

平成 29 年度 特別研究報告書

SNS 投稿画像に付けるハッシュタグ
を予測するプログラムの検討

龍谷大学 理工学部 情報メディア学科

T140446 小野勝也

指導教員 三好 力 教授

内容概要

本研究では、趣味だけでなく商品の紹介や販売と言ったビジネスとしても利用されているインスタグラムなどの SNS に投稿する画像を多くの利用者に見てもらうために、インスタグラムなどの SNS でよく利用されている人気ハッシュタグリストから投稿画像に適した人気ハッシュタグを付けるためのシステムを提案する。また、提案システムに沿って自作したプログラムが入力画像に適した人気ハッシュタグを予測できるか確認を行うため、画像認識の正解率およびハッシュタグの関連付けができているかの実験とインスタグラムへの投稿画像を入力し、入力画像に適したハッシュタグの予測、予測した結果と予測した人気ハッシュタグと同時によく使われている人気ハッシュタグを表示させることができるかの実験を行う。

目次

第1章	はじめに.....	4
第2章	既存技術.....	6
2.1	clarifai.....	6
2.2	TAGGER.....	7
2.3	ハッシュレコ.....	8
2.4	ハッシュタグ手帳.....	9
2.5	既存技術の問題点.....	10
第3章	提案システム.....	11
第4章	実験.....	12
4.1	実験1.....	12
4.1.1	システム.....	12
4.1.2	実験方法.....	12
4.1.3	実験1 結果.....	13
4.2	実験2.....	17
4.2.1	システム.....	17
4.2.2	実験方法.....	17
4.2.3	実験2 結果.....	18
4.3	考察.....	19
第5章	まとめ.....	20
謝辞	21
参考文献	22
付録	23

第1章 はじめに

SNSとは、ソーシャル・ネット・ワーキングの略称で、web上で社会的ネットワークを構築可能にするサービスである。今日、SNSは増々発展しておりそれと同様に、利用者也増加している。SNSは国や性別、年齢が様々な人々とネットの世界でつながることができる。

インスタグラムとは、無料の写真共有アプリケーションソフトウェアである。画像や動画を投稿することを主としている。画像投稿の際にハッシュタグをつけることでより多くの人に見てもらえることができる。また利用者は、気に入った画像に「いいね」と言うものをつけることができ、画像投稿者はその「いいね」をもらうため投稿画像に力を入れる。利用者也様々で旅行で撮った写真を投稿する人や作った料理の画像を投稿する人、プロやアマチュアカメラマン、デザイナーなどが自分の作品を知ってもらうために画像を投稿するなど様々である。最近では、販売店が人気商品や入荷商品の画像を投稿するなど広告として利用していることもある。

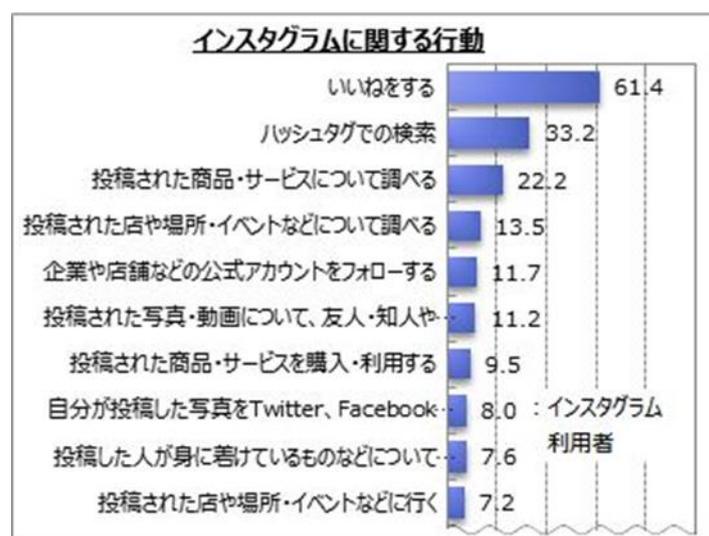


図1:Instagramに関する行動

図1からわかるように他人の投稿を閲覧し「いいね」をするだけでなく、商品やサービスの販売促進、店やイベントの宣伝などビジネスとしても役立っていることが分かる。参考文献[1]参照。

インスタグラムでの画像投稿は、趣味や「いいね」をもらうためだけでなく、商品の紹介や販売と言ったビジネスとしても利用されている。それを多くの利用者に見てもらうためには、インスタグラム内での人気ハッシュタグを付ける事やハッシュタグを多くつける事が必要である。しかし SNS 利用者も今人気のハッシュタグをすべて把握することは難しい。またハッシュタグは、多くつければつけるほど手間であり画像投稿の際、「いいね」は欲しいが、ハッシュタグ付けは面倒と感ずることが多々ある。

画像認識を用いた研究やサービスは多くあるが、インスタグラムなど SNS に特化した画像認識を用いたサービスはまだ少ない。そこで、インスタグラムで画像投稿を行う際に必要なハッシュタグ付けをサポートするプログラム作成を考えた。

