



UTOKYO

画像解析

<https://github.com/alpv95/MemeProject>
<https://github.com/affinelayer/pix2pix-tensorflow>

Finals in a month.
SHOULD I STUDY?

2

元々のアイデア

西洋の笑い画像「ミーム」

Nah, Komabasai is more fun.

悪意あるケルメット

旧@TodaiMemes (ツイッター)



For the record, the answer is no.

by giveuahint

4,616 views, 72 upvotes, 60 comments



これは鳩ですか？

- 昔の画像、広く知られている画像を使って、著作権の被害しないもの
- 流行っているもので、90年代初めの漫画とアニメ。
- アニメの4回目で主人公は蝶々を見て、「これは鳩ですか？」と聞く。
- 昨日ごろもアメリカで独立記念日で、上左りは毎年夜遅くまで外で音の多い花火やパーティある社会問題。

これは鳩ですか？（続き）

- 他にもアメリカの銃問題も扱っている。
- 今年5月にアメリカでリリースされた曲「これはアメリカ（の事実）」で、得にインパクト強いのは3－4回の大音な銃で人を殺すカット。
- 同じごろ以上の鳩画像「ミーム」は流行っていて、TwitterなどでPhotoshopが使える人達は右真ん中の画像を作る。
- 鳩や蝶々は平和なイメージしかないのに、どうしてこういう風になったのかも自分でも謎です。



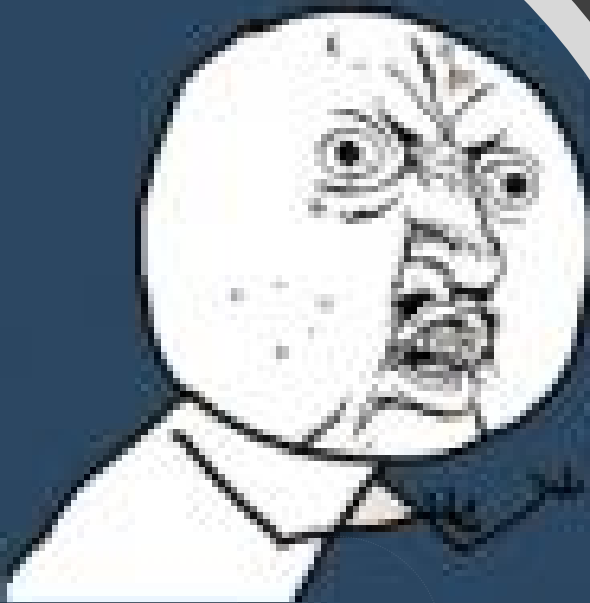


これはまたヒドイな！



やるべきことが不明で
現在、焦ってます。

5
すべてのAをBせよ！！



何で君は〜〜〜なの？！

他の有名なミーム

<https://imgflip.com/memetemplates>

私の彼氏は困った人で
他人を見ている。



Dank Learning: Generating Memes Using Neural Networks

Abel L. Peirson V

Department of Physics
Stanford University
alpv95@stanford.edu

E. Meltem Tolunay

Department of Electrical Engineering
Stanford University
meltem.tolunay@stanford.edu

Abstract

We introduce a novel meme generation system, which given any image can produce a humorous and relevant caption. Furthermore, the system can be conditioned on not only an image but also a user-defined label relating to the meme template, giving a handle to the user on meme content. The system uses a pre-trained Inception-v3 network to return an image embedding which is passed to an attention-based deep-layer LSTM model producing the caption - inspired by the widely recognized Show and Tell Model. We implement a modified beam search to encourage diversity in the captions. We evaluate the quality of our model using perplexity and human assessment on both the quality of memes generated and whether they can be differentiated from real ones. Our model produces original memes that cannot on the whole be differentiated from real ones. <https://github.com/alpv95/MemeProject>

1 Introduction

'A meme is an idea, behavior, or style that spreads from person to person within a culture, with the aim of conveying a particular phenomenon, theme, or meaning represented by a particular subject' [17], [18]

are ubiquitous in today's day and age; their language has become a part of almost every form of media, with new forms of media providing a platform for humor to be shared, until...

