitter

13.4K Tweets

### #TuesdayThoughts

76.5K Tweets

### Eid Mubarak

Celebrities and world leaders post their Eid wishes

### #PetsForJin

4,048 Tweets

### #SIS19

### Kim Seokjin

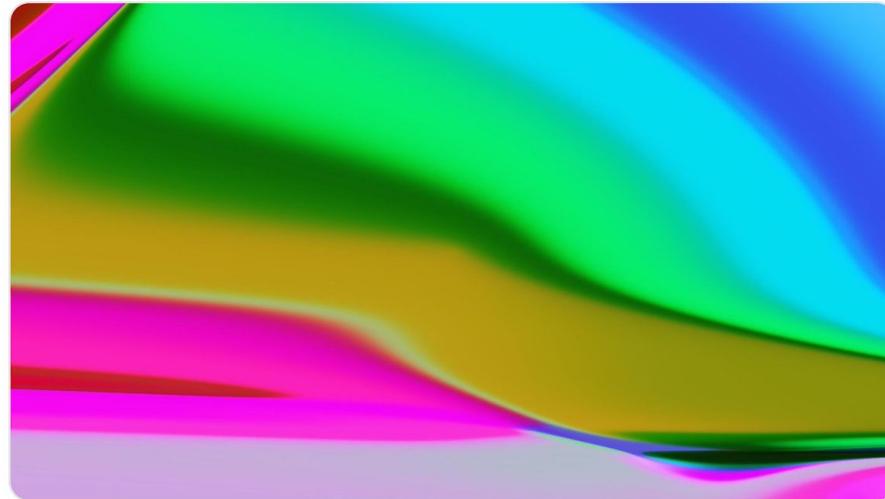
113K Tweets

### #U20WC

15.3K Tweets

Soul Snapshot @Soul\_Snapshot · Jun 2

He gave her a shilling: she put it into an old man's penetration he understood all that he himself was killed, but he quickly roused himself with a sinking heart—watching the bridge and stopped the infantry.



Soul Snapshot @Soul\_Snapshot · Jun 1

first snapshot test