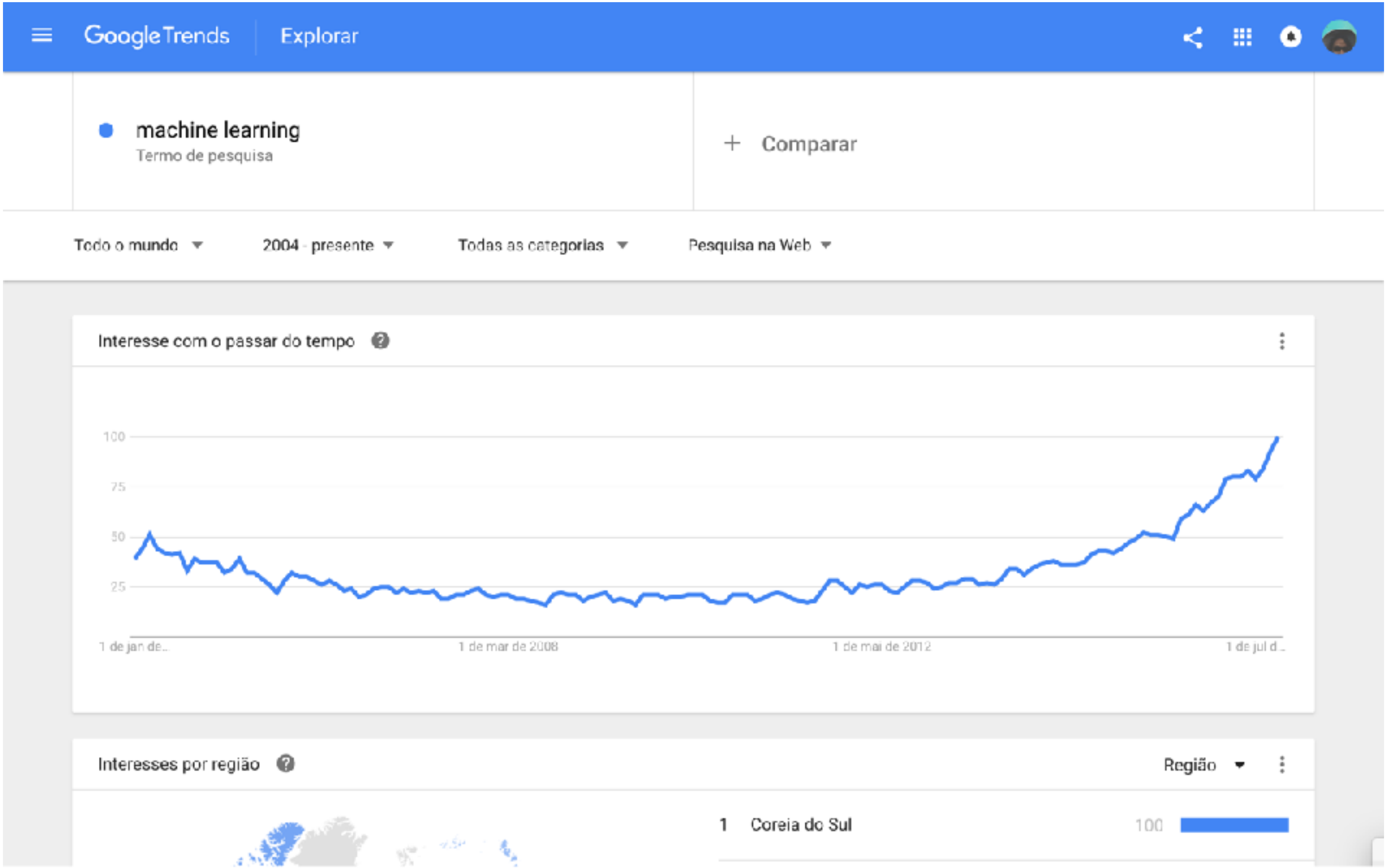
Google ❤️ Machine Learning

@olarclara || #gdgtechtour

Maria Clara

- Embaixadora Auth0;
- Women Techmakers Lead;
- Entusiasta de Machine Learning;


