

# ジオタグ付き写真を用いた玄関口クラスターの発見

伊藤光太郎<sup>†</sup> 横山 昌平<sup>†,‡</sup>

<sup>†</sup> 東京都立大学大学院システムデザイン研究科 〒191-0065 東京都日野市旭が丘 6-6

<sup>‡</sup> 東京大学生産技術研究所 〒153-8505 東京都目黒区駒場 4-6-1

E-mail: <sup>†</sup>ito-kotaro@ed.tmu.ac.jp, <sup>‡</sup>shohei@tmu.ac.jp

**あらまし** 近年, Twitter や Instagram などの SNS が普及し, インターネット上に多くのジオタグ付きのデータが存在するようになった. そのデータをクラスタリングし, 活用する機会が増加した. しかし, クラスタリングアルゴリズムに関する研究の多くは「データをクラスターに分けること」に焦点を当てており, 作成したクラスター自体に着目しているものは少ない. 本研究では我々が提案した密度ベースのクラスタリングアルゴリズムである EBSCAN を用いてジオタグ付き写真をクラスタリングし, 各写真に画像認識ライブラリである YOLO を適用することで, 各クラスター内で撮影された物体を検出する. その検出結果をもとに, 各クラスターの特性や入力したデータの範囲内地域における“玄関口”の役割を果たすクラスターを発見する.

**キーワード** クラスタリング, ジオタグ付き画像, YOLO, セマンティクス

## 1 はじめに

昨今, SNS や IoT の普及に伴い, インターネット上に存在するデータ量が爆発的に増加している. Twitter, Inc [1] によると, 2021 年 7 月から 9 月における収益化可能な 1 日当たりのアクティブユーザー数の平均は 2 億 1100 万人であると報告されている. また, 総務省が発表している令和 3 年版情報通信白書 [2] によると, 2020 年時点での世界における IoT デバイス数は約 253.0 億台であり, 2023 年には約 340.9 億台に達すると予測されている. これに加えて, MaaS や自動運転といったデジタルトランスフォーメーションの波が到来すると考えられている. これらのことから, 地理情報を含むビッグデータの分析技術の重要度は今後さらに高まっていくと予想される.

地理情報を含むソーシャルビッグデータの分析によって得られる知見の例として Sakaki ら [3] の研究がある. この研究では, 地震等のイベント抽出を地震計ではなく, SNS ユーザーの投稿を分析する事で行える事を示した. また Kan ら [4] は, タクシーの GPS データから交通渋滞の検出が可能である事を示した. このように, 地理情報を有するソーシャルビッグデータの分析を行うことは様々な場面で有益である.

しかし, 前述したようにインターネット上のデータ量は膨大であるため, 全てのデータを処理することが困難な場合がある. そのような場合, 何らかの処理によって一部のデータを抽出, あるいは加工し, 重要度の高いデータに焦点を当てて分析を行うことがある.

こうしたデータ処理の代表的な例として, クラスタリングが挙げられる. ここでのクラスタリングとは, 大量のデータを何らかのルールに則り, “かたまり (クラスター)”に分ける処理を指す.

クラスタリングの中でも, 地理情報を含むソーシャルビッグデータの分析によく使われるアルゴリズムとして, Ester ら [5]

の DBSCAN (Density-Based Spatial Clustering of Applications with Noise) が挙げられる. これはデータの密度に着目したアルゴリズムであり, それぞれのデータに対して, 近傍距離  $\epsilon$  と近傍点数 MinPts という二つのパラメータから定義される密度を満たすか否かでデータのクラスタリングを行う. DBSCAN が用いられる理由は, ソーシャルビッグデータは地理空間上に離散的に存在するデータであり, 多くの場合で, その分析は, 広大な空間に分散している離散データから, 高密度のクラスターを発見する事が第一歩となるためである.

この DBSCAN ではデータを単なる点群として扱っている. そのため, DBSCAN ではそのデータをたどる経路や順序といった情報は切り捨ててクラスタリングをすることになる. この場合, 距離は近くても実際に移動することが困難な二点が同一のクラスターに属してしまうといった状況が発生し, それが問題となる場合がある.

例えば, 森田ら [6] は SNS から取得した位置情報付き画像に対して DBSCAN を適用することで, 観光スポットごとのクラスター生成を目的としたクラスタリングを実行している. このようにデータから観光スポットを発見するためにクラスタリングを行う際, 川や壁といった障害物によって隔てられた二点が同一のクラスターに存在する可能性がある. また, 「実際には二ヶ所の観光スポットが一つのクラスターにまとまってしまう」といった事象が起こる可能性もある. 観光スポットやホットスポットを抽出する場合, 大きな障害物を越えずに移動ができるような地点のみを一つのスポットとして抽出することが求められていると考える方が自然である. しかし, DBSCAN により作り出されたクラスターには, 人の移動に関する情報が含まれていないため, 理想的な結果を得ることが難しい場合がある.