ミーム画像ではどう言う
Textが使われるか？

Stanford大学のNLP授業
で大笑いされた二人

<https://web.stanford.edu/class/cs224n/reports/6909159.pdf>

パソコン (TensorFlow) が 作った画像

- 最初にMemeGenerator.netから取った (Scrapeした) 画像と画像のHTML ALT TEXTの言葉 (Captions)。
- それをやるにはScraper.pyのPython フังก์ションを使った。
- Captions.txtや画像.jpgをTensorFlowの Evaluate.pyファンクションLibraryを使い、 configuration.pyやshow_and_tell_model.pyで言葉の教師あり分類学習させる。
- おしまいには、右の画像が成り立つ。

<https://github.com/alpv95/MemeProject>



これは残念。直せるかな？

ERROR 1

Scraper.pyのコードを載せてある物の通りに走らせようと

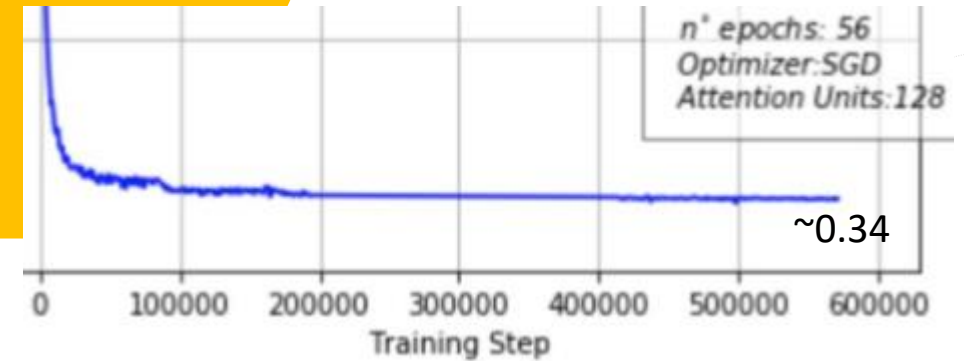
したら、エラーや注意だらけの使いにくいものであった。

- 1) Scraperを走らせたらず初に出るのはIndexとStringのタイプエラー：直すには半日かかった。
- 2) 次にHTTPのResponseが200出ないエラー。
これも調べたら、URLのRegexが間違っていた。
- 3) Websiteが変わっていた為、他のサイトで一から書き直し。
- 4) 一から書き直した今も、Pythonコードに間違いがある
ようで、出るが像の数が限られるやCaptionは28回も同じ
ものがファイルにプリントされている。

ERROR 2

```
C:\Users\MadIceTea\Documents\github-repos\MemeProject\im2txt>python scraper.py
Traceback (most recent call last):
  File "scraper.py", line 40, in <module>
    R = requests.get(URL)
  File "C:\Program Files\Python35\lib\site-packages\requests\api.py", line 72, in ge
t
    return request('get', url, params=params, **kwargs)
  File "C:\Program Files\Python35\lib\site-packages\requests\api.py", line 58, in re
quest
    return session.request(method=method, url=url, **kwargs)
  File "C:\Program Files\Python35\lib\site-packages\requests\sessions.py", line 512,
in request
    resp = self.send(prepare, **send_kwargs)
  File "C:\Program Files\Python35\lib\site-packages\requests\sessions.py", line 622,
in send
    r = adapter.send(request, **kwargs)
  File "C:\Program Files\Python35\lib\site-packages\requests\adapters.py", line 445,
in send
    timeout=timeout
  File "C:\Program Files\Python35\lib\site-packages\urllib3\connectionpool.py", line 600,
in urlopen
    chunked
  File "C:\Program Files\Python35\lib\site-packages\urllib3\connectionpool.py", line 343, in _ma
ke_request
    self._validate_conn(conn)
  File "C:\Program Files\Python35\lib\site-packages\urllib3\connectionpool.py", line
849, in _validate_conn
    conn.connect()
  File "C:\Program Files\Python35\lib\site-packages\urllib3\connection.py", line 356,
in connect
    ssl_context=context)
  File "C:\Program Files\Python35\lib\site-packages\urllib3\util\ssl_.py", line 359, i
n ssl_wrap_socket
    return context.wrap_socket(sock, server_hostname=server_hostname)
  File "C:\Program Files\Python35\lib\ssl.py", line 377, in wrap_socket
    _context=self)
  File "C:\Program Files\Python35\lib\ssl.py", line 752, in __init__
    self.do_handshake()
  File "C:\Program Files\Python35\lib\ssl.py", line 988, in do_handshake
    self._sslobj.do_handshake()
  File "C:\Program Files\Python35\lib\ssl.py", line 633, in do_handshake
    self._sslobj.do_handshake()
```

```
25 CELEBRATED 4TH OF JULY STILL HAVE ALL MY FINGERS
26 CELEBRATED 4TH OF JULY STILL HAVE ALL MY FINGERS
27 CELEBRATED 4TH OF JULY STILL HAVE ALL MY FINGERS
28 CELEBRATED 4TH OF JULY STILL HAVE ALL MY FINGERS
29 FORGETS TO TURN OFF THE AC IN THE CAR OPENS CAR DOOR
```