第2章 既存技術

2.1 clarifai

clarifai は画像にタグ付けする API を提供しているサービスである。この API はユーザー登録をすることによって誰でも利用することができる。

プランは無料、有料があり、無料プランで試す場合にもクレジットカード情報の登録が必要。

デモ

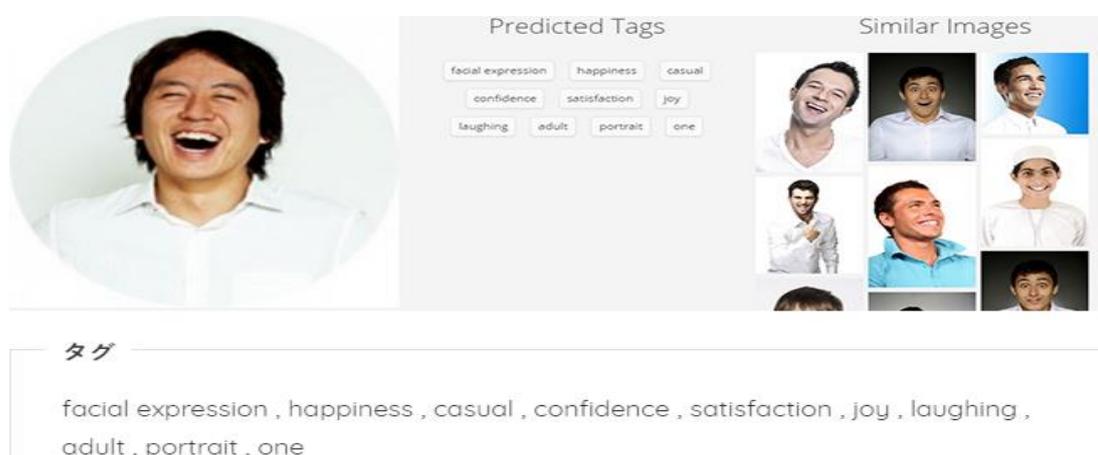


図 2:clarifai デモ 1

図 2 のように笑顔の男性を読み込むと、**facial expression**（表情）、**confidence**（自信）、**satisfacti**(満足)などと認識されている。

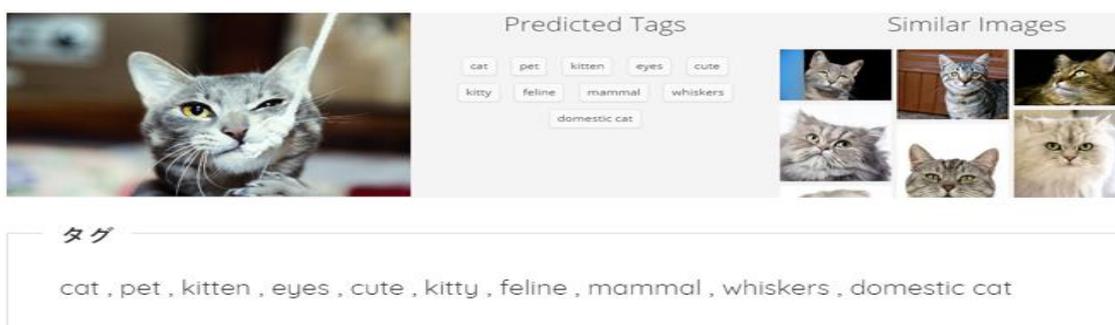


図 3:clarifai デモ 2

図 3 のように 猫。cat（猫）、kitten（子猫）など、人間以外の生物だと動物名のタグが付く。

2.2 TAGGER

TAGGER は AI（人工知能）で画像を分析して、最適なタグを表示。利用者は、コピー&ペーストするだけでタグ付けができる。



図 4:TAGGER デモ

図 4 のようにピンクのアイスクリームの画像を読み込むとタグの予想が下に表示される。予測されたタグの中から好きなものを選び、コピー&ペーストを行い、インスタグラム等で利用することができる。

