

# Deep Learning 画像認識エンジン 生成作業効率化インターフェースの開発

池松大志<sup>†1</sup> 中江 俊博<sup>†1</sup> 長森 藤江<sup>†1</sup> 井前 麻理子<sup>†1</sup>  
宮下 直也<sup>†1</sup> 木全 英明<sup>†1</sup>

**概要:** 機械学習を用いた画像認識では、高精度の判定を可能にするために、大量の画像データと正解ラベルからなる学習データセットを用意する必要がある。学習データセット構築作業の効率化を目的としてラベル付与の自動化やクラウドソーシングが実施されているが、属人的・専門的な判断を要する場合には、むしろエンドユーザ自身によってラベル付与を実施することが望ましいと考える。そこで、本研究ではエンドユーザ自身が画像収集からエンジン生成までを効率的に実施することが可能な学習器のインターフェースの開発を行い、エンジン生成に要する作業時間の短縮を図った。具体的には、Deep Learning フレームワークとして広く利用されている Caffe を用いた Deep Learning 学習 API (Application Programming Interface) の開発および画像検索 API とのマッシュアップによる画像収集・登録・学習インターフェースを開発した。

**キーワード:** Deep Learning, 画像認識, アノテーション, API, ユーザインターフェース

## An Efficient User Interface to Simplify Training Process of Deep Learning for Image Recognition

Taishi IKEMATSU<sup>†1</sup> Toshihiro NAKAE<sup>†1</sup> Fujie NAGAMORI<sup>†1</sup> Mariko INOMAE<sup>†1</sup>  
Naoya MIYASHITA<sup>†1</sup> Hideaki KIMATA<sup>†1</sup>

**Abstract:** Image recognition by machine learning requires a large amount of image data with ground truth labels. Crowdsourcing is one method to gather image data from the Internet to create dataset of annotated images, however, it is not always suitable when target images are specialized for users. In this paper we propose an image recognition system with Deep Learning engine developed based on the assumption that most users are likely to put labels on just enough images to train according to their own criteria of prediction, especially for specialized tasks. The proposed system has an efficient user interface to provide uses with one-stop simplified operation from gathering images to generating a trained model, in order to moderate complicated operations and reduce total time for the operations. The proposed interface connects to our developed Deep Learning Training API (Application Programming Interface) that enables users to train image features based on a widely used Deep Learning framework “Caffe” and a publicly available image search API.

**Keywords:** Deep Learning, Image Recognition, Annotation, API, User Interface

### 1. はじめに

近年、SNS やデジタルコンテンツ等の流行により Web 上の画像が爆発的に増大し、画像の意味内容を自動的に認識させる技術に対する要求が高まっている。中でも機械学習技術を用いた画像認識は、計算機の処理性能の飛躍的な向上により、2000 年頃から盛んに研究が行われ始め、特に 2012 年には ILSVRC[1]において Deep Learning を用いた Hinton らのチームが、他と 10%以上のエラー率の差をつけたことにより注目を集めている。

こうしたデータ駆動アプローチにおいて重要となるのは学習データセットの構築である。一般に、学習データセットはデータ数が多いほど認識精度が向上すると言われていたが、それぞれの画像データに対して正解ラベルの付与が必要となるため、データセットが大規模になるほど構築

が困難となる。既存のデータセットとして、Caltech-101[2]、Caltech-256[3]があるが、これらは単一の研究グループが人手によって集めたものであるため、比較的画像の枚数が少ない。これに対して、より大規模なデータセットの構築を目的として、画像アノテーション技術がいくつか研究されている。例として、AnnoSearch[4]では、Web上のオンラインフォトサイトの240万枚の画像とユーザによって付加されたキーワードを用いて、類似画像検索を用いた自動画像アノテーションを行っている。また、柳井らは指定キーワードでWebクロールした画像に対して、ノイズとなる画像を低減するため、EMアルゴリズムを応用した繰返し手法を提案している[5]。また、自動アノテーションによるノイズを低減させるため、クラウドソーシングによるアノテーションもいくつか実施されている。ESP game[6]は、画像アノテーションをオンラインゲーム化し、ネットワーク上の多くの人々によってラベル付けを行ってもらう試みを実施している。また、2005年に米Amazonによって開始され

<sup>†1</sup> NTT コムウェア(株) NTT Comware Corporation.