このような問題に対処するために, 我々 [7] [8] は EBSCAN (Entanglement-Based Spatial Clustering of Applications with Noise) という密度クラスタリングアルゴリズムを提案した. これはデータ間の移動軌跡における交点の絡まりに

着目することで、実際の移動を考慮に入れたクラスタリングが可能なアルゴリズムである。

また、多くのクラスタリングアルゴリズムにおいて、生成されるクラスタは単なるデータのかたまりであり、“クラスタの意味”は自動的に決まらない。従って、クラスタリングを行った人間がそのクラスタの意味を推定する必要がある。そのため、膨大なデータに対してはクラスタリングを行った後に多くのクラスタが生成され、その一つ一つに対し人間が確認するといった作業をしなければならない。この作業に対し、ユーザは多大な時間や労力をかける必要があった。

しかし、EBSCAN では移動軌跡を用いることで、実際の地理情報や移動情報に即したクラスタリングが可能である。そこで我々は、この EBSCAN を用いることで、地理情報付きのデータを自動的に「意味のあるかたまり」に分けることができるのではないかと考えた。

また、ジオタグ付き画像を扱い、生成されたクラスタに対してそのクラスタを分析する研究は行われている。しかし、その分析にはタグやタイトルといったユーザが付記するテキスト情報などを必要とするものが多い。本研究では、このようなユーザが付記するテキスト情報を使用せず、画像情報のみからそのクラスタが持つ意味を考える。そして最終的に、ユーザが指定した意味を持つクラスタを自動的に抽出することを目指す。

本研究では EBSCAN によってできたクラスタに対して、Alexey ら [9] が開発した物体検出アルゴリズムである YOLOv4 (You Only Look Once version4) を適用し、各クラスタ内で撮影された画像に写った物体の種類とその確度を検出する。その後、ユーザが設定する“目的軸”に基づきスコアの算出を行う。この算出されたスコアを意味スコアと定義する。この意味スコアを用いて、ユーザが設定した軸の意味を持つクラスタを発見する手法を提案する。

また、実際にインターネットからクロールしたデータを用いて実験を行うことで、提案手法によりユーザが設定した軸の意味を持ったクラスタが発見できることを示す。本研究では、この軸を飛行機や鉄道、船といった公共交通機関の乗り物に設定することで、その地域の「玄関口」にあたる意味を持つクラスタを発見した。“玄関口”の詳細については 3.3 節に記載する。さらに、本研究では異なる軸を設定した例として、「食事処に関するクラスタ」を発見する実験を行った。

本論文の構成は以下のとおりである。2 章では、関連研究を述べる。3 章では、提案手法について述べる。4 章では提案手法を用いた実験結果を述べ、5 章で考察を述べる。6 章で本研究のまとめと今後の課題を述べる。

## 2 関連研究

まず、地理的なデータを対象としたクラスタリングアルゴリズムの研究について述べる。Kisilevich ら [10] はジオタグ付き写真のコレクションを使用した、DBSCAN に基づく密度ベースのクラスタリングアルゴリズムである P-DBSCAN を提案した。この研究ではデータの近傍密度を計算する際に、写真の数

そのものではなく、その近傍に存在する写真を撮影したユーザ数をパラメータとしている。また、コア点に関する密度の変化率を考慮することで、密度が急激に変化した場所を検出可能としている部分も DBSCAN と異なる点である。

Shi ら [11] はジオソーシャルネットワークのユーザが訪れる場所を密度ベースでクラスタリングする手法を提案した。この手法は、ユークリッド距離を正規化した空間距離と、ユーザ間の友人関係から計算される社会的距離の両方を考慮した「ジオソーシャル距離」を用いて密度の計算を行う。これにより、社会的にまとまりがあり、高品質なクラスタの発見を可能とした。

次に、クラスタの意味について焦点を当てた研究について述べる。Kisilevich ら [12] は、ジオタグ付きの写真コレクションからセマンティクスを作ることで、イベントや場所の分析を容易とした。この研究では、作成クラスタ内の写真数を時系列分析することで、そのクラスタが持つ時間的特性を推測し、テキストクラスタリングや外部の POI データベースと組み合わせることで、そのクラスタ領域の情報を取得している。

しかし、この研究では時系列分析だけでわかるのはクラスタが持つ時間的特性のみであり、クラスタの意味を知るためには、ユーザが設定したタグやタイトルといったテキスト情報が必要である。本研究ではこうしたテキスト情報を使わずに、クラスタ内で撮影された画像とそれに付随するジオタグの情報のみでクラスタの意味を推測する。

最後に、ジオタグ付き画像に対して物体認識を適用している研究について述べる。橋本ら [13] は、来遊者に興味を与える地域特性を発見するために、SNS 上の画像を用いて、来遊者の興味度合いを評価する手法を提案した。この研究では、ある地域で撮影された画像に対して物体認識を行い、その物体の出現頻度を調べることで、来遊者の興味度合を定量的な値として表現している。この手法では sky や outdoor といった「冗長なタグ」が出現頻度の上位に来てしまっており、それを手動で削除している。本研究ではこのような「どのクラスタでも撮影される物体」がクラスタの意味へ与える影響を小さくするため、撮影された物体のクラスごとに正規化を施している。

また、中野ら [14] はユーザが撮影した観光写真に対して物体検出を行い、その結果と計画ルート情報を用いることで、ユーザの好みに基づく観光スポットを推薦する方法を提案した。この研究ではジオタグ付きの写真をクラスタとして取り出せるデータベースを作成している。その取り出したクラスタの中の“平均的な画像”からそのクラスタのベクトルと、“ユーザが撮影した画像群”の物体検出結果から個人の好みベクトルを作成する。その二つのベクトルのコサイン類似度を計算することで、取り出したクラスタが個人の好みに近いかどうかの予測を行っている。

この研究では、観光スポットの平均的な画像を用いて、そのクラスタのベクトルを作成している。そのため、クラスタの特徴を考えるという部分で本研究と似ている点がある。本研究ではクラスタの意味を求める部分に焦点を当て、ユーザが設定した意味を持つクラスタの発見方法を提案している。また、実際の地理情報と合わせることで、発見したクラスタが「ユーザが