Google Trends
Explorar
machine learning
Termo de pesquisa
Comparar
Todo o mundo
2004 - presente
Todas as categorias
Pesquisa na Web
Interesse com o passar do tempo
100
75
50
25
1 de jan de...
1 de mar de 2008
1 de mai de 2012
1 de jul d...
Interesses por região
Região
1
Coreia do Sul
100

Bringing Impressionism to Life with Neural Style Transfer in *Come Swim*

Bhautik J Joshi*
Research Engineer, Adobe

Kristen Stewart
Director, *Come Swim*

David Shapiro
Producer, Starlight Studios

Figure 1: *Usage of Neural Style Transfer in Come Swim, left: content image, middle: style image, right: upsampled result. Images used with permission, (c) 2017 Starlight Studios LLC & Kristen Stewart.*

arXiv:1701.04928v1 [cs.CV] 18 Jan 2017

Abstract

Neural Style Transfer is a striking, recently-developed technique that uses neural networks to artistically redraw an image in the style of a source style image. This paper explores the use of this technique in a production setting, applying Neural Style Transfer to redraw key scenes in *Come Swim* in the style of the impressionistic painting that inspired the film. We document how the technique can be driven within the framework of an iterative creative process to achieve a desired look, and propose a mapping of the broad parameter space to a key set of creative controls. We hope that this mapping can provide insights into priorities for future research.

Keywords: style transfer, rendering, applied computer graphics

Concepts: •**Computing methodologies** → *Computer graphics;* **Image-based rendering;** •**Applied computing** → *Media arts;*

1 Introduction

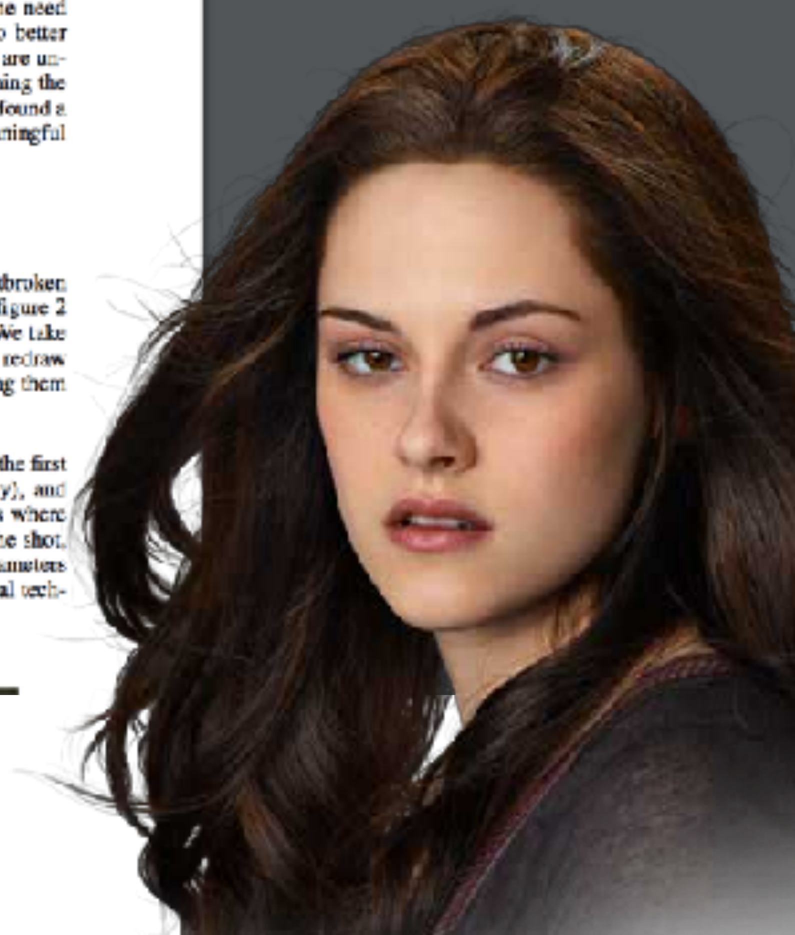
In *Image Style Transfer Using Convolutional Neural Networks*, Gatys et al [Gatys et al. 2015] outline a novel technique using convolutional neural networks to re-draw a content image in the broad artistic style of a single style image. A wide range of implementations have been made freely available [Liu 2016; Johnson 2015; Athalye 2015], based varyingly on different neural network evaluators such as Caffe [Jia et al. 2014] and Tensorflow [Abadi and et al 2016] and wrappers such as Torch and PyCaffe [Bahrampour et al. 2015]. There has been a strong focus on automatic techniques, even with extensions to coherently process video [Ruder et al. 2016].