たAmazon Mechanical Turk[7] は世界中にいるワーカーに対して比較的安価で画像データへの注釈作業を依頼し、2万以上のカテゴリ、1400万枚以上の画像からなるImageNet[8]の構築を実現している。

一方で、これらのデータセット構築手法は全て一般物体認識の精度向上に向けた取り組みであり、全ての画像認識に適用できる手法とは言えない。例えば、Webコンテンツ監視業務への適用事例として、画像が適切か不適切か判定するフィルタリングエンジンを開発する場合、判定基準は一般物体認識と異なりユーザ個人または集団独自で有する場合が殆どであり、そうした判定基準を他者と齟齬なく共有してアノテーションを実施することは難しいと考えられる。同様に、医療分野への適用事例として、がん細胞の有無などを判定する場合にも、専門家でないワーカーがアノテーションを実施するのは極めて難しいと考えられる。こうした属人的・専門的な判定を機械学習技術により実現する場合、他者に依存するデータセット構築手法では実現が難しいため、Caltech-101/256のようにユーザ自身で独自の学習データセットを構築し、認識エンジンを作成する必要があると考える。

そこで本研究では、ユーザ自身でデータセット構築・認識エンジン作成を効率的に実施可能なインターフェースを検討し、既存のデータセット構築手法では困難な認識エンジン開発において必要な人手作業の簡略化を図る。具体的には、機械学習技術やツール開発に関する知識を持っていない人をメインターゲットとし、Webベースで直感的に操作可能なインターフェースの検討を行う。

## 2. 問題分析

著書らが想定する認識エンジン開発の一連の作業と各作業における問題点は以下の通りである。

### (1) 画像収集およびラベル付け

前述のとおり、属人性・専門性のある判定業務の自動化を機械学習によって実現するためには、ユーザ自身で画像アノテーションを行う必要があると考える。その際、ユーザが学習用の画像を保有していない場合は、まず何らかの方法で学習用の画像を用意する必要がある。準備方法として、前述したImageNet等の大規模データセットに対して新たにラベル付を行うことも可能だが、これらは一般物体認識を行うために収集された画像群であり、ユーザの実現したい判定業務の判定対象を全てカバーしているとは言えない。こうした既存のデータセットの不足部分を補うためには、作成したいエンジンに合わせて画像収集もユーザ自身で行う必要があると考える。画像の収集先としてインターネットを利用する場合、ツール開発等の知識がないユーザはGoogle等の画像検索エンジンを利用して画像ダウンロードおよびラベル付け作業を一枚毎に繰り返す必要がある、

極めて効率が悪い。

### (2) 学習・テスト・再学習

前述したデータセット構築作業は、画像を一枚一枚目視確認する必要がある、可能な限り少ない枚数でユーザの求める認識精度のエンジンを開発できることが望ましい。そのため、ある程度の枚数をデータセットとして登録した時点でユーザが認識エンジンを開発でき、認識テスト実施により、随時精度を確認することができれば、必要以上の画像登録作業を回避することができ、効率的なエンジン開発が可能になると考える。しかしながら、そうした作業を実施するためには、ユーザは機械学習用ライブラリの複雑な使い方を理解する必要があり、知識の少ないユーザにとって困難な作業である。

## 3. 開発要件

前述した問題を解決するために、ターゲットとするユーザの条件とインターフェースに求められる要件を以下のよう

### ターゲットとするユーザ

- ユーザは学習用の画像を保有しておらず、かつ判定対象となる画像はインターネットから取得可能。
- ユーザはツール開発や機械学習に関する専門的な知識を有していない。

### 開発要件

- ユーザに馴染みの深いGoogle等の画像検索エンジンと同様のインターフェースで画像検索が可能であり、かつ一覧表示された画像の選定およびラベル付けを効率的に実施可能であること。
- 大量の画像を確認することを想定して、確認作業の進捗状況を把握しやすく、かつ確認忘れ等が発生しにくいインターフェースにすること。
- ラベル付け機能により登録したデータセットを用いて学習が可能であること。尚、ユーザが機械学習に関する必要な専門知識を有さずとも、学習実施可能にすること。
- 認識エンジンの精度を画像収集の途中でも確認でき、不足があれば追加可能であること。
- ユーザの作業効率向上および不快感低減のため、可能な限り操作に対する応答を素早く返却すること。特に