数学的な話

K -Nearest Neighbor (k -NN) 型神経回路で

k を使わず、上位100言葉の確率分布0-1のパラメータ

<https://arxiv.org/pdf/1806.04510.pdf>

$$f(p)_i = \frac{p_i^{1/T}}{\sum_j p_j^{1/T}}$$

Furthermore, we found that inference based on standard beam search, in which we keep k outputs in memory at each time-step, sequentially compute their "descendents" and then finally output the k sentences with the overall highest total probability scores, gives adequate but non-optimal results. In order to generate the freshest memes and diverse outputs for the same template we implement a temperature function into the beam search algorithm. Instead of selecting the top k most probable words, the k words are sampled from a probability distribution of the top 100 words, where the temperature of the distribution is a hyperparameter. A probability distribution p can be modified with temperature T by the function f where $T = 1$ corresponds to unchanged probabilities, high T lead to a very flat distribution (random pick) and low T leads to argmax (greedy search).



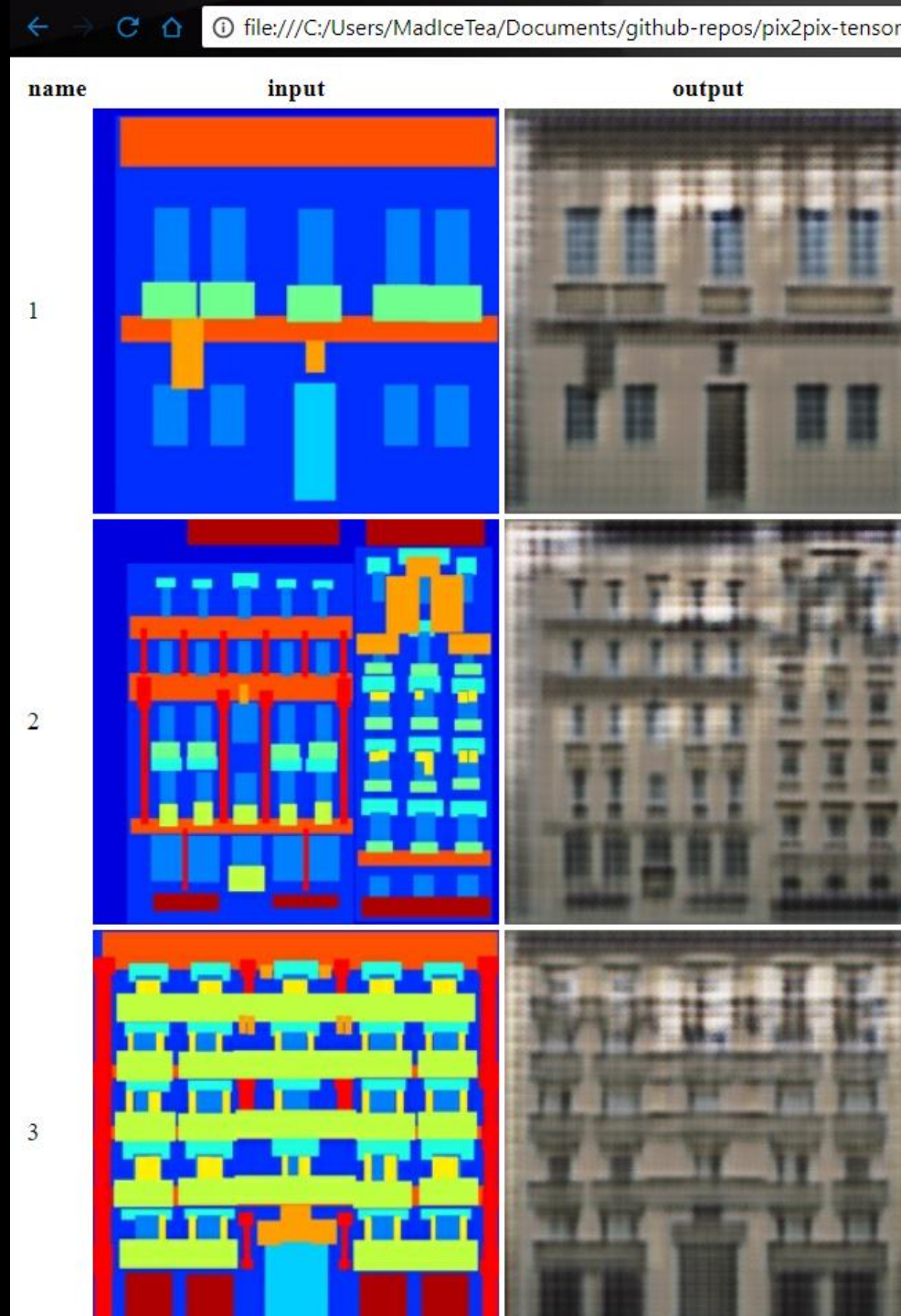
結局やったこと

2Dの変色画像から
3Dの写真を予想

何をやってますか？

- pix2pixとはTensorFlowを使って、RGB3色の多角形の図をビルの写真に予想により、相当していくpythonプログラム。
- 米加州、州立大学Berkeley校が持っているトレーニングセットをダウンロードする。
- pix2pix.pyをトレーニングモード(facades_train)で走らせる。最後のイメージが知られているので、このステップでは認識パターンのアルゴリズムが成立される。Epochs（時期数）も最初は1-2でやりましょう。
- 同じBerkeley校のデータにfacades_testのフォルダもある。その中の画像を走らせると自分のパソコンのアルゴリズムから作られる画像と本当の画像がHTMLページで比べられる。
- 時間があったら、時期数を増やして正確さを調整しよう。