2.3 ハッシュレコ

まず、キーワードを入力して検索する。
インスタグラムで投稿したい写真（動画）を表すキーワードを入力して虫眼鏡ボタンを押す。

（例）犬を撮影した写真を投稿する場合は、「犬」や「dog」、「チワワ」など関係性のあるキーワードを入力する。

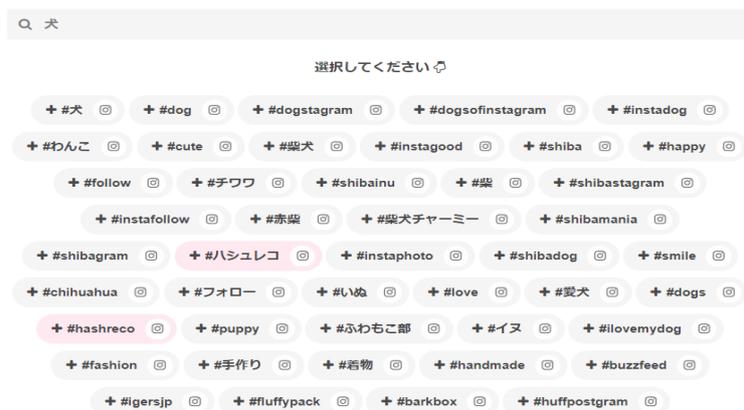


図 5:ハッシュレコ デモ 1

次に、ハッシュタグを選択する。結果が表示されたら、図 5 に表示されている+のマークをクリックすることで、図 6 の下のボックスに追加されていく。

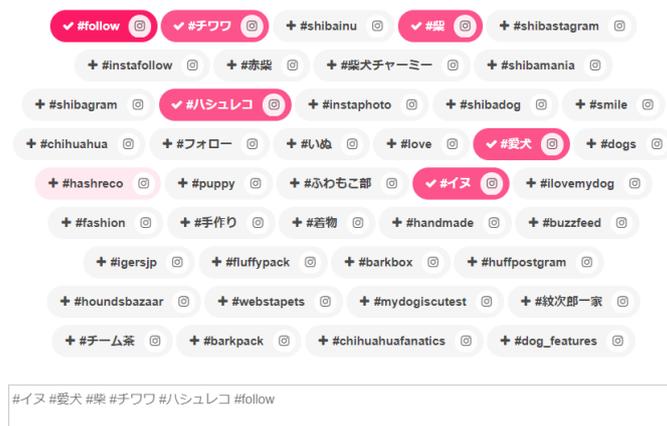


図 6:ハッシュレコ デモ 2

次に、貼り付けを行う。図 6 の下のボックスに出てきたハッシュタグをコピーして、インスタグラムの投稿画面で貼り付ける。

2.4 ハッシュタグ手帳

ハッシュタグ手帳は、自分のよく使うハッシュタグを登録しておくことで、画像を投稿する際に、コピー&ペーストですぐにハッシュタグ付けができる。

また、虫眼鏡のマークを押すことでジャンル別によく使われているハッシュタグを探し、マイハッシュタグに登録しておくことで投稿の際にすぐ使うことができる。



図 7:ハッシュタグ手帳デモ

2.5 既存技術の問題点

既存手法の多くは、画像認識を用いて認識結果をハッシュタグ候補として表示するものや、利用者が投稿画像から考えられるハッシュタグ候補を入力すると、それに関連するハッシュタグを多数表示するものである。clarifai や TAGGER では、画像認識の正解率が高いが人気ハッシュタグを予測することができない。

また、ハッシュレコやハッシュタグ手帳では、投稿画像に関連するハッシュタグを利用者が1つ文字入力することで、関連するハッシュタグが出てくる。ハッシュタグ手帳では、人気ハッシュタグも候補に出てくる。しかし、画像認識では無く、文字入力しなければならないこと。候補が多く人気ハッシュタグがどれかは利用者が知ってなければいけない。

第3章 提案システム

既存技術の問題点を解決し、インスタグラムなどの SNS でよく利用されている人気ハッシュタグリストから投稿画像に適した人気ハッシュタグを付けるためのシステムを提案する。

参考文献[6]「Instagram ジャンル別 139 選人気ハッシュタグ」に掲載されているようにインスタグラムで使われている人気ハッシュタグは多くあるので投稿画像に関連のある人気ハッシュタグだけを画像認識を用いてつける事ができるようにしたいと考えている。既存技術では、犬の画像なら犬の画像グループ、猫の画像なら猫の画像グループに分類して学習画像を集めているが、今回は、インスタグラム内で人気ハッシュタグ検索し、それら人気ハッシュタグを付けて投稿されている画像を人気ハッシュタグごとの画像グループにそれぞれ手動で集め、各画像グループに分類するという方法を試みる。そして、人気ハッシュタグを投稿画像に付けるために、投稿画像を入力するとその入力画像に関連のある人気ハッシュタグ候補のみを表示させるアルゴリズムを考えた。これを利用することで、利用者は今どのハッシュタグが人気なのか分からなくても、投稿画像に関連する人気ハッシュタグを付けることができ、投稿画像をより多くの人に見てもらえるだろう。

参考文献[6]のジャンル別人気ハッシュタググループを参考に人気ハッシュタググループとそれに対応した人気ハッシュタググループごと画像グループを作成する。

学習画像は、インスタグラム内で人気ハッシュタグを検索し、それら人気ハッシュタグを付けて投稿されている画像を人気ハッシュタグごとの画像グループにそれぞれ手動で集め、各画像グループに分類する。

keras などのニューラルネットワークの実装が可能なライブラリーを利用して、機械学習により学習画像と対応する人気ハッシュタググループを出力する識別機を教師あり学習により作成する。