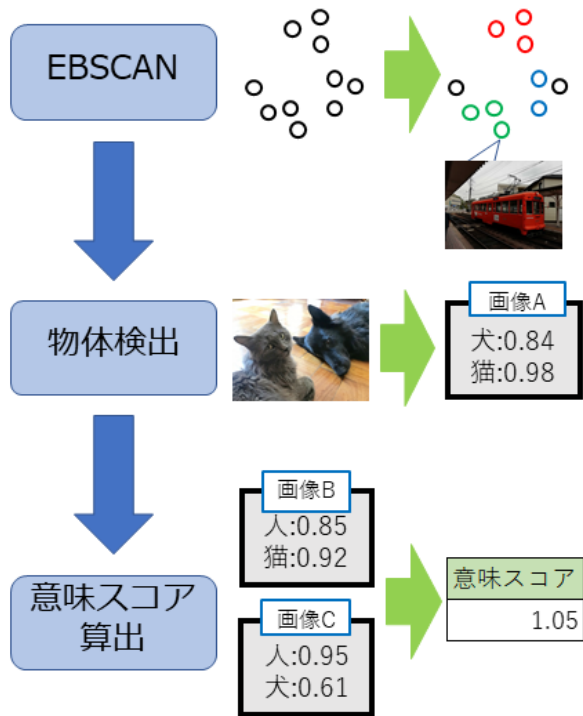


図 1 提案手法の概要

取得したかった（と想定される）クラスタかどうか」の検証を行っていることも本研究の特徴である。

### 3 提案手法

図 1 に提案手法の概要を載せる。まず、入力データとなるジオタグ付き画像を収集する。続いて、収集した画像に対して EBSCAN を適用し、クラスタリングを行う。次に、物体検出アルゴリズムである YOLOv4 を用いて、それぞれのクラスに属する画像内に写っている物体を検出する。続いて、ユーザーが「どのような意味のクラスを抽出したいか」によって“目的軸”を決定し、その目的軸を用いて意味スコアを計算する。最後に意味スコアを用いて、そのクラスが求めたい意味を持つクラスかどうかを判定する。

本章ではこれらの各ステップについて節ごとに分け、詳しい説明を行う。

#### 3.1 EBSCAN

EBSCAN は我々が提案した、移動軌跡に着目した密度クラスタリングアルゴリズムである。このアルゴリズムの特徴として、実際の移動を考慮に入れた密度クラスタリングが可能であることが挙げられる。

図 2 に EBSCAN の概要を載せる。図 2 の下部には、EBSCAN の各タスクにおけるデータ処理のイメージを示している。イメージ中において、赤と青の円はデータを表し、同じ色の円は同じユーザーによってアップロードされたことを示す。また、赤と青の矢印はそれぞれの移動軌跡を表している。そして、黒の円は移動軌跡の交点を表し、緑色の楕円はそこにあるデータが同じクラスに所属していることを表している。

EBSCAN の手順は大きく分けて軌跡計算タスク、交点探索タスク、クラスタリングタスクの三種類のタスクから構成される。

##### 3.1.1 軌跡計算タスク

はじめに、軌跡計算タスクについて述べる。まず、インターネット上の API などを用いてデータを取得する。ここで、取得する内容はクラスタリング対象のデータ（画像やテキスト）に加えて、データの作成者や日時、緯度経度といった情報が含まれている必要がある。

本研究では EBSCAN を適用した後に YOLOv4 を用いて物体検出を行う。従って、本研究で扱うデータは「移動軌跡を考えられ、物体検出ができる画像」が対象となる。本研究における実験では、この制約を満たすデータとして、写真共有サイトである Flickr からジオタグ付き画像を取得した。取得した内容は画像情報に加え、画像固有の photoID、その画像を投稿したユーザ ID、その画像を撮影した緯度経度、および撮影日時といった情報である。

データを取得後、そのデータのユーザ ID や日時などの情報を元に、移動軌跡を計算する。ここでデータが持つ緯度経度情報に対応する位置が移動軌跡の端点にあたる。

##### 3.1.2 交点探索タスク

続いて交点探索タスクを行う。軌跡計算タスクで計算した移動軌跡の情報を、緯度経度で表す地理座標系から  $x, y$  で表す直行座標系に置き換え、移動軌跡の交点を探索する。

交点の探索方法は Bentley ら [15] が提案した Bentley-Ottmann アルゴリズムを改良したものをを用いて行う。これにより、高速な交点探索を実現している。

##### 3.1.3 クラスタリングタスク

最後に、発見された交点の座標を地理座標系に戻し、クラスタリングタスクを実行する。

クラスタリングタスクでは、まず初期状態として、全てのデータを“未所属”に設定する。次に、「移動軌跡の交点を持つ線分の端点」にあたるデータを取り出す。この取り出した交点を有する線分の組み合わせに対して、端点間の距離を計算する。

計算された距離が事前に設定された閾値未満である場合、端点の所属状況に応じて、クラスタリングを行う。なお、この閾値を *toofar* と呼び、パラメータとしてユーザーが与える必要がある。また、いずれかのクラスに所属した端点の情報は所属済み端点リストで管理される。また、クラスタリングの順序は、「生成された時期が早い交点に関する線分の端点」が優先される。

端点の所属状況に応じたクラスタリング処理は以下のとおりである。

(1) どちらの端点もクラスに属していない場合  
新しくクラスを作成し、そのクラスに二端点を追加する。所属済み端点リストにその二端点の番号を所属先クラス番号とともに追加する。

(2) 片方の端点のみクラスに属している場合  
所属が“未所属”の端点を、もう片方の端点が所属するクラスに追加する。追加した端点の番号と所属先クラス番号を所属済み端点リストに追加する。