execute efficiently and predictably. In a production setting, however, a great deal of creative control is needed to tune the result, and a rigid set of algorithmic constraints run counter to the need for this creative exploration. While early investigations to better map the low-level neural net evaluations to stylistic effects are underway [Li et al. 2017], in our paper we focused on examining the higher-level parameter space for Neural Style Transfer and found a set of working shortcuts to map them to a reduced but meaningful set of creative controls.

2 Realizing Directorial Intent

Come Swim is a poetic, impressionistic portrait of a heartbroken man underwater. The film itself is grounded in a painting (figure 2 by coauthor Kristen Stewart) of man rousing from sleep. We take a novel artistic step by applying Neural Style Transfer to redraw key scenes in the movie in the style of the painting, realizing them almost literally painting that underpins the film.

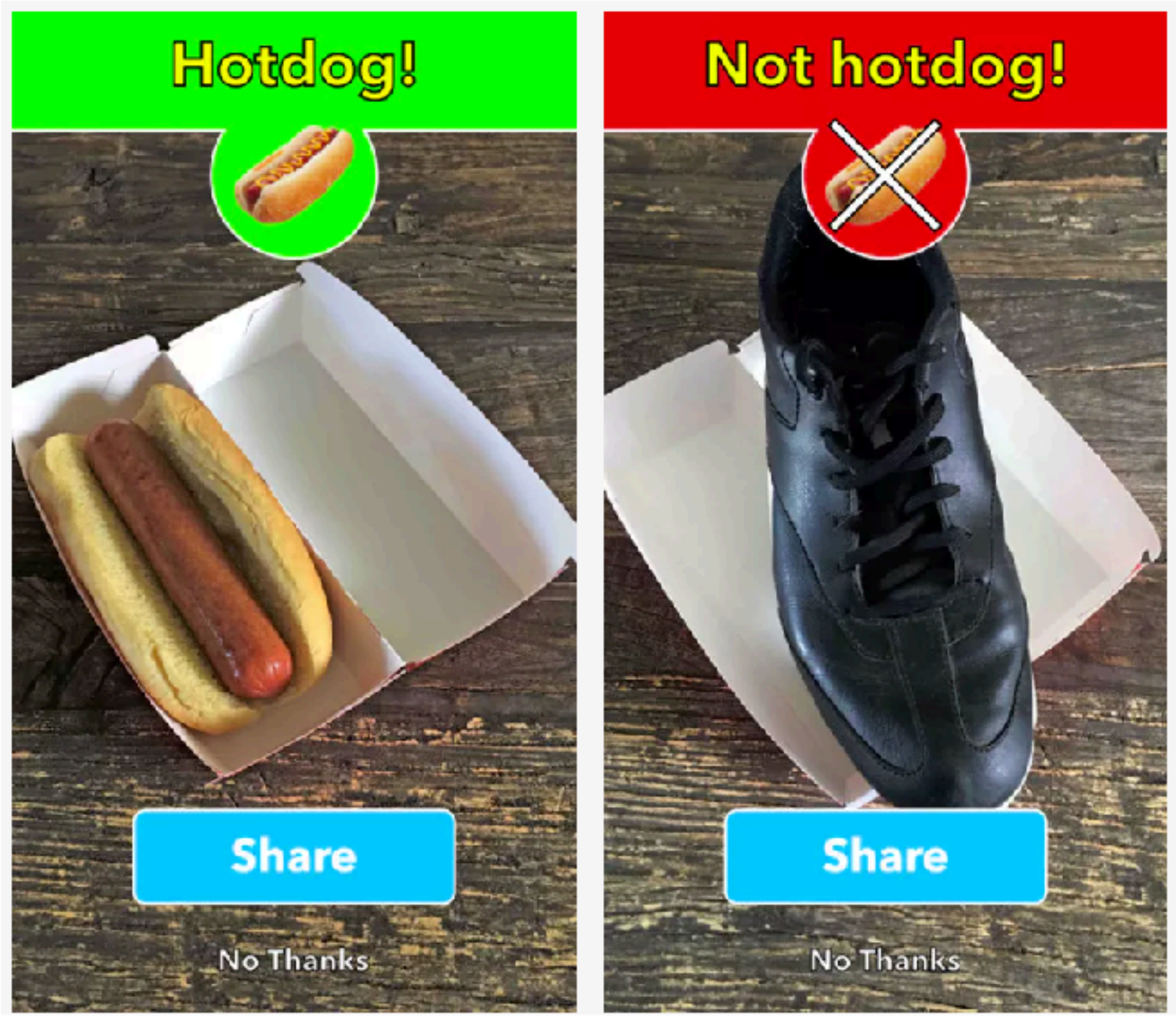
The painting itself evokes the thoughts an individual has in the first moments of waking (fading in-between dreams and reality), and this theme is explored in the introductory and final scenes where this technique is applied. This directly drove the look of the shot, leading us to map the emotions we wanted to evoke to parameters in the algorithm as well as making use of more conventional techniques in the 2D compositing stage.