マッチ度の高い順に複数個人気ハッシュタググループを選択する。

これに投稿画像を入力するとその入力画像に適した人気ハッシュタグが出力されるシステムとなる。

第4章 実験

4.1 実験 1

実験 1 では、提案システムで考えたアルゴリズムを参考に作成したプログラムで画像認識の正解率および人気ハッシュタグの関連付けができているかの確認を行う。

4.1.1 システム

参考文献[6]で掲載されていたカテゴリー別人気ハッシュタグからファッションやインテリアといったカテゴリー別人気ハッシュタグ 7 つを抜粋し、それに対応した 7 つの画像グループを作成した。インスタグラムに投稿されている画像の内 7 つの抜粋した人気ハッシュタグを付けて投稿されている画像を検索し、各グループの教師データとするため、各グループの学習画像 70 枚、テスト画像 5 枚を、`snipping tool` を利用し画像を集めた。

(例)「インテリア」の画像を集める場合は、ハッシュタグ「インテリア」を付けた投稿画像を学習画像として利用する。

`snipping tool` とは、ディスプレイに表示されている画面の好きな範囲をトリミングし画像として保存できるアプリケーションである。

上記データを用いて、プログラムは、「Keras による、ものすごくシンプルな画像分類 (りんごとオレンジ)」(参考文献[7]参照)に掲載されているプログラムを参考に、画像認識の正解率を上げるためニューラルネットワークに層を追加し、入力が画像で出力が 7 つのグループの予測値となるようニューラルネットワークで学習し、予測結果に全画像グループである人気ハッシュタグの予想確率を数値で確認できるようにした。

画像は、まず画像 **R**、**G**、**B** の 3 つの情報を含む 2 次元配列として読み込む。これを {(RED の配列)、(GREEN の配列)、(BLUE の配列)} の形に変換する。次に、ニューラルネットワークで学習させるため `numpy` 配列に変換し学習させる。

各グループの学習画像を学習させた後、テスト画像各 5 枚を入力画像として予測確率を表示させ、それぞれ一番近いグループを結果とした。

4.1.2 実験方法

今回の実験では、インスタグラムに投稿される実際の画像サイズは 1080×1080 pixel であるがインスタグラムで以前採用されていた画像サイズ 612×612 pixel と現在採用されている画像サイズ 10 分の 1 の 108×108 pixel の 2 パタ

ーン、学習回数は 100、500、1000 の 3 パターン、一回のオペレーションで処理する量であるバッチサイズは 25 の 1 パターンをそれぞれ行うので実験は 6 パターン行う。そして、最も画像認識の正解率の高い結果となった画像サイズ、学習回数、バッチサイズの数値を実験 2 で採用する。実際の画像サイズである 1080×1080pixel では 1 回の実験に時間がかかってしまう上、精度も 612×612pixel の時と差がない結果であったので今回は省略した。

今回の実験では、投稿画像の予測が間違っていたとしても画像内に他のハッシュタグの要素が含まれており、その少し含まれた他の要素から予測結果を表示していてもハッシュタグとの関連付けはできているとする。これを以下に示す。



図 8:ハッシュタグ「ペット」で投稿された画像

図 8 のように服を着た犬が 2 匹写っている画像を入力した場合、「ペット」だけでなく、「ファッション」、「自然」と予測されても正解とする。理由は、犬が服を着ており「ファッション」に関連していることや芝生のある場所で撮影されており「自然」に関連しているからである。

4.1.3 実験 1 結果

実験 1 の結果では画像サイズ、学習回数、バッチサイズの数値と画像認識の正解率及び画像認識で正解ではなくてもハッシュタグの関連付けはできている画像の枚数、これと画像認識の正解率を合わせた正解率を示す。

(1) 画像サイズ:612×612pixel

学習回数:100

バッチサイズ:25

表 1:(1)の各グループの正解率と総合正解率

label(正解)	0	1	2	3	4	5	6	
画像グループ	インテリア	ファッション	フィットネス	ペット	自然	自撮り	料理	総合
正解率	80.0%	80.0%	60.0%	40.0%	40.0%	40.0%	100.0%	62.9%

