

2017年5月30日

株式会社インプレスR&D

<http://nextpublishing.jp/>

低解像度画像を高解像度に変換する「超解像」をフルカラーで詳しく解説！

「TensorFlow はじめました2」発行

Google が提供する機械学習ライブラリのチュートリアル第二弾！

インプレスグループで電子出版事業を手がける株式会社インプレス R&D は、『TensorFlow はじめました2』（著者：有山圭二）を発行いたします。

『TensorFlowはじめました2』

<http://nextpublishing.jp/isbn/9784844397748>



著者：有山圭二

小売希望価格：電子書籍版 1000 円（税別）／印刷書籍版 1500 円（税別）

電子書籍版フォーマット：EPUB3／Kindle Format8

印刷書籍版仕様：B5 判／カラー／本文 82 ページ

ISBN：978-4-8443-9774-8

発行：インプレス R&D

<<本書の特徴>>

- (1) Google の機械学習ライブラリ「TensorFlow」のチュートリアルシリーズ第二弾！
- (2) 低解像度の画像を機械学習により高解像度に変換する「超解像」をフルカラーで詳しく解説
- (3) 技術書専門の同人誌即売会「技術書典」からの商業出版第二弾！

<<本書の内容>>

本書は Google が公開している機械学習ライブラリ「TensorFlow(テンソルフロー)」を初めて使う読者のためのチュートリアルガイドです。シリーズ2冊目となる本書では、低解像度の画像を機械学習をつかって高解像度に変換する「超解像」をテーマに、機械学習に初めて触れるエンジニアのための TensorFlow の基礎、実際に画像を使った機械学習に取り組むための初歩的な知識を掲載しています。

(本書は、次世代出版メソッド「NextPublishing」を使用し、出版されています。)

低解像度の画像を高解像度画像に変換する「超解像」の仕組みを解説

図2.7 超解像処理をした画像 (サイズ調整済み)



ご覧の通り、解像度を上げるところが、低解像度の画像すら再現できていません。次章では、いったい何が原因でこうなったのか。どのようにして解決するのかを検討します。

1 <https://arxiv.org/abs/1501.00022>
 2 https://en.wikipedia.org/wiki/Feature_extraction
 3 ブロックのサイズは、使用するモデルの定数INPUT_SIZEで定義されています。今回の場合は32x32になります
 4 これは、使用している超解像度アップスケーリングの仕様に依存するものです。PixelShuffleも超解像度を向上させる手段、入力データは (height, width) となりますが、model.pyの定義、Output_width (channel) と、変更が必要となります
 5 Softmaxと学習率は、2.9の1、3.9の1と、2.5の6の場合でも取り扱えるため超解像度入力層の調整がしやすいたと考えたためです

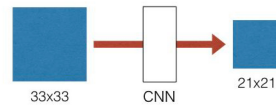
第3章 超解像奮闘記

第2章で作成したモデルは、期待した動作をしませんでした。本章では、失敗の原因を探り、どのようにすれば期待通りに動作するのか。その方法を探ります。

畳み込み処理とバディング

第2章のモデルは、3層の畳み込み層で構成されるCNNです。32pxの画像 (lr_image) を入力すると、21pxの画像を出力します。

図3.1 CNNの入出力



そもそもなぜ、入力と出力の大きさが異なるのでしょうか。それを知るには、モデルを構成している「畳み込み層」の処理に踏み込む必要があります。

畳み込み層

畳み込み層は、model_base.pyの関数conv2dで定義しています (リスト3.1)。

リスト3.1: srcnn/model/model_base.py

```
def conv2d(name, input_layer,
           weights_shape, weight_stddev,
           biases_shape, biases_value,
           strides=(1, 1, 1), padding='VALID'):
    with tf.variable_scope(name) as scope:
        weights = _get_weights(weights_shape, stddev=weight_stddev)
        conv = tf.nn.conv2d(input_layer, weights, strides,
                             padding=padding)
```

解像度の変化を比較しながら検証

図3.13 評価用画像 (拡大)
Low Resolution



Super Resolution



活性化関数

ノイズの原因を探るため、ブロックを再構築した直後のNumPy Arrayの最大値 (np.argmax) を確認すると、1.0以上の値が含まれていることがわかりました。

リスト3.8: result_imageの最大・最小値を表示

```
blocks = __extract_blocks(resized_image, stride)
blocks = __process(blocks, train_dir)
result_image = __reconstruct(blocks)

amax = np.argmax(result_image)
amin = np.argmin(result_image)
print('amax: %f, amin: %f' % (amax, amin))

result_image = result_image * 255
```

amax: 1.120654, amin: 0.000000

1.0を超える値に255を積算しているため、ピクセルの値域である0.255を超過した箇所が

ノイズとなっています。

では、なぜ1.0を超える値が出力されているのでしょうか。ここで、畳み込み層の活性化関数に注目します。

ReLU

現在のネットワークでは、すべての畳み込み層で活性化関数にReLU (Rectified Linear Unit) を使用しています (リスト3.9)。

リスト3.9: srcnn/model/model015.py

```
def inference(lr_images):
    # 省略

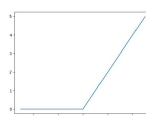
    conv3 = conv2d('conv3',
                  input_layer=conv2,
                  weights_shape=[5, 5, 32, CHANNELS],
                  weight_stddev=1e-3,
                  biases_shape=[CHANNELS], biases_value=0.0,
                  strides=[1, 1, 1, 1],
                  padding='VALID')
    conv3 = tf.nn.relu(conv3)

    return conv3
```