Google Cloud APIs

Vision API

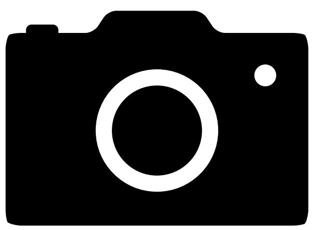
- Detecção de faces;
- Detecção de logomarcas;
- Detecção de labels;
- Detecção de texto;
- etc…

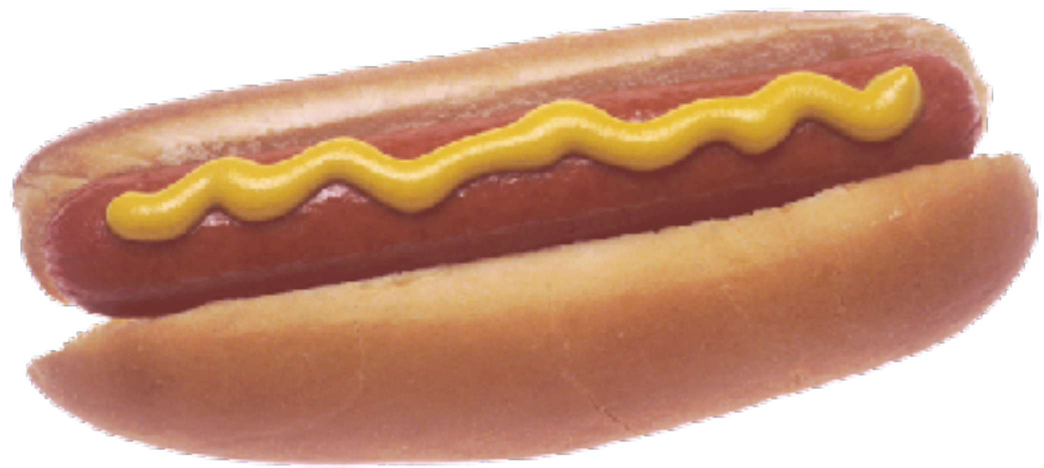

Hotdog!
Share
No Thanks
Not hotdog!
Share
No Thanks

```python
from google.cloud import vision
vision_client = vision.Client()

def detect_labels_uri(uri):
    image = vision_client.image(source_uri=uri)

    labels = image.detect_labels()
```