画像認識において正解ではないがハッシュタグの関連付けはできている画像:5枚

画像認識において正解ではないがハッシュタグの関連付けはできている画像と画像認識の正解率を合わせた正解率:77.1%

(1)の実験結果で画像認識において正解ではなくてもハッシュタグの関連付けはできている画像 5 枚を入力した時の出力結果を図 9 に示す。

```
C:\Users\masaya1\insta\test\ファッション\4.JPG
1/1 [=====] - 0s
label: 1 result: 5
[[ 5.43214055e-03  3.32623035e-01  5.28588817e-02  2.15941225e-03
   1.71693348e-04  6.06540620e-01  2.14247470e-04]]

C:\Users\masaya1\insta\test\フィットネス\2.JPG
1/1 [=====] - 0s
label: 2 result: 5
[[ 4.38494812e-04  5.33160273e-05  6.67544156e-02  3.86615975e-05
   3.15313664e-05  9.28618133e-01  4.06548986e-03]]

C:\Users\masaya1\insta\test\フィットネス\4.JPG
1/1 [=====] - 0s
label: 2 result: 1
[[ 1.75036443e-03  9.76552427e-01  2.70197938e-06  6.16838443e-05
   4.45830665e-05  1.48908943e-02  6.69732736e-03]]

C:\Users\masaya1\insta\test\自撮り\zi1.JPG
1/1 [=====] - 0s
label: 5 result: 1
[[ 5.47080487e-03  9.13313866e-01  6.11372665e-02  7.44379638e-03
   4.11916478e-03  8.48620292e-03  2.88194915e-05]]

C:\Users\masaya1\insta\test\自撮り\zi2.JPG
1/1 [=====] - 0s
label: 5 result: 1
[[ 2.61710375e-04  9.90825951e-01  1.38845076e-06  8.51900841e-04
   7.57438647e-06  7.99433142e-03  5.71595083e-05]]
```

図 9: 画像認識において正解ではないがハッシュタグの関連付けはできている 5 枚の画像を入力した時の出力結果

図9には、上からテスト画像名およびその画像の場所、画像認識の残り処理時間、label(正解ラベル)、result(画像認識後の出力結果)、7つのlabelの予測確率を画像5枚それぞれ示している。

図9では、ファッション/4.jpgの正解は1である。しかし、この画像は自撮りされた写真でありlabel5の自撮りと予測しても正解としている。フィットネス/2.jpg以降も同様の理由でハッシュタグの関連付けはできているとして正解としている。



図10:ファッション/4.jpgの画像

(2)画像サイズ:612×612pixel

学習回数:500

バッチサイズ:25

表2:(2)の各グループの正解率と総合正解率

label(正解)	0	1	2	3	4	5	6	
画像グループ	インテリア	ファッション	フィットネス	ペット	自然	自撮り	料理	総合
正解率	80.0%	80.0%	40.0%	20.0%	40.0%	40.0%	80.0%	54.3%

画像認識において正解ではないがハッシュタグの関連付けはできている画像:3枚

画像認識において正解ではないがハッシュタグの関連付けはできている画像と画像認識の正解率を合わせた正解率:60.0%

(3)画像サイズ:612×612pixel

学習回数:1000

バッチサイズ:25

表 3:(3)の各グループの正解率と総合正解率

label(正解)	0	1	2	3	4	5	6	
画像グループ	インテリア	ファッション	フィットネス	ペット	自然	自撮り	料理	総合
正解率	80.0%	60.0%	20.0%	60.0%	40.0%	40.0%	60.0%	51.4%

画像認識において正解ではないがハッシュタグの関連付けはできている画像:4枚

画像認識において正解ではないがハッシュタグの関連付けはできている画像と画像認識の正解率を合わせた正解率:62.9%

(4)画像サイズ:108×108pixel

学習回数:100

バッチサイズ:25

表 4: (4)の各グループの正解率と総合正解率

label(正解)	0	1	2	3	4	5	6	
画像グループ	インテリア	ファッション	フィットネス	ペット	自然	自撮り	料理	総合
正解率	80.0%	60.0%	20.0%	20.0%	60.0%	40.0%	40.0%	45.7%

画像認識において正解ではないがハッシュタグの関連付けはできている画像:4枚

画像認識において正解ではないがハッシュタグの関連付けはできている画像と画像認識の正解率を合わせた正解率:57.1%

(5)画像サイズ:108×108pixel

学習回数:500

バッチサイズ:25

表 5: (5)の各グループの正解率と総合正解率

label(正解)	0	1	2	3	4	5	6	
画像グループ	インテリア	ファッション	フィットネス	ペット	自然	自撮り	料理	総合
正解率	80.0%	100.0%	20.0%	20.0%	40.0%	40.0%	40.0%	45.7%

画像認識において正解ではないがハッシュタグの関連付けはできている画像:3枚

画像認識において正解ではないがハッシュタグの関連付けはできている画像と画像認識の正解率を合わせた正解率:54.3%

(6)画像サイズ:108×108pixel

学習回数:1000

バッチサイズ:25

表 6: (6)の各グループの正解率と総合正解率

label(正解)	0	1	2	3	4	5	6	
画像グループ	インテリア	ファッション	フィットネス	ペット	自然	自撮り	料理	総合
正解率	80.0%	60.0%	0.0%	40.0%	20.0%	40.0%	40.0%	40.0%

画像認識において正解ではないがハッシュタグの関連付けはできている画像:3枚

画像認識において正解ではないがハッシュタグの関連付けはできている画像と画像認識の正解率を合わせた正解率:48.6%

正解率は小数点第一位まで表示する。

4.2 実験 2

実験 2 ではシステムに投稿画像を入力すると、入力画像は 7 つのグループの内どのグループに予測され、予測した結果とそれに関連するハッシュタグを表示させることができるかの確認を行う。

