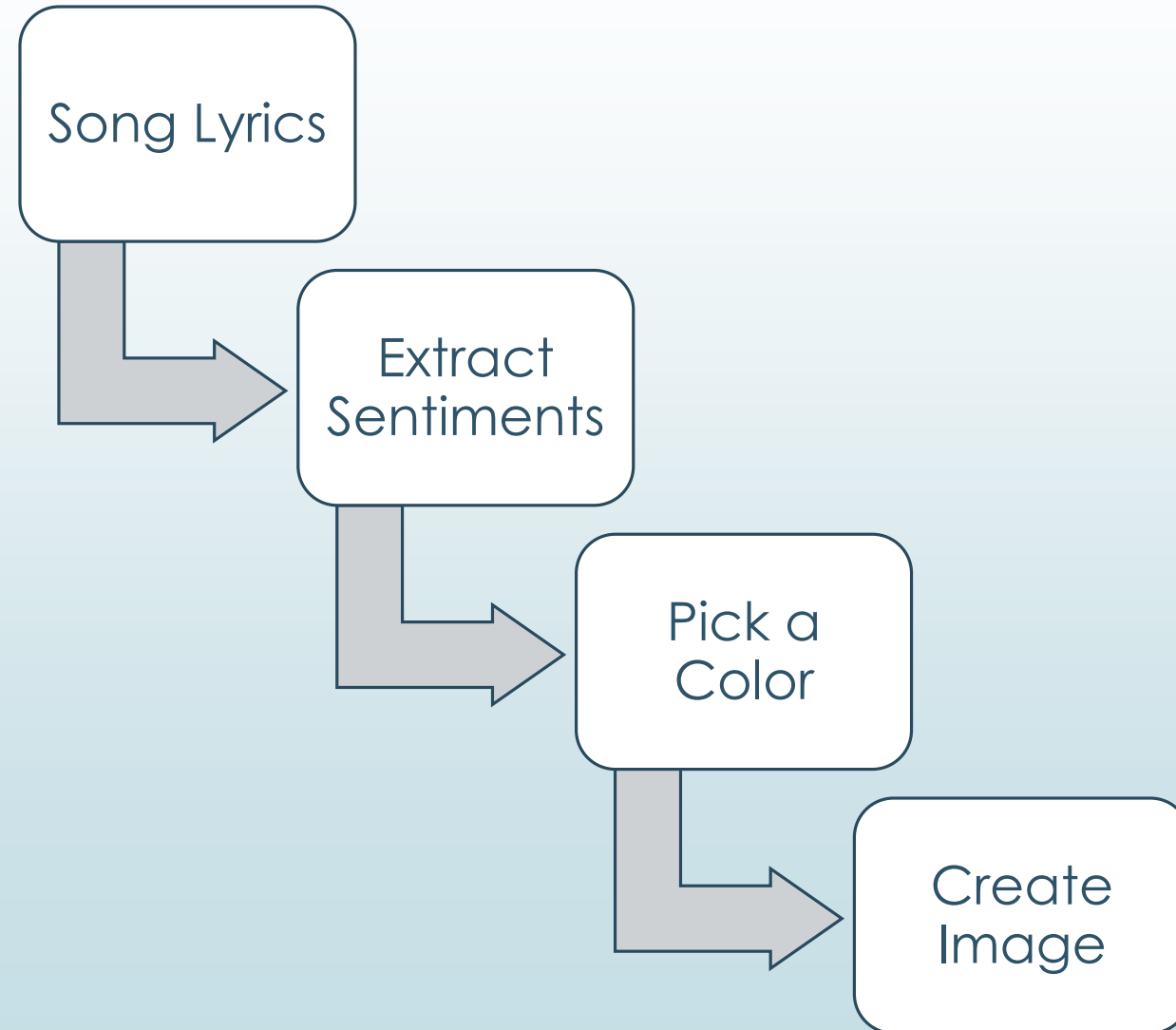




ART CONVERSION

Songs → Images!