Input:

```
python vision-api.py labels-uri image-uri
```





Output:

```
Labels: frankfurter würstchen, hot dog, bockwurst, knackwurst, sausage, kielbasa, cervelat, german food, hot dog bun, chili dog
```

```python
def detect_faces_uri(uri):
    image = vision_client.image(source_uri=uri)

    faces = image.detect_faces()


def detect_text_uri(uri):
    image = vision_client.image(source_uri=uri)

    texts = image.detect_text()


def detect_logos_uri(uri):
    image = vision_client.image(source_uri=uri)

    logos = image.detect_logos()
```

Input:

```
python vision-api.py faces-uri image-uri
```

Output:

```
Faces:
anger: Likelihood.VERY_UNLIKELY
joy: Likelihood.VERY_LIKELY
surprise: Likelihood.VERY_UNLIKELY
face bounds: (794,98),(992,98),(992,328),
(794,328)
```

(Fonte: https://dribbble.com/shots/2707118-Google-Translate-Update)

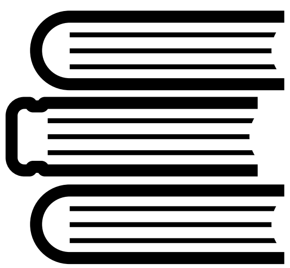
Natural Language API

- Análise de sentimento;
- Análise de entidades;
- Análise de sintaxe;

```python
def detect_sentiment(movie_review_filename):
    with open(movie_review_filename, 'r') as review_file:
        document = language_client.document_from_html(review_file.read())

        annotations = document.annotate_text(include_sentiment=True,
                                             include_syntax=False,
                                             include_entities=False)
```

Input:

```
python natural-language-api.py
sentiment sample.txt
```



sample.txt > “My hope is that the sequels are actual attempts at movies. The world doesn't need any more toothless cinema.”

Output:

```
Overall Sentiment: score of 0.2 with
magnitude of 0.4
```

```python
def detect_entities(text_snippet_filename):
    with open(text_snippet_filename, 'r') as snippet_file:
        document = language_client.document_from_html(snippet_file.read())

        annotations = document.annotate_text(include_sentiment=False,
                                             include_syntax=False,
                                             include_entities=True)
```

Input:

```
python natural-language-api.py entities
sample.txt
```

sample.txt > “My hope is that the sequels are actual attempts at movies. The world doesn't need any more toothless cinema.”

Output:

```
Entity: 0, name: hope, salience: 0.4828899
Entity: 1, name: sequels, salience: 0.17383586
Entity: 2, name: movies, salience: 0.13395432
Entity: 3, name: attempts, salience: 0.123159915
Entity: 4, name: world, salience: 0.05997689
Entity: 5, name: cinema, salience: 0.026183115
```

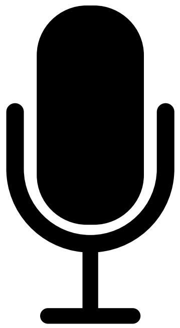
Speech API

- Reconhecimento síncrono;
- Reconhecimento assíncrono;
- Reconhecimento via streaming;

```python
def transcribe_speech_sync(speech_file):
    with io.open(speech_file, 'rb') as audio_file:
        content = audio_file.read()
        audio_sample = speech_client.sample(
            content=content,
            source_uri=None,
            encoding='LINEAR16',
            sample_rate_hertz=16000)

    alternatives = audio_sample.recognize('en-US')
```