4.2.1 システム

プログラムは、実験 1 と学習までは同じものを使用し 7 つの人気ハッシュタグと同時によく使われている人気ハッシュタグ（参考文献[6]参照）リストを作成し 7 つのそれぞれのグループに関連付けした。そして入力画像から 7 つの人気ハッシュタグのどれに近いかに予測し出力する。次の行に参考文献[6]を参照して作成した 7 つの人気ハッシュタグと同時によく使われている人気ハッシュタグリストの内を 1 行目で予測された結果に関連付けした人気ハッシュタグリストを出力する。

4.2.2 実験方法

実験 1 で画像認識の正解率が最も高かった画像サイズ、学習回数、バッチサイズの数値を実験 2 のプログラムで採用し、実験 1 で使用したテスト画像を抜粋し、実験 2 のため作成したプログラムに画像を入力する。

4.2.3 実験2 結果

図 11 の画像を入力すると図 12 が出力された。



図 11: 実験 2 で読み込んだ画像

#料理
#foodpic#foodstagram#foodlovers#foodporn#eat#amazing#クッキングラム#つくおき

図 12: 図 11 を入力した時の出力結果

次に、実験 1 で画像認識において正解ではないがハッシュタグの関連付けはできている画像としたファッション/4.jpg を入力すると図 13 が出力された。

#自撮り
#me#selfies#selfietime#セルフイー

図 13: ファッション/4.jpg を入力した時の出力結果

図 11 を入力すると、料理のグループに一番近いと予測し図 12 のように 1 行目に#料理が出力される。そして 2 行目にハッシュタグ料理と同時によく使われている人気ハッシュタグが候補として出力されている。

ファッション/4.jpg を入力すると自撮りのグループに一番近いと予測し 1 行目に#自撮りが出力される。そして 2 行目にハッシュタグ自撮りと一緒によく使われている人気ハッシュタグが候補として出力されている。

実験 1 のプログラムと学習までは同じなので入力画像に対して 77.1%の確率で人気ハッシュタグの対応付けができるシステムであった。

4.3 考察

実験 1 の画像認識における総合正解率に関しては最高値で 62.9%であった。表 (1)から(6)を見ると、612×612pixel と 108×108pixel では 612×612pixel の方が画像認識の正解率が高い結果となった。また、画像サイズが同じだと学習回数を上げても画像認識の正解率が上がるとは限らないことが分かった。

本研究では、インスタグラムに投稿されている画像の内 7 つの抜粋した人気ハッシュタグを付けて投稿されている画像を検索し、各グループの教師データとするため、各グループの学習画像を集めたが、この画像の集め方では既存技術のように具体的な画像ばかりで各グループの画像を集めるよりも画像グループの画像が抽象的な画像なので情報量が多く認識率が低い結果につながったのではないかと考えている。また、インスタグラムの投稿画像には様々なハッシュタグが付けられており、「いいね」をしてもらうため、多くのユーザーに見てもらうために投稿画像にあまり関係のないハッシュタグを付けている投稿者がいたこと。そして投稿者は、一枚の画像に多くのハッシュタグを付けているため投稿画像に様々なものが写っており、一枚の画像の情報量が多いことが原因でないかと推測した。

教師データと違う予測をしてしまっても、画像認識においては正解ではないが、ハッシュタグの関連付けはできている画像と画像認識の正解率を合わせた正解率は 80%~90%程度だろうと考えていたが、結果は 77.1%と予想を下回ってしまった。

実験 2 では、システムに投稿画像入力すると、入力画像は 7 つのグループの内どのグループに予測され、予測した結果とそれに関連する人気ハッシュタグを表示させることができるかの実験を行った。結果は入力画像に人気ハッシュタグおよび予測した結果とそれに関連する人気ハッシュタグを付けることはできたが、実験 1 同様に画像認識の正解率と画像認識において正解ではないがハッシュタグの関連付けはできている画像を合わせた正解率は 77.1%である。

つまり、投稿画像に全く関係のない人気ハッシュタグを付ける可能性が 22.9%もあるということだ。これでは、投稿者は人気のハッシュタグを付け見ってもらう機会は増えるが、ハッシュタグ検索でカテゴリ一別に人気のある画像を見ようと思ったユーザーは関係のない画像も検索にかかりハッシュタグの意味がなくなってしまう。画像認識の正解率も意識しなければならなかった。

第5章 まとめ

本研究では、インスタグラムなどの SNS への投稿画像をより多くの人に見てもらうために、人気ハッシュタグを画像認識を用いて予測させるプログラムの作成を行った。結果として、画像認識の正解率は 62.9%、正解ではないが人気ハッシュタグの関連付けはできている画像と画像認識の正解率を合わせた正解率:77.1%と予想を下回る正解率だった。