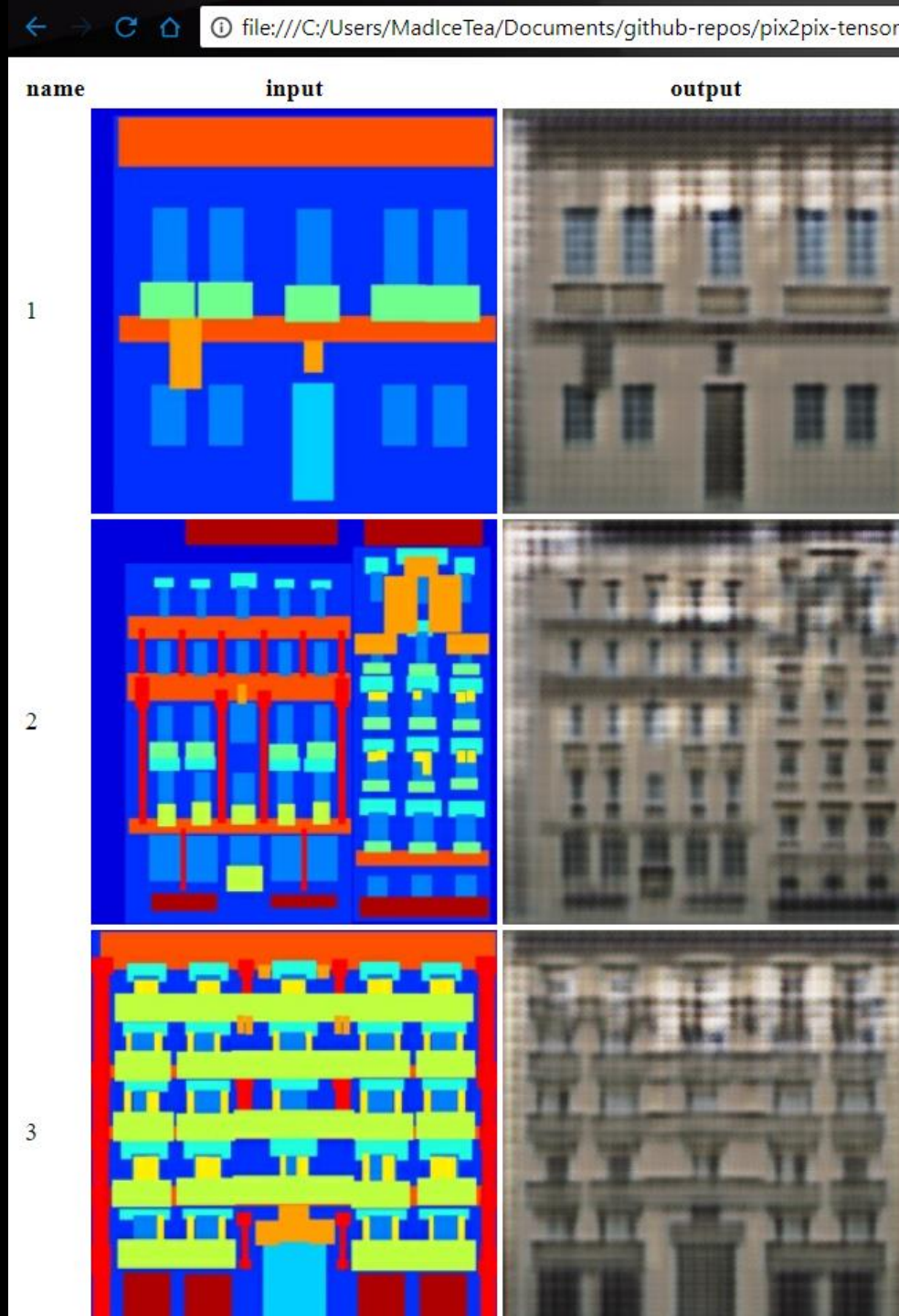
<https://github.com/affinelayer/pix2pix-tensorflow>



僕の設定

file:///pix2pix-tensorflow ¥ facades_train ¥ options.json

- *"aspect_ratio": 1.0,*
- *"batch_size": 1,*
- *"input_dir": "facades/train",*
- *"max_epochs": 2,*
- *"max_steps": null,*
- *"mode": "train",*
- *"output_dir": "facades_train",*
- *"output_filetype": "png",*
- *"summary_freq": 100*