Input:

```
python speech-api.py resources/
speech_sync_sample.raw
```



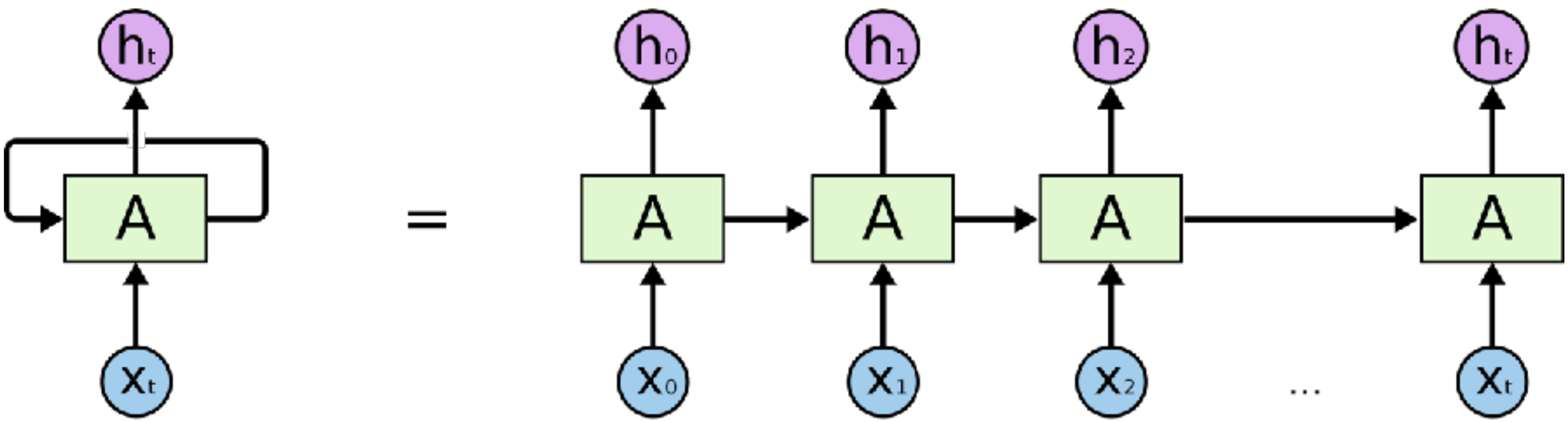
speech_sync_sample.raw > “how old is the Brooklyn Bridge”

Output:

```
Transcript: how old is the Brooklyn Bridge,
Confidence: 0.987628996372
```

Comprimindo imagens com Redes Neurais Recorrentes

Redes Neurais Recorrentes



(Fonte: http://colah.github.io/posts/2015-08-Understanding-LSTMs/)

Vamos comprimir imagens!

(Fonte: https://github.com/tensorflow/models/tree/master/compression)