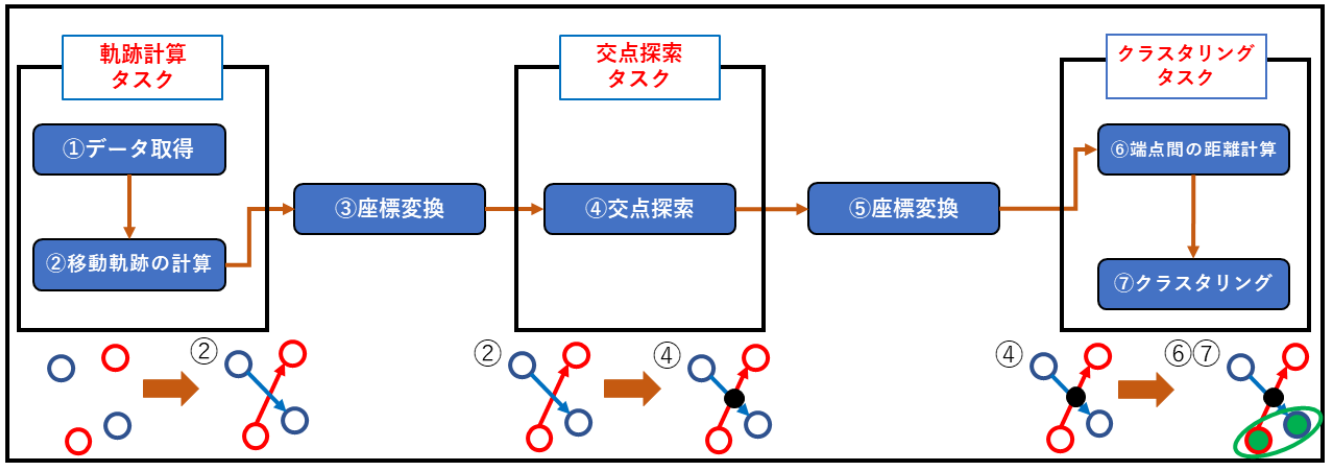


図 2 EBSCAN の概要

(3) 両方の端点がクラスタに属している場合

そのクラスタに追加された際の距離と、現在の二端点での距離を比較し、現在の二端点での距離の方が短ければ、クラスタの更新を行う。

両方の端点がクラスタに属している場合においては、その時点で所属しているクラスタに追加された際の距離（既存距離）と、現在の端点間の距離に応じて、より詳細な場合分けをすることで、クラスタの更新を行う。場合分けは以下のとおりである。

(3-c) 片方の既存距離が現在の距離よりも長い場合  
既存距離が長い方のクラスタから要素を除去し、既存距離が現在の距離よりも短い方のクラスタにその要素を追加する。また、移動させた端点の所属済み端点リストにあるクラスタ番号を追加した方のクラスタ番号に書き換える。

(3-b) 両方の既存距離が現在の距離よりも長い場合  
二つのクラスタを結合し、関係する所属済み端点リストの情報を書き換える。

(3-c) 両方の既存距離が現在の距離よりも短い場合  
クラスタに関する変更は行わない。

これらのルールに基づき、交点探索タスクで発見した交点を持つ軌跡に関して、端点の組み合わせを全て処理するまでクラスタリングを行う。その後、要素の移動により要素数が 1 以下となったクラスタを除去する。

また、要素数が 1 になったクラスタの要素に加え、移動軌跡が交点を持たないような端点や、他の軌跡との交点を持つが、その軌跡の二端点間の距離が *toofar* よりも大きい組み合わせしか持たない端点は外れ値として扱う。この外れ値はどのクラスタにも属さない点となる。従って、最終的に入力された写真（端点）はいずれかのクラスタに属する要素、もしくは外れ値としてクラスタリングされる。

### 3.2 YOLOv4

本研究ではクラスタリングした画像に対する物体の検出方法として、YOLOv4 を用いた。YOLOv4 は Redmon ら [16] が開発したリアルタイム物体検出手法である YOLO(You Only

Look Once) を改良したアルゴリズムである。YOLO の特徴として、画像を一度だけ調査し、その中に物体があるかどうかを予測し検出することで、高い検出精度と検出速度を両立していることが挙げられる。本研究では、オープンソースのニューラルネットワークフレームワークである darknet を用いて実装された YOLOv4<sup>1</sup> を使用する。

また、YOLOv4 の GitHub ページには Microsoft 社の COCO データセット<sup>2</sup> を使用した学習済みモデルが存在する。これは、“person”や“cat”, “clock”といった 80 クラスの学習をしたモデルである。本研究ではこの学習済みモデルを用いて、物体検出を行う。

また、YOLOv4 を用いて画像の物体検出を行う場合、「検出された物体とその確度」が計算され、出力される。このとき、閾値を設定することで、検出された物体の確度が低いものは出力しないようにすることができる。本研究では誤検出による影響を減らすため、確度が 0.5 (50%) を下回るものは出力しないように設定した。

### 3.3 目的軸の設定と意味スコアの算出

本節では目的軸の設定と意味スコアの算出方法について述べる。目的軸を決定する前に、検出する物体のクラスごとに「クラスタ内で検出された確度の合計」を計算する。この過程を図 3 に示す。まず、クラスタ内で撮影された各画像に対して YOLOv4 を用いて物体検出を行う。これにより、それぞれの画像において各クラスの物体が撮影された確度を出力する。そして、各クラスタで検出された物体の確度を足し合わせる。この足し合わせた値が「クラスタ内で検出された確度の合計」となる。

続いて、“目的軸”の設定を行う。この目的軸により、最終的にどのような意味を持つクラスタを出力するかを決定する。目的軸の設定は物体検出に使うモデルで学習させたクラスを選択することによって行う。