Histogram mode

OVERLAY

OFFSET

Offset time axis

STEP

RELATIVE

WALL

Runs

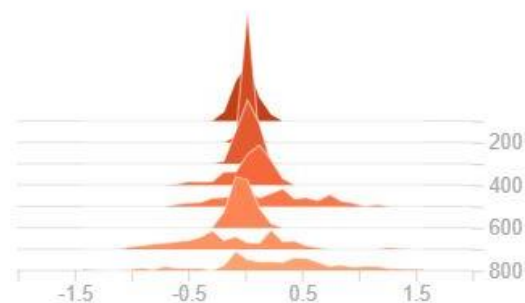
Write a regex to filter runs



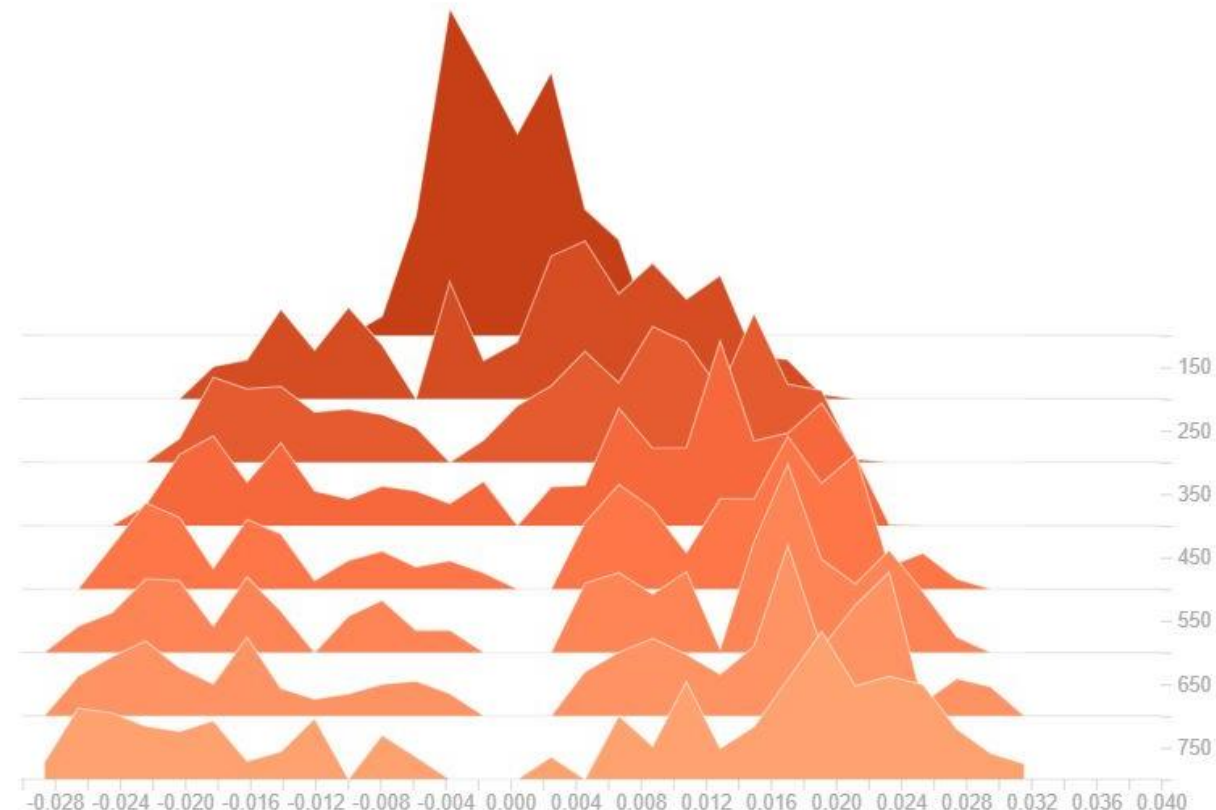
Q uncertainty

Tags matching /uncertainty/

discriminator/layer_1/conv2d/bias/gradients



discriminator/layer_1/conv2d/bias/values



使い方：
TENSORBOARD

グーグルのML Library, *TensorFlow*は2D や3D の分布図やそのCSVデータの発行できるプラットフォーム*TensorBoard*を内蔵されています。試しませんか？
Pythonのpipインストール時点でパソコンに入っています。