FLOW





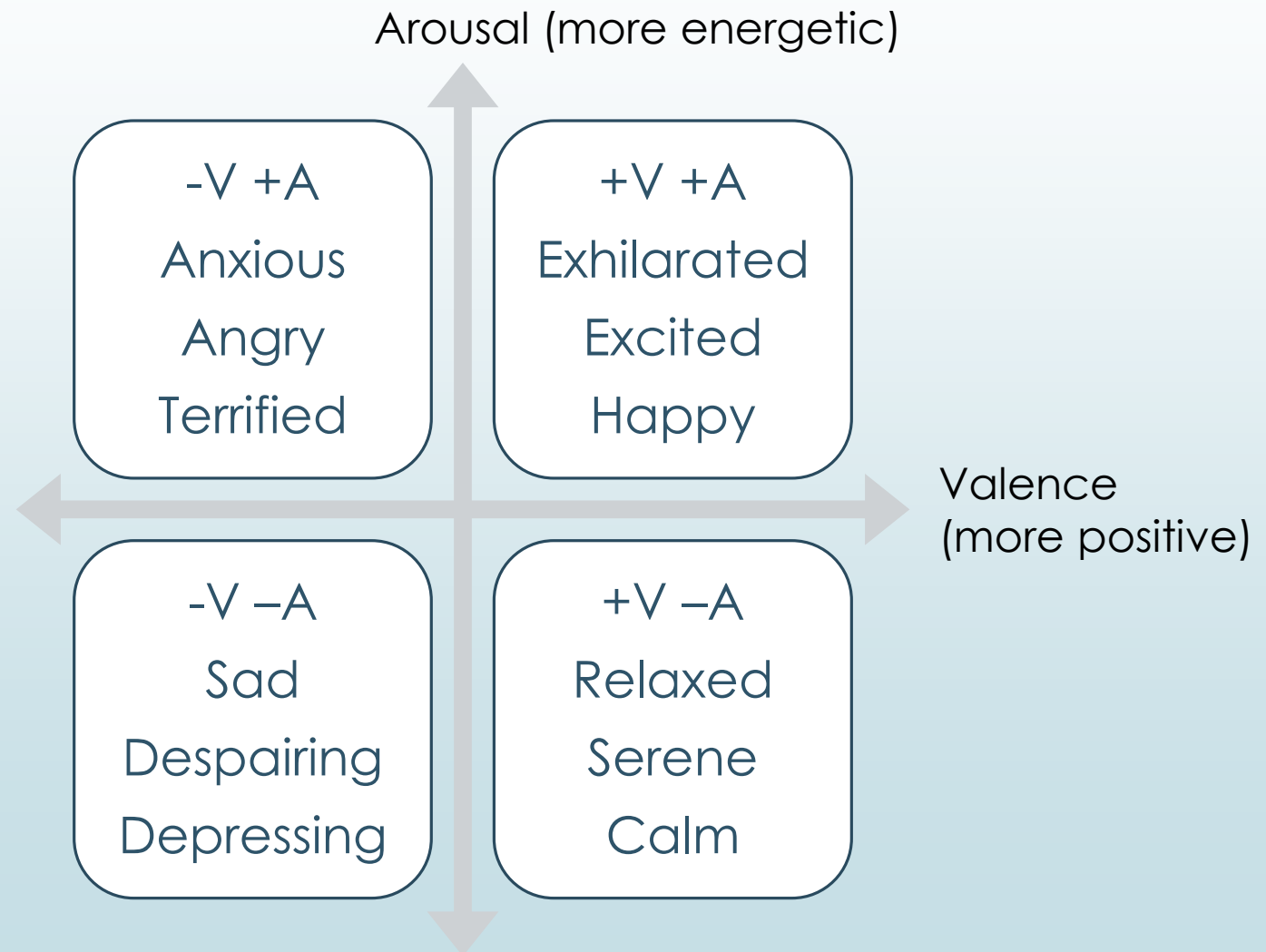
SONG LYRICS

- ▶ API's used:

- ▶ Last.fm: <http://www.last.fm/api>

- ▶ PyLyrics: Python module to get lyrics from lyrics.wikia.com
<https://pypi.python.org/pypi/PyLyrics/>

SENTIMENT ANALYSIS: Russell's Model of Mood



SENTIMENT ANALYSIS

► Dictionaries used:

► ANEW (Affective Norms for English Words)

- List of ~1000 English words

- Values of Valence and Arousal for each word on a scale of 1 to 10

► Extended ANEW dictionary

- List of ~14k English words

► Wordnet (Princeton)

- POS tags (E.g.: 'n' – Noun, 'v' – Verb, 'a' – Adjective, 'r' – Adverb)

- Synsets

E.g.: `wn.synsets('happy')` → [`Synset('happy.a.01')`, `Synset('felicitous.s.02')`,
`Synset('glad.s.02')`, `Synset('happy.s.04')`]

SENTIMENT ANALYSIS

- ▶ Tokenize Lyrics
 - ▶ Split the entire lyrics into a list of POS-tagged tokens
- ▶ Get the root word for each of the tokens
 - ▶ E.g.: `wn.synsets('loving', pos = 'v')` → `[Synset('love.v.01'), ...]`
- ▶ Get the valence and arousal values for each word
- ▶ Calculate the mean valence and arousal to get the sentiment of the song