本研究ではこの目的軸を「玄関口」の意味を持つように設定

1 : <https://github.com/AlexeyAB/darknet>

2 : <https://cocodataset.org>



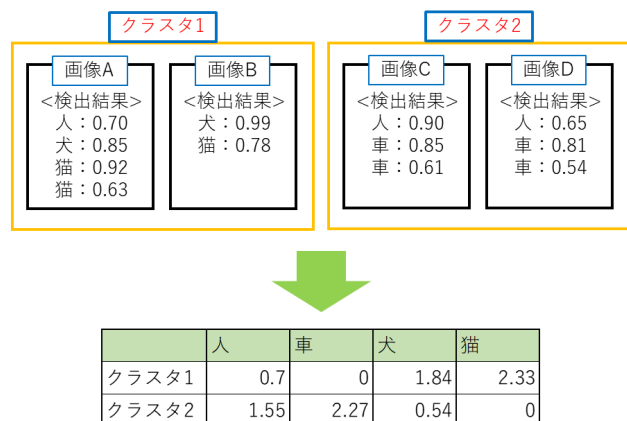


図3 各クラスにおける確度の合計値計算

することで、「玄関口クラス」の発見を目指す。ここで、玄関口クラスの定義について述べる。

デジタル大辞泉<sup>3</sup>で「玄関口」の項を引くと「ある場所への入り口。特に、国や都市への入り口となる鉄道の駅や空港、港などのこと。」と記されている。そのため、本研究では「入力データの範囲に存在する、駅・空港・港のうち、入力データ範囲外からアクセスが多いと考えられる場所」を玄関口とした。それに伴い、玄関口クラスは玄関口に当たる場所の近傍にあるクラスと定義した。

また、この軸は玄関口以外の意味を目的として設定することもできる。例を挙げると、「食事処」の意味があるクラスターの発見が目的なら”knife”や”spoon”といった食器のクラスを用いるといったように、目的軸を構成するクラスは変えることができる。そのため、ユーザは自身の目的に応じた軸を設定すれば良い。本研究ではその一例として、「食事処」の意味を持つクラスターを発見する実験を行った。実験の詳細は4.2節に記載する。

最後に「意味スコア」の算出を行う。意味スコアの算出過程の例を図4に示す。まず、物体検出が可能なクラスに関して、全クラスで検出された物体の確度の合計値を求める。次に、求めた合計値を用いて、クラスごとに、前述した「各クラスで検出された物体の確度」の値を割り、正規化する。その後、設定した目的軸に属するクラスの値を選択する。最後に、各クラスで選択した全クラスの値を合計する。この合計した値を意味スコアとする。

このスコアに対し、ユーザが設定した閾値を超えたクラスを「目的軸に設定した意味を持つクラス」として出力する。

## 4 実験

本章では、インターネットから取得したジオタグ付き画像群に対して、提案手法を用いて特定の意味を持つクラスを発見する。本実験では、Flickr APIを用いて取得したデータを入力データとする。今回は、岡山県及び、岡山県付近の瀬戸内海の島を一つの地域としてみなし、この地域内における特定の意味を持つクラスターの発見を目指す。従って、使用するデータは、

	人	車	犬	猫
クラスタ1	0.7	0	1.84	2.33
クラスタ2	1.55	2.27	0.54	0
合計値	2.25	2.27	2.38	2.33

↓ 正規化

	人	車	犬	猫
クラスタ1	0.31111	0	0.77311	1
クラスタ2	0.68889	1	0.22689	0

↓ 目的軸(クラス)の選択

	人	車	犬	猫
クラスタ1	0.31111	0	0.77311	1
クラスタ2	0.68889	1	0.22689	0

↓ 意味スコアの算出

	人	車	犬	猫	スコア
クラスタ1	0.31111	0	0.77311	1	1.773109
クラスタ2	0.68889	1	0.22689	0	0.226891

図4 意味スコアの算出過程



図5 Flickr から取得したデータ

東経 133.260 度から東経 133.410 度、北緯 34.400 度から北緯 35.350 度で撮影され、2018 年にアップロードされた画像 7589 件である。図5は入力データを地図上にマッピングしたものである。

### 4.1 玄関口クラスターの発見

本節では、提案手法を用いて玄関口クラスターを発見する。また、発見できたクラスターが、実際の地図情報から割り当てた正解クラスターかどうかの検証を行う。

まず、入力データに対し、EBSCAN を用いたクラスターリングを行った。なお、パラメータ設定は  $toofar=5000m$  に設定した。この時点で、クラスター数は 119 となっていた。

ここで、本実験の目的は「岡山県及び岡山県付近の瀬戸内海の島」における玄関口クラスターを発見することであるので、クラスターの全要素が兵庫県、鳥取県、広島県に位置するクラスターを手動で削除した。この時点でのクラスター数は 84 であった。

範囲外のクラスターを削除した後のクラスターリング結果を図6

3 : <https://daijisen.jp/digital/index.html>



図 6 EBSCAN 適用後のクラスタ

表 1 正解クラスタの割り当て

公共交通機関乗り場	正解クラスタ
岡山空港	50(gray), 51(darkred)
岡山駅	60(orange), 63(lightgray), 64(darkpurple)
倉敷駅	52(cabernetblue)
新倉敷駅	24(lightgreen)
岡山港	65(purple)
宇野港	32(darkred)
男木島	33(red)
犬島	41(orange)
豊島	43(lightgreen)
小豆島	46(purple)