デフォルトとしてTensorFlowは
各100回しかデータを記録しない。

なので、800回（2時期）の右のデータは
短い方。だけど、1日とか走らせると
いい記録ができる可能性は高いです。

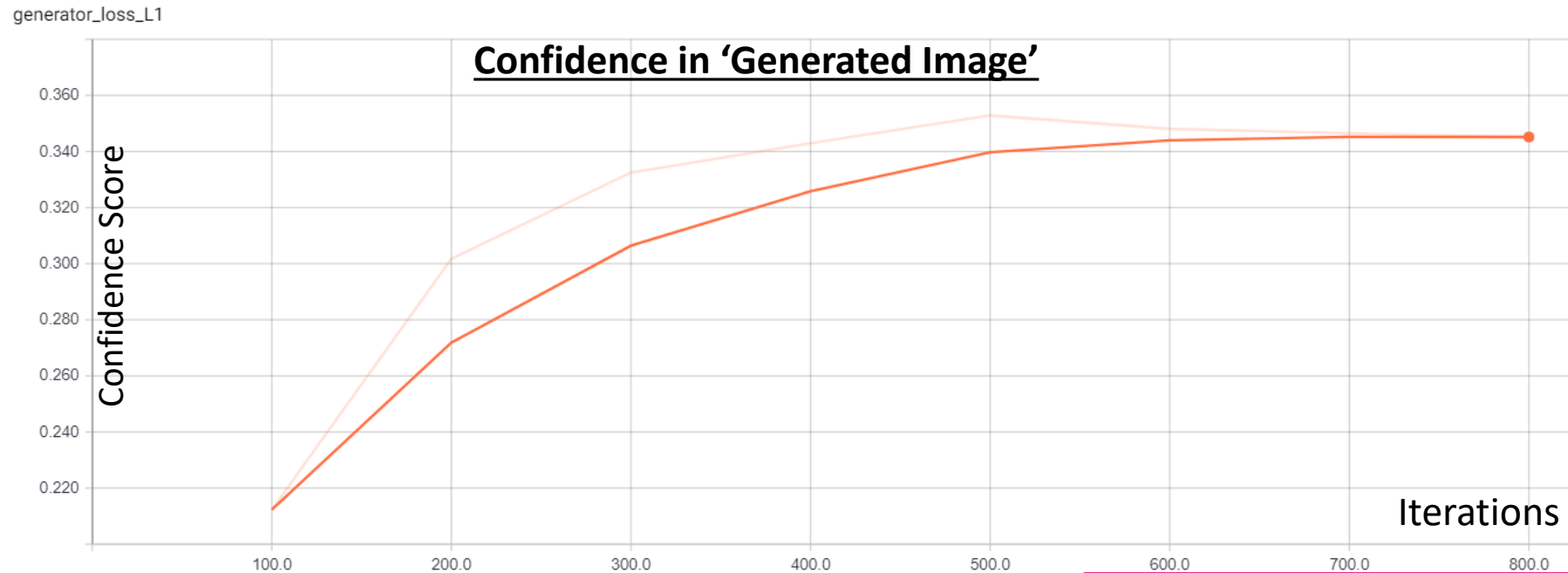
CSVはTensorBoardのGUIからでも、Pythonの
スクリプトからでも取りやすいものです。

Wall time	Step	Value		Wall time	Step	Value
1.53E+09	100	0.770427227		1.53E+09	100	0.212311491
1.53E+09	200	0.676876485		1.53E+09	200	0.301736474
1.53E+09	300	0.556499124		1.53E+09	300	0.332509786
1.53E+09	400	0.545846701		1.53E+09	400	0.342950135