Imagem original

Imagem comprimida

Imagem original

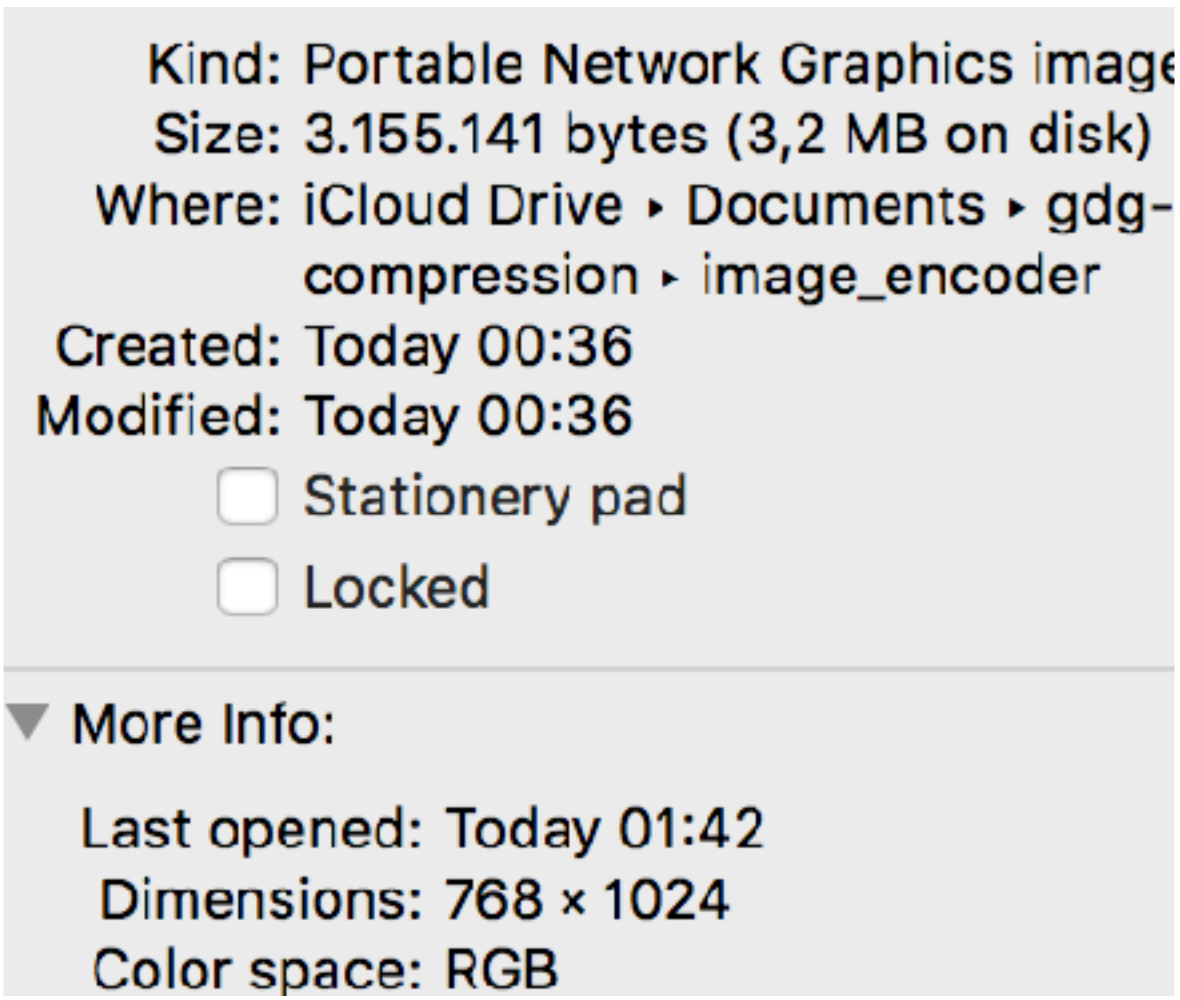
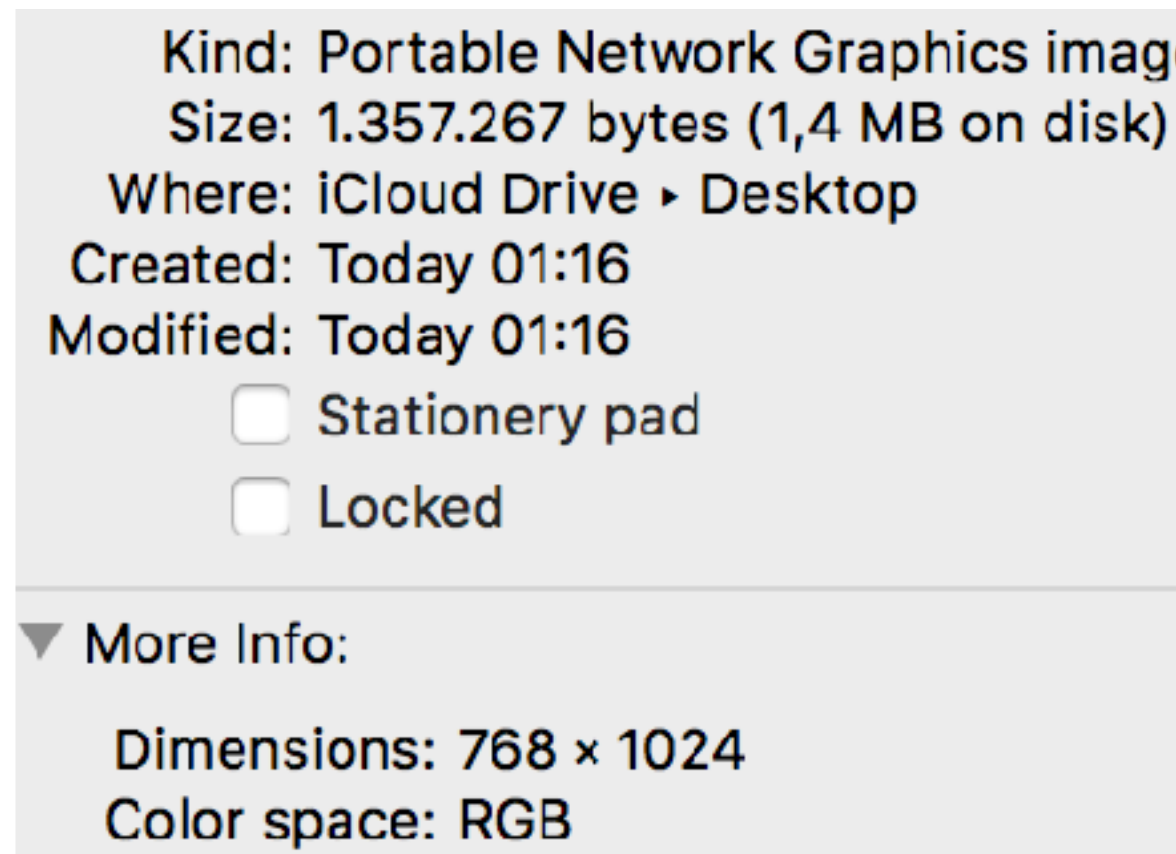