7種類の分類でこの正解率であれば、グループが増えたとき増々正解率が下がってしまう可能性がある。しかし、動物であれば動物と抽象的な画像グループではなく、犬や猫などより具体的な画像グループに分類しなければ正解率が上がらないと思うが、それを行うと人気ハッシュタグとの関連付けやグループの分類の数が段違いに増えてしまうことや人気ハッシュタグは有名なインスタグラム利用者や芸能人などが利用しているものが多く、流行りがあるので人気ハッシュタグを掲載しているサイトの更新も定期的に確認しなければいけないことの解決方法を考えることが今後の課題であると感じた。

謝辞

本研究を進めるにあたり、ご指導頂きました三好力教授に深く感謝いたします。

また、多くの方々のご指導、ご支援をいただきました。心より深くお礼申し上げます。

参考文献

- [1] インスタグラムの利用のアンケート調査 | ネットリサーチのマイボイスコム
<https://www.myvoice.co.jp/biz/surveys/22614/index.html>

- [2] 画像や映像を自動でタグ付けする AI サービス Clarifai を触ってみた
<https://qiita.com/OMOIKANESAN/items/c45ba1a3dc2bad00ed2a>

- [3] AI が画像に自動でタグ付け『TAGGER』。Instagram の「いいね」が増えるかも
<http://www.diamondblog.jp/aggregate/god-apps/2017/10/24/10178>

- [4] おすすめのハッシュタグを教えてくれる超便利ツール - AppBank
<http://www.appbank.net/2016/09/15/iphone-application/1252539.php>

- [5] よく使うハッシュタグの管理や検索が出来る『ハッシュタグ手帳』が便利！
<https://isuta.jp/category/iphone/2015/06/430147/>

- [6] Instagram ジャンル別 139 選人気ハッシュタグ
<http://www.kagua.biz/social/instagram/instagram-hushtag-2017.html>

- [7] Keras による、ものすごくシンプルな画像分類（りんごとオレンジ）
<https://qiita.com/hiroeorz@github/items/ecb39ed4042ebdc0a957>