	Arousal > 5	Arousal < 5
Valence > 5	Happy	Calm
Valence < 5	Angry	Sad

COLORS AND SENTIMENTS

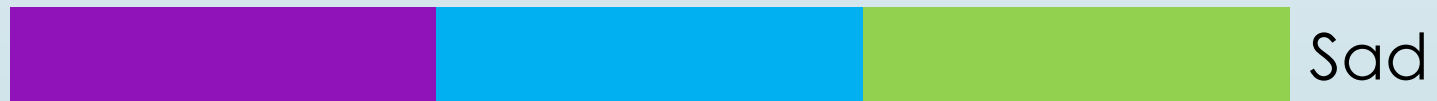
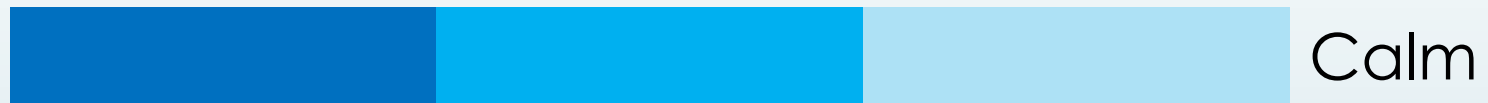


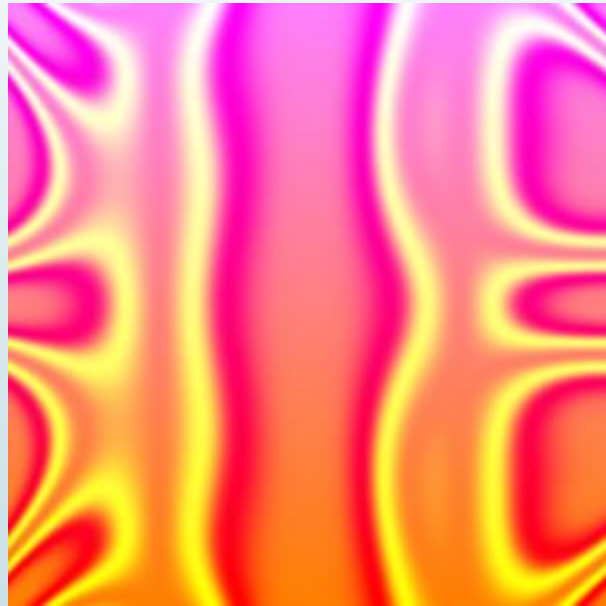


IMAGE GENERATION

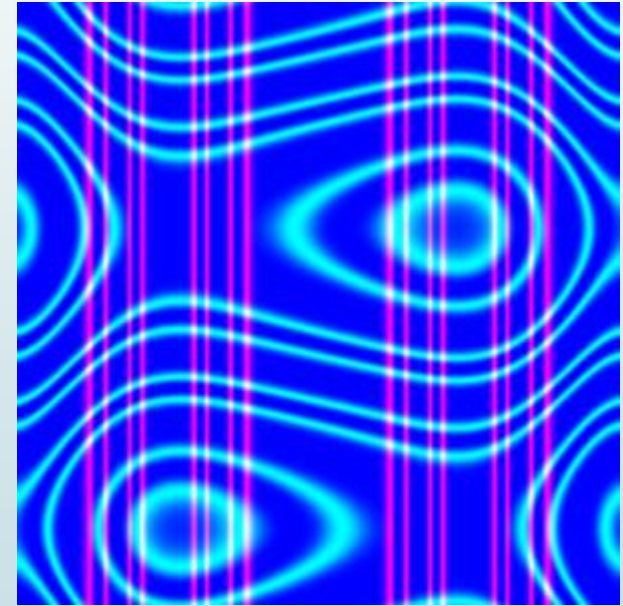
- Use Mathematical Functions to generate random images
- Generate three “infinite” composite functions (f, g, h)
E.g.: $\cos(\pi \cdot \cos(\pi \cdot \sin(\pi \cdot \cos(\pi \cdot y))))$
- At each pixel, calculate $f(x, y), g(x, y), h(x, y)$
- Scale these values to get a pixel intensity i.e. 0 to 255.
- Generate an RGB image using the intensity values obtained.

IMAGE GENERATION (EXAMPLES)

Limits (Happy)



Best Mistake (Calm)





THANK YOU

PANKTI KANSARA