Imagem comprimida



Composição do Sistema

- Encoder (E);
- Binarizer (B);
- Decoder (D);

Residual Gated Recurrent Unit

Iteration	BPP	Compression Ratio
0	0.125	192:1
1	0.250	96:1
2	0.375	64:1
3	0.500	48:1
4	0.625	38.4:1
5	0.750	32:1
6	0.875	27.4:1
7	1.000	24:1
8	1.125	21.3:1
9	1.250	19.2:1
10	1.375	17.4:1
11	1.500	16:1
12	1.625	14.7:1
13	1.750	13.7:1
14	1.875	12.8:1
15	2.000	12:1


image_00.png


image_01.png

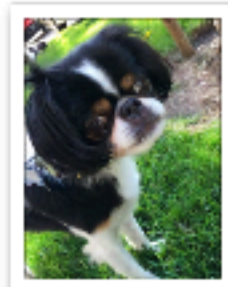


image_02.png


image_03.png

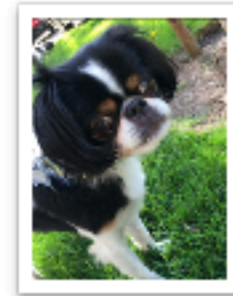


image_04.png


image_05.png


image_06.png


image_07.png


image_08.png


image_09.png


image_10.png

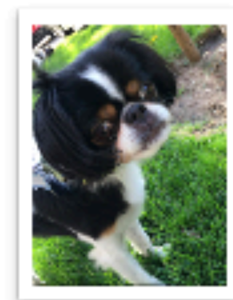


image_11.png


image_12.png


image_13.png


image_14.png


image_15.png


MakeAGIF.com

Verificando a similaridade entre as imagens:

- MM-SSIM (multi-space structural similarity) rate = 0.990523568071;

Obrigada. 🙂

https://olarclara.github.io