- [8] 斎藤康毅（2016）「ゼロから作る Deep Learning Python で学ぶディープラーニングの理論と実装」 株式会社オーム社

付録

```
ソースコード 1
from keras.models import Sequential
from keras.layers import Activation, Dense, Dropout, Convolution2D, Flatten,
MaxPooling2D
from keras.utils.np_utils import to_categorical
from keras.optimizers import Adagrad
from keras.optimizers import Adam
import numpy as np
from PIL import Image
import os

image_list = []
label_list = []

for dir in os.listdir(r"C:\Users\masaya1\insta\train"):
    if dir == ".DS_Store":
        continue

    dir1 = r"C:\Users\masaya1\insta\train\" + dir
    label = 0

    if dir == "インテリア":
        label = "0"
    elif dir == "ファッション":
        label = "1"
    elif dir == "フィットネス":
        label = "2"
    elif dir == "ペット":
        label = "3"
    elif dir == "自然":
        label = "4"
    elif dir == "自撮り":
        label = "5"
    elif dir == "料理":
        label = "6"

    for file in os.listdir(dir1):
        if file != ".DS_Store":
            # 配列 label_list に正解ラベルを追加
            label_list.append(label)
            filepath = dir1 + "/" + file
            image = np.array(Image.open(filepath).resize((612, 612)))
            print(filepath)
            # 配列を変換し、[[Redの配列],[Greenの配列],[Blueの配列]]
            # の形にする。
            image = image.transpose(2, 0, 1)
            print(image.shape)
            # 出来上がった配列を image_list に追加。
            image_list.append(image / 255.)

# keras に渡すために numpy 配列に変換。
image_list = np.array(image_list)

Y = to_categorical(label_list)

# モデルを生成してニューラルネットを構築
model = Sequential()
model.add(Convolution2D(32, 3, 3, border_mode='same', input_shape=(3, 612,
612)))
model.add(Activation("relu"))
model.add(Convolution2D(32, 3, 3))
model.add(Activation("relu"))
model.add(MaxPooling2D(pool_size=(2, 2), border_mode="same"))
model.add(Dropout(0.25))

model.add(Flatten())

model.add(Dense(200))
model.add(Activation("relu"))
model.add(Dropout(0.6))

model.add(Dense(200))
model.add(Activation("relu"))
model.add(Dropout(0.6))

model.add(Dense(7))
model.add(Activation("softmax"))

# オプティマイザに Adam を使用
opt = Adam(lr=0.0001)
# モデルをコンパイル
model.compile(loss="categorical_crossentropy", optimizer=opt,
metrics=["accuracy"])
# 学習を実行。
model.fit(image_list, Y, nb_epoch=100, batch_size=25, validation_split=0.1)

for dir in os.listdir(r"C:\Users\masaya1\insta\train"):
    if dir == ".DS_Store":
        continue

    dir1 = r"C:\Users\masaya1\insta\test\" + dir
    label = 0

    if dir == "インテリア":
        label = "0"
    elif dir == "ファッション":
        label = "1"
    elif dir == "フィットネス":
        label = "2"
    elif dir == "ペット":
```

```
        label = "3"
    elif dir == "自然":
        label = "4"
    elif dir == "自撮り":
        label = "5"
    elif dir == "料理":
        label = "6"

    for file in os.listdir(dir1):
        if file != ".DS_Store":
            label_list.append(label)
            filepath = dir1 + "/" + file
            image = np.array(Image.open(filepath).resize((612, 612)))
            print(filepath)
            image = image.transpose(2, 0, 1)
            result = model.predict_classes(np.array([image / 255.]))
            print("label:", label, "result:", result[0])
            pred = model.predict(np.array([image / 255.]))
            print(pred)
```

```
ソースコード 2
from keras.models import Sequential
from keras.layers import Activation, Dense, Dropout, Convolution2D, Flatten,
MaxPooling2D
from keras.utils.np_utils import to_categorical
from keras.optimizers import Adagrad
from keras.optimizers import Adam
import numpy as np
from PIL import Image
import os
```

```
image_list = []
label_list = []

for dir in os.listdir(r"C:\Users\masaya1\insta2\train"):
    if dir == ".DS_Store":
        continue

    dir1 = r"C:\Users\masaya1\insta2\train\" + dir
    label = 0

    if dir == "インテリア":
        label = "0"
    elif dir == "ファッション":
        label = "1"
    elif dir == "フィットネス":
        label = "2"
    elif dir == "ペット":
        label = "3"
    elif dir == "自然":
        label = "4"
    elif dir == "自撮り":
        label = "5"
    elif dir == "料理":
        label = "6"

    for file in os.listdir(dir1):
        if file != ".DS_Store":
            # 配列 label_list に正解ラベルを追加
            label_list.append(label)
            filepath = dir1 + "/" + file
            image = np.array(Image.open(filepath).resize((612, 612)))
            print(filepath)
            # 配列を変換し、[[Redの配列],[Greenの配列],[Blueの配列]]
            # の形にする。
            image = image.transpose(2, 0, 1)
            image_list.append(image / 255.)

# keras に渡すために numpy 配列に変換。
image_list = np.array(image_list)

Y = to_categorical(label_list)

# モデルを生成してニューラルネットを構築
model = Sequential()
model.add(Convolution2D(32, 3, 3, border_mode='same', input_shape=(3, 612,
612)))
model.add(Activation("relu"))
model.add(Convolution2D(32, 3, 3))
model.add(Activation("relu"))
model.add(MaxPooling2D(pool_size=(2, 2), border_mode="same"))
model.add(Dropout(0.25))

model.add(Flatten())

model.add(Dense(200))
model.add(Activation("relu"))
model.add(Dropout(0.6))#クラス数

model.add(Dense(200))
model.add(Activation("relu"))
model.add(Dropout(0.6))#クラス数

model.add(Dense(7))#クラス数指定
model.add(Activation("softmax"))

# オプティマイザに Adam を使用
opt = Adam(lr=0.0001)
# モデルをコンパイル
model.compile(loss="categorical_crossentropy", optimizer=opt,
metrics=["accuracy"])
```

```

# 学習を実行。
model.fit(image_list, Y, nb_epoch=100, batch_size=50, validation_split=0.1)
filepath = r"C:\Users\masaya1\insta2\test\test.JPG"
image = np.array(Image.open(filepath).resize((612, 612)))
print(filepath)
image = image.transpose(2, 0, 1)
#image = image.reshape(1, image.shape[0] * image.shape[1] *
image.shape[2]).astype("float32")[0]
result = model.predict_classes(np.array([image / 255.]))
print(label)
#print(pred.shape)
print("result:", result[0])
pred = model.predict(np.array([image / 255.]))
print(pred)

if result[0] ==0:
    print("#インテリア")

if result[0] ==1:
    print("#ファッション")

if result[0] ==2:
    print("#フィットネス")

if result[0] ==3:
    print("#ペット")

if result[0] ==4:
    print("#自然")

if result[0] ==5:
    print("#自撮り")

if result[0] ==6:
    print("#料理")

if result[0] ==0:
    print("#livstagrammer#home#interior#living#loom#office#mygoodroom#roomclip
#myntori#instahome#coffetime#DIY#暮らし")

if result[0] ==1:
    print("#fashion#ootd#ootn#outfit#instafashion#coode#makeup#coordinate#kawaii#
コーデ#コーディネート#コーデ")

if result[0] ==2:
    print("#fitness#traning#health#fit#body#jog#run#gym#workout#sport#hot#diet#ワーク
アウト#筋トレ")

if result[0] ==3:
    print("#pet#animal#petstagram#動物名")

if result[0] ==4:
    print("#nature#mountain#water#forest#国名#県名#地域名")

if result[0] ==5:
    print("#me#selfies#selfietime#セルフイー")

if result[0] ==6:
    print("#foodpic#foodstagram#foodlovers#foodporn#eat#amazing#クッキング
グラム#つくおき")

```