に示す。図 6 において、同じクラスタに属するデータは同じ色かつ同じ記号のマーカーによって表示している。

続いて、正解クラスタを設定する。今回の実験は、「岡山県及び岡山県付近の瀬戸内海の玄関口クラスタ」を発見することが目的である。そこで、空港や乗降客数の多い駅、港といった大人数が使用すると考えられる公共交通機関の乗り場に近いクラスタを正解のクラスタに設定した。公共交通機関の乗り場と正解クラスタの割り当ては表 1 のとおりである。表 1 の正解クラスタ列には、クラスタ番号と、図 6 中のマーカーの色を記載した。本実験では全部で 13 個の正解クラスタを設定した。

次に、EBSCAN で作成したクラスタ内にある各画像に対し、YOLOv4 を用いて物体検出を行い、それぞれのクラスについてクラスタごとに検出された確度の合計を求め、目的軸を設定した。本研究では玄関口クラスタを発見するため、「airplane», 「boat», 「bus», 「train」の四つのラベルを元に目的軸を作成し、各クラスタの意味スコアを算出した。ここで意味スコアの閾値は 0.01 に設定した。

意味スコアが 0.01 以上のクラスタを抽出し、意味スコアに関して降順に並べたグラフを図 7 に示す。図 7 に記載されている全クラスタが「玄関口クラスタ」とであると判定されたクラスタである。また、図 7 では正解クラスタに設定したクラスタをオレンジで表し、それ以外のクラスタを白で表している。

## 4.2 食事処クラスタの発見

本節では目的軸の設定の自由度の高さを示すために、「玄関口以外の意味を持つクラスタ」を発見する実験を行う。具体的には「食事処を撮影したクラスタ」を発見することを目指す。本実験では食事処を「その場所で食事ができ、かつ食事の提供が

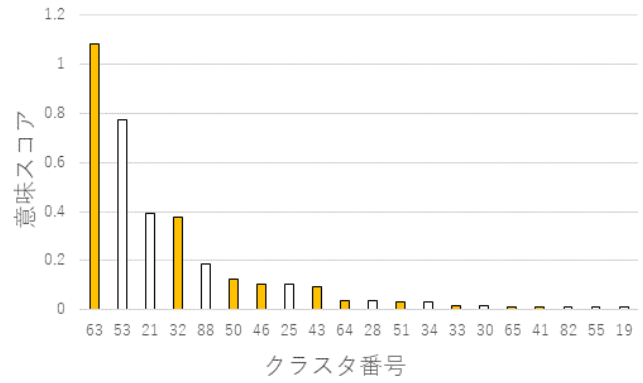


図 7 意味スコアが 0.01 以上のクラスタ

主たる目的である場所」と定義した。

また、本実験では、正解クラスタを地図情報から設定することが困難であった。そのため、本実験では画像ごとに「食事処を撮影した画像かどうか」といった正解ラベルを貼り、提案手法で閾値を超えたクラスタとそれ以外のクラスタを比較した。正解ラベルの基準は以下のとおりである。

### (1) 食事処の内部を撮影

基準) 料理やメニューが写っている。

### (2) 食事処の外観を撮影

基準) 店舗に加え、料理や食事処の名前が写っている。

入力データは玄関口クラスタの発見に関する実験と同様のものを用いる。この入力データに対して *toofar* を 30m に設定した EBSCAN を用いてクラスタリングを行った。この時点で、クラスタ数は 484 となっていた。

その後、玄関口クラスタの時と同様に、目的軸と意味スコアの閾値の設定を行った。本実験では、目的軸は「cup», 「fork», 「knife», 「spoon», 「bowl」の五つのラベルを元に作成し、意味スコアの閾値は 0.02 に設定した。この閾値を超えたクラスタ数は 32 個であった。

次に、それぞれのクラスタについて、以下の式を用いて食事処スコアを計算した。ここで、分母と分子の画像枚数はどちらも「各クラスタ内における画像枚数」のことを指す。

$$\text{食事処スコア} = \frac{\text{正解ラベルが貼られた画像枚数}}{\text{画像枚数}} \quad (1)$$

本実験で閾値を超えていないクラスタの食事処スコア平均は 0.0359 であり、閾値を超えたクラスタの食事処スコア平均は 0.485 であった。閾値を超えたクラスタの食事処スコア分布を図 8、それ以外のクラスタの食事処スコア分布を図 9 に示す。

## 5 考 察

本研究では玄関口と食事処といった二種類の意味を持つクラスタに関して、実験を行った。結果として、玄関口クラスタでは 13 個中 10 個の正解クラスタを発見できた。また、食事処クラスタに関しても、閾値を超えていないクラスタの食事処スコア平均は 0.0359 であったのに対し、閾値を超えたクラスタの食事処スコア平均は 0.485 であったことから、「提案手法によって

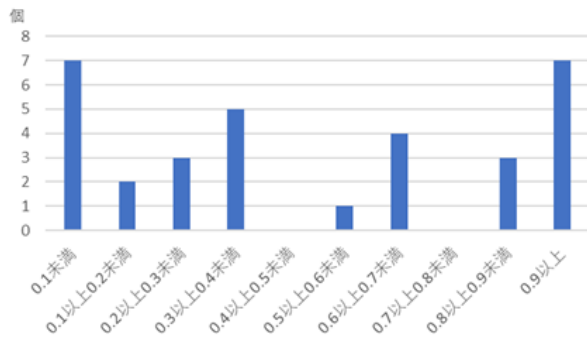


図 8 閾値を超えたクラスターの食事処スコア分布



図 9 閾値以下のクラスターの食事処スコア分布

食事処の撮影割合が高いクラスターを多く出力できた」と言える。

ここからはそれぞれの実験において「誤検出したクラスター」や「出力されなかった正解クラスター」が発生した理由、およびその改善策について述べる。

### 5.1 誤検出したクラスターについて

まず、本研究で行った二つの実験にて、正解ではないクラスターが出力された理由を述べる。玄関口クラスターに関する実験では 20 個のクラスターが閾値を超え、その中には正解クラスターではないクラスターが 10 個含まれていた。このようなクラスターを玄関口クラスターとして取得した理由は、大きく分けて三つある。