1.53E+09	500	0.531756461		1.53E+09	500	0.35284102
1.53E+09	600	0.561052918		1.53E+09	600	0.348059833
1.53E+09	700	0.542005897		1.53E+09	700	0.346470386
1.53E+09	800	0.526081622		1.53E+09	800	0.345107168
Loss CSV データ				Generator CSV データ		

神経回路の シミュレーション結果

CSV、グラフ編





神経回路の
シミュレーション結果

作られたイメージに対し
ての正確さ

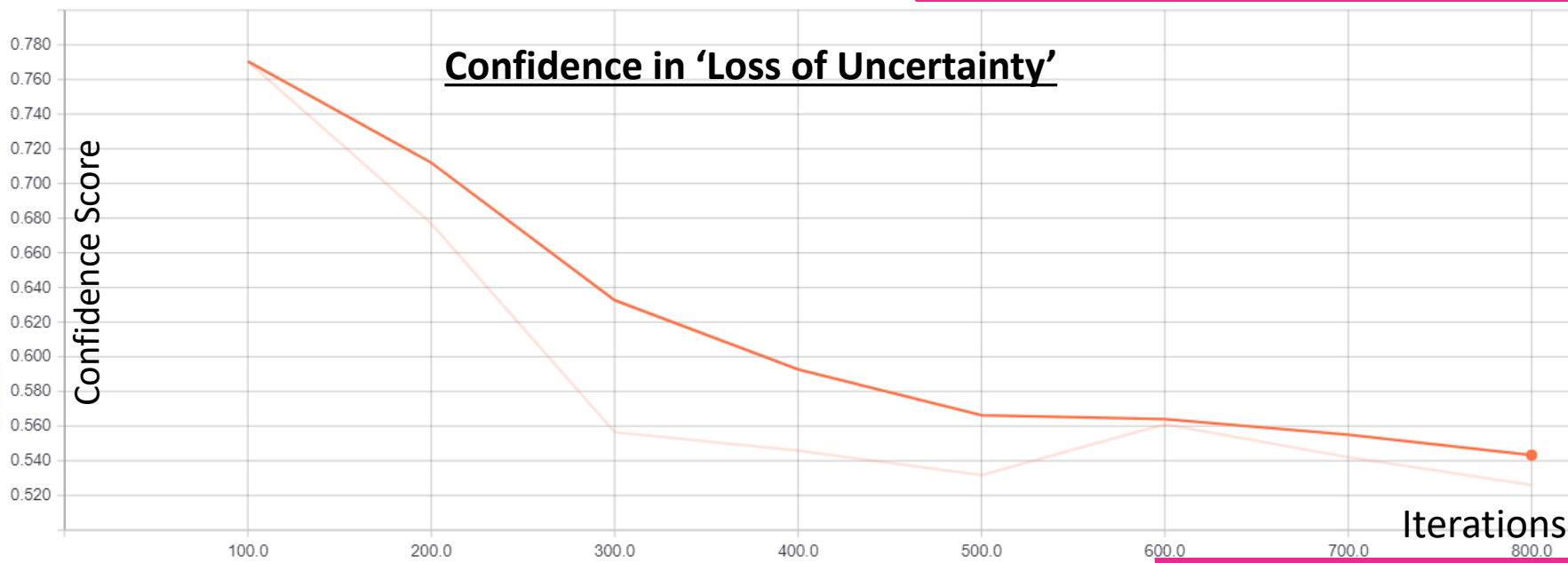
2時期しか使わずでもなんと約**1.5割**から
3.5割に分類できるピケセルが増えています。