図3.14は、ReLUの値域を示すグラフです。

0以上の値をそのまま通すため、1.0を超える値でもそのまま最終層の出力になり、ノイズの原因になっています。

図3.14 Rectified Linear Unit



モノクロ画像だけでなくカラー画像の超解像も解説

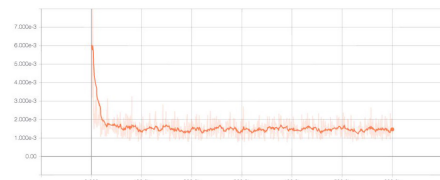
```
MODEL.INPUT_SIZE, MODEL.OUTPUT_SIZE,  
channels=MODEL.CHANNELS,  
scale=FLAGS.scale,  
batch_size=FLAGS.batch_size // 4)  
  
# 省略
```

図4.6は、表4.3の条件で学習した場合の誤差の変化です。カラー画像なので、学習に時間がかかると思つて60万ステップに設定しましたが、実際には、10万ステップ以降は大きな変化がありませんでした。

表4.3 学習条件

条件	値
画像の縮率 (scale)	2x
学習率 (learning_rate)	0.001
ミニバッチサイズ (batch_size)	128
ステップ数 (max_step)	600,000
チャンネル数 (CHANNELS)	3 (RGBカラー)

図4.6 誤差の変化 (smoothing=0.9)



評価用の画像として、縦横510pxのカラー画像を用意しました(図4.7)。

図4.7 評価用画像 (510x510, RGBカラー)



これをBICUBICで2倍に拡大したものを低解像度画像とします(図4.8)。

図4.8 低解像度画像 (1020x1020, RGBカラー)



図4.9は、評価用の画像を2倍に拡大後、学習済みのモデルを使って超解像処理をした結果です。

<< 目次 >>

TensorFlow の基礎

TensorFlow とは

データフローグラフ

Tensor(テンソル)

変数とブレースホルダー

演算子のオーバーロード

ブロードキャスト

CNN で超解像

超解像とは

モデルの定義

学習

画像処理

評価

超解像奮闘記

畳み込み層とバデイング

画像の読み込み処理

活性化関数

さまざまなモデル

モデル(9-5-5)

画質の指標(PSNR, SSIM)

Batch Normalization の導入

<<著者紹介>>

有山 圭二(ありやま けいじ)

有限会社シーリス代表。Android アプリケーションの受託開発や、Android に関するコンサルティング業務の傍ら、技術系月刊誌への記事執筆。最近では趣味で機械学習(ディープラーニング)に取り組んでいる。著書として「Android Studio ではじめる 簡単 Android アプリ開発(技術評論社刊)」「TensorFlow はじめました(インプレス R&D 刊)」など。

<<販売ストア>>

電子書籍:

Amazon Kindle ストア、楽天 kobo イーブックストア、Apple iBookstore、紀伊國屋書店 Kinoppy、Google Play Store、honto 電子書籍ストア、Sony Reader Store、BookLive!、BOOK☆WALKER

印刷書籍:

Amazon.co.jp、三省堂書店オンデマンド、honto ネットストア、楽天ブックス

※ 各ストアでの販売は準備が整いしだい開始されます。

※ 全国の一般書店からもご注文いただけます。

【株式会社インプレス R&D】 <http://nextpublishing.jp/>

株式会社インプレス R&D (本社: 東京都千代田区、代表取締役社長: 井芹昌信) は、デジタルファーストの次世代型電子出版プラットフォーム「NextPublishing」を運営する企業です。また自らも、NextPublishing を使った「インターネット白書」の出版など IT 関連メディア事業を展開しています。

※NextPublishing は、インプレス R&D が開発した電子出版プラットフォーム(またはメソッド)の名称です。電子書籍と印刷書籍の同時制作、プリント・オンデマンド(POD)による品切れ解消などの伝統的出版の課題を解決しています。これにより、伝統的出版では経済的に困難な多品種少数数の出版を可能にし、優秀な個人や組織が持つ多様な知の流通を目指しています。

【インプレスグループ】 <http://www.impressholdings.com/>

株式会社インプレスホールディングス(本社: 東京都千代田区、代表取締役: 唐島夏生、証券コード: 東証1部 9479)を
持株会社とするメディアグループ。「IT」「音楽」「デザイン」「山岳・自然」「モバイルサービス」を主要テーマに専門性の高いコンテンツ+サービスを提供するメディア事業を展開しています。

【お問い合わせ先】

株式会社インプレス R&D NextPublishing センター

〒101-0051 東京都千代田区神田神保町1-105

TEL 03-6837-4820

電子メール: np-info@impress.co.jp