一つ目は、物体検出の誤検出があったことが挙げられる。例としてクラスター番号 88 のクラスターでは、物体検出の時点で“airplane”の値が高いことが意味スコアを高くしている一因となっていた。しかし、実際の画像を見てみると、飛行機が写っている画像以外に、レーシングカーを“airplane”として誤検出しているものがあつた。現状、意味スコアの算出はラベルごとに“そのクラスターでその物体が検出された数は全体の何割か”を考えていることに加え、本実験で用いた入力データの画像中において、airplane の撮影数は全体でも多くなかった。従って、1 枚の誤検出でも意味スコアに与える影響が大きく、想定していない結果となったと考えられる。この問題は物体検出アルゴリズムの精度に起因するため、今後、YOLO や他の物体検出アルゴリズムの精度が上昇することで解決できると考える。

二つ目は、今回の実験で使用したモデルのクラス分けが細かくなかったことが挙げられる。今回、物体検出に使ったモデルは大きな船と小さな舟がどちらも“boat”という同じラベルで

出力される。クラスター番号 53 では、物体検出の時点で“boat”の値が非常に高く、最終的な意味スコアもそれに連動して高くなった。これは実際の画像に倉敷美観地区の川舟流しの舟が写っていたことが原因であった。この問題は誤検出ではないが、「大きな船」を“ship”、「小さな舟」を“boat”などとラベル付けをするといった、学習クラスの細分化を行うことで解決できる。

三つ目は、規模の異なる玄関口を取得したことが挙げられる。これは特に“train”と“bus”の陸路の交通機関に見られた。例をあげると、クラスター番号 28 のクラスターでは、駅に停車中の西日本旅客鉄道宇野線の車両が写っており、緯度経度情報から撮影位置は茶屋町駅であることがわかった。このことから、「駅」という玄関口に関するクラスターの取得には成功しており、市や町としての玄関口クラスターとしてみれば悪い結果ではないと言える。しかし、今回の実験では「岡山県」の玄関口という、より大きな規模の玄関口クラスターを発見することが目的であったため、このクラスターは正解クラスターに含めていなかった。

仮に、このような「求めたい規模以外の玄関口クラスター」を除外したい場合、モデルのクラス分けを細分化する方法が考えられる。今回の例ならば、鉄道の中でも“新幹線”と“在来線”などのように、別のクラスとして分けて学習させたモデルを作成する。この場合、“新幹線”のクラスのみが大きな規模の玄関口クラスターの条件を満たすと考えられる。従って、“新幹線”と“airplane”などのクラスで構成される軸を用いることで、大きな規模の玄関口クラスターのみが取得できるのではないかと考える。

続いて、食事処クラスターに関する実験について述べる。この実験では、正解クラスターという形ではないが、閾値を超えたクラスターの中に正解ラベルが貼られた画像の割合が低いクラスターが含まれていた。これは、前述した物体検出時点での誤検出に加え、二つの理由が存在する。

一つは、休憩所のような場所などが撮影されたクラスターを取得したことである。これは、本実験において食事処を「食事の提供が主たる目的である場所」として定義したことが原因であり、「食事がとられている場所」と定義すれば問題がなかったと考えられる。また、本実験で用いた入力データには「人が直接見ても食事処かどうか判断に迷う画像」も一部存在しており、こうした画像の存在も結果に影響していると考えられる。

もう一つは、画像の枚数が多いクラスターを取得しやすい傾向があったことが挙げられる。画像の枚数が多いクラスターは食事処以外も撮影されていることがあり、そうしたクラスターは食事処の撮影割合が低かったと考えられる。このようなクラスターは「ユーザが指定した意味を持つクラスター」ではあるが、複数の意味を持つ。仮に、食事処の意味しか持たないクラスターだけを取得したい場合は、意味スコア算出を行う際に、そのクラスターの画像枚数などで補正をかけることで対処できると考える。

### 5.2 出力されなかった正解クラスターについて

続いて、発見できなかった正解クラスターについて述べる。玄関口クラスターに関する実験で発見できなかった正解クラスターはいずれも「駅に関する玄関口」と想定した場合の玄関口クラスターである。発見できなかった原因はいずれのクラスターも、目的

軸に属するクラスの物体がクラスタ内で撮影された画像に写っていないためである。しかし、クラス番号 24 のクラスは鉄道の内部から景色を撮影したものや、列車内部を撮影したと思われる画像が存在していた。このことから、レールや信号標識、あるいは車両内の案内といったクラスを学習させ、目的軸に追加することで、玄関口クラスタとして取得できるようになる可能性がある。

また、食事処クラスタを発見する実験では、数は少ないものの、「閾値を超えなかったが、正解ラベルが貼られた画像の撮影割合が高いクラスタ」も存在した。このようなクラスタは、クラスタの画像枚数が少ない状態で、物体検出の確度が低い（検出精度が甘い）場合や、メニューや外観の写真が多い場合に発生しやすいことがわかった。そのため、前節で挙げた問題と同様に、物体検出アルゴリズムの精度向上や意味スコア算出時のアルゴリズムを補正することで結果を改善できると考えられる。