将来も:

また2時期の神経回路トレーニングするば
モデルは~5割かそれ以上まで正確さを
あげられるだろうか?



神経回路の シミュレーション結果 元のイメージに対しての 非正確さ



本当のカーブは薄く書いてあるもので、上に載せているカーブは5割の正確さで回帰したものです。

元のイメージは8割のピクセルが分類の出来ないもので、パソコンが2時期・800回後は5割に減っている！

時期ごとに正確さがよくなるので、i5CPUのこのパソコンでも、一日（約17時期）で3割まで行けてもおかしくはないでしょう。

将来のProjectアイデア

<https://github.com/marcbelmont/cnn-watermark-removal>

- 最近見つけたのは、写真からWatermarkを神経回路でキレイに取り出す方法。
- 右上の写真は例えば、僕です。
この前の成人式からで、PreviewサイトからDeveloperツール使ってダウンロードした。
- その画像をWatermarkなしで、保存できるならたのましい。保存はしたいでも、見た目が結構悪いですね。



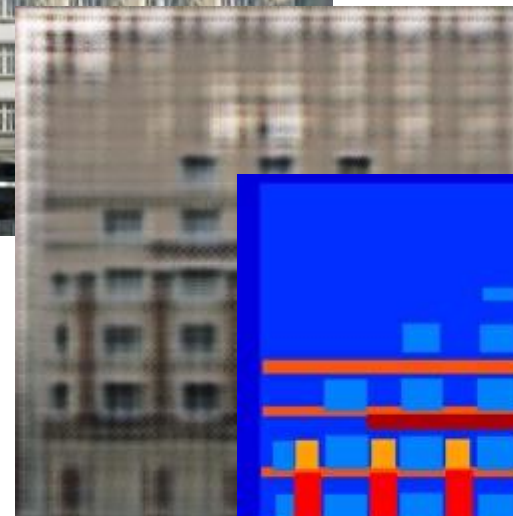
Hello
World!

ソフトを作ったベルmont氏から一つの例。



ご清聴ありがとうございました
何か質問等あればぜひ下さい。

Administrator: Command Prompt - tensorboard --logdir="C:\Users\MadIceTea\Docum
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.
C:\Windows\system32>pip install --upgrade tensorboard
Collecting tensorboard
Using cached https://files.pythonhosted.org/packages/9
p/tensorboard-1.9.0-py3-none-any.whl
Requirement not upgraded as not directly required: wheel
ite-packages (from tensorboard) (0.31.1)
Requirement not upgraded as not directly required: werkz
m tensorboard) (0.14.1)
Requirement not upgraded as not directly required: markd
tensorboard) (2.6.11)
Requirement not upgraded as not directly required: numpy



[Redacted]



[Redacted]



alex@g.ecc.u-tokyo.ac.jp