## 6 まとめ・今後の課題

本研究では、EBSCAN によって生成されたクラスタに対し物体検出を行い、意味スコアを算出することで、「ユーザが設定した意味を持つクラスタ」を発見する手法を提案した。提案手法を用いれば、ユーザは取得したい意味に応じて軸を設定し、意味スコアの閾値を決めることで、それぞれのクラスタで撮影された画像を確認せずとも設定した意味のあるクラスタを抽出することが可能である。

本論文では目的軸を「玄関口」として設定し、“airplane”, “boat”, “bus”, “train”といったクラスの物体検出を行うことで、玄関口クラスタを発見する実験を行った。今回の実験では、出力された 20 個のクラスタの内、正解クラスタが 10 個含まれており、「ユーザが指定した意味を持つクラスタを発見する」という目的をある程度達成できることを示した。また、“cup”などの食器に関するクラスを用いて目的軸を構成した実験を行うことで、玄関口以外の意味を持つクラスタにも提案手法が適用できることを示した。

また、出力結果には正解クラスタ以外のクラスタも含まれていた。この問題は物体検出アルゴリズムの精度や学習させるクラスの問題に起因する部分がある。そのため、物体検出アルゴリズムの精度向上や学習させるデータ量の増加、学習させるクラスの細分化といった方法によって、対処できる可能性がある。

今後の課題として、本研究では意味スコアの閾値をユーザの知識によって決定していた。そのため、目的軸に設定したクラス数に応じて自動的に閾値を設定するなど、適切な閾値を決めるための方法について考える必要がある。また、今回の意味スコアの算出では単純に正規化した物体検出の確度の和を用いた。しかし、実際の画像では、同じ物体を撮影していても、「狙って撮影した画像」と「偶然映り込んだ画像」があると考えられる。そのため、「その物体が画像中に占める大きさの割合」などを用いて確度を補正することなども考えられる。こうしたスコア算出のアルゴリズムを発展させることも今後の課題である。

**謝辞** 本研究の一部は JSPS 科研費 19K11982 の助成を受

けた。

## 文 献

- [1] Twitter, Inc. Q3 letter to shareholders. [https://s22.q4cdn.com/826641620/files/doc\\_financials/2021/q3/Final-Q3-21-Shareholder-letter.pdf](https://s22.q4cdn.com/826641620/files/doc_financials/2021/q3/Final-Q3-21-Shareholder-letter.pdf).
- [2] 総務省. 令和 3 年版情報通信白書. <https://www.soumu.go.jp/johotsusintokei/whitepaper/ja/r03/html/nd105220.html>.
- [3] Takeshi Sakaki, Makoto Okazaki, and Yutaka Matsuo. Earthquake shakes twitter users: real-time event detection by social sensors. In *Proceedings of the 19th international conference on World wide web*, pp. 851–860. ACM, 2010.
- [4] Zihan Kan, Luliang Tang, Mei-Po Kwan, Chang Ren, Dong Liu, and Qingquan Li. Traffic congestion analysis at the turn level using taxis’ gps trajectory data. *Computers, Environment and Urban Systems*, Vol. 74, pp. 229–243, 2019.
- [5] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*, Vol. 96, pp. 226–231, 1996.
- [6] 森田格, 荒木光一. シェムリアップにおける twitter・flickr の画像・位置情報等を活用した観光資源種毎の穴場スポット抽出. 人工知能学会全国大会論文集 第 34 回全国大会 (2020), pp. 4Rin161–4Rin161. 一般社団法人 人工知能学会, 2020.
- [7] Shohei Yokoyama, Ágnes Bogárdi-Mészöly, and Hiroshi Ishikawa. Ebscan: An entanglement-based algorithm for discovering dense regions in large geo-social data streams with noise. In *Proceedings of the 8th ACM SIGSPATIAL International Workshop on Location-Based Social Networks*, pp. 1–10, 2015.
- [8] 伊藤光太郎, 横山昌平. 移動軌跡の交点を用いた密度クラスタリングアルゴリズム. 情報処理学会論文誌データベース (TOD), Vol. 14, No. 3, pp. 1–15, 2021.
- [9] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, 2020.
- [10] Slava Kisilevich, Florian Mansmann, and Daniel Keim. P-dbscan: A density based clustering algorithm for exploration and analysis of attractive areas using collections of geo-tagged photos. In *Proceedings of the 1st international conference and exhibition on computing for geospatial research & application*, pp. 1–4, 2010.
- [11] Jieming Shi, Nikos Mamoulis, Dingming Wu, and David W Cheung. Density-based place clustering in geo-social networks. In *Proceedings of the 2014 ACM SIGMOD international conference on Management of data*, pp. 99–110. ACM, 2014.
- [12] Slava Kisilevich, Daniel Keim, Natalia Andrienko, and Gennady Andrienko. Towards acquisition of semantics of places and events by multi-perspective analysis of geotagged photo collections. In *Geospatial visualisation*, pp. 211–233. Springer, 2013.
- [13] 橋本幸二郎, 三代沢正, 宮部真衣, 土屋健, 尾崎剛, 広瀬啓雄. 来遊者に興味を与える地域特性を発見するための computer vision api に基づく sns 投稿画像分析. 電気学会論文誌 C (電子・情報・システム部門誌), Vol. 140, No. 8, pp. 916–924, 2020.
- [14] 中野広貴, 荒澤孔明, 渡邊稜平, 服部峻. 観光写真から抽出した撮影者の好みに基づく観光スポット推薦. 電子情報通信学会技術研究報告, Vol. 118, No. 408, pp. 45–50, 2019.
- [15] Jon Louis Bentley and Thomas A Ottmann. Algorithms for reporting and counting geometric intersections. *IEEE Transactions on computers*, No. 9, pp. 643–647, 1979.
- [16] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 779–788, 2016.