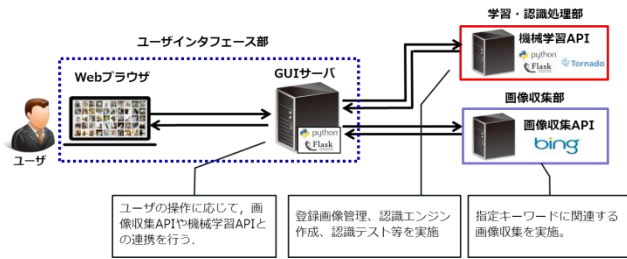


図 1 システムの全体構成

処理に時間を要する作業は、進捗状況を把握できること。

## 4. 提案インターフェース

### 4.1 システム全体構成

提案するインターフェースを実現するためのシステム全体構成を図 1 に示す。

本システムは主に 3つの機能部に分けられる。

#### (1) 画像収集部 (画像収集 API)

検索キーワードを入力パラメータとして、画像検索リクエストを送信すると、キーワードに関連する画像の URL 群を JSON 形式で返却する API である。本機能については、一般公開されている Bing Search API[10]を利用した。

#### (2) 学習・認識処理部 (機械学習 API)

送信される画像登録、学習および認識リクエスト等に応じて、それぞれの処理結果を JSON 形式で返却する API である。API 形式で提供することで、作成したエンジンによる認識処理を他のシステムに組み込みやすくなるというメリットがある。本研究では機械学習技術の一つである Deep Learning を実装対象とし、Deep Learning ライブラリとして広く利用されている Caffe[11]と、Python の Web フレームワークである flask[12]および tornado[13]を用いて実装する。

#### (3) ユーザインターフェース部 (GUI サーバ)

画像収集部や学習・認識処理部から取得した情報を整形し、ユーザに Web ブラウザベースの操作画面を提供する。本機能のサーバサイドは Python の Web フレームワークである flask, クライアントサイドは JavaScript を用いて実装する。

次節ではユーザインターフェースベースで詳細な実装について説明する。

### 4.2 インターフェースの詳細

#### 4.2.1 画像検索機能

図 2 に実装した画像検索機能の UI と操作イメージを示す。また、図 3 に画面検索機能の処理フローを示す。



図 2 画像検索機能の UI および操作イメージ



図 3 画面検索機能の処理フロー

図 2 に示す通り、テキスト入力フォームに検索キーワードを入力し、検索ボタンを押下することで (図 2. ①), サイズを固定した 50 枚のサムネイル画像を一覧で表示させる仕様となっている (図 2. ②)。また、画面下部のページネーション機能を利用することで、別の 50 枚を順次表示させることが可能になる (図 2. ③)。画像の一覧表示のさせ方として、無限スクロールによる実装も考えられたが、スクロールにより表示される画像を増加させていくと、どの程度画像を確認したかといった作業の進捗具合がわかりづらくユーザに不快感を与える原因となるとともに、表示された画像の確認を意図せず飛ばしてしまう可能性が高くなると考える。一方で、ページネーション機能は、1 ページに表示される画像の枚数が決まっており、ページが作業の進捗状況を示す目安となるとともに、全ての画像が一画面で収められ確認もしやすいため、画像のチェック忘れ等も発生しにくいと考える。また、ユーザへの画面レスポンスを向上させるため、ユーザサイドで画像を一覧表示させる際に、図 3 に示すように画像収集部やインターフェース部の間で画像データのやりとりは行わず、URL のみを送受信し、通信時間の短縮を図った。更に、URL はソース画像とサムネイル画像両方を送受信し、ユーザサイドで一覧表示

させる際は、サムネイル画像を用いることで画面表示の速度向上を図った。

#### 4.2.2 画像選択およびラベル付け機能

図 4 に実装した画像選択およびラベル付け機能の UI と操作イメージを示す。また、図 5 に画像選択およびラベル付け機能の処理フローを示す。

画像選択およびラベル付け機能では、まず作成したいエンジンに合わせて構築する学習データセットの ID を払い出す (図 4. ①)。これは、複数のデータセットを一意に特定するために存在する。尚、この際、学習データセットのタイプをタグ型かスコア型か選べるようになっており、タグ型の場合は 8 層畳み込みニューラルネットワーク [1] を用いたクラス分類用のネットワークを、スコア型の場合は 8 層畳み込みニューラルネットワークの最終層を二乗誤差関数に置き換えて判定結果を一次元量で出力するネットワークを作成可能となるようにした。後者のネットワークは、著者らの既存研究 [9] において高精度なフィルタリングを実現できることを報告している。学習データセットを選択後、一覧表示された画像から登録する画像を選択し (図 4. ②)、選択した画像に対してテキスト入力フォームにてスコア (またはラベル) を定義し (図 4. ③)、画像登録ボタン押下により登録処理を学習・認識処理部で実施する (図 4. ④)。この際、図 5 に示すように、クライアントサイドからは選択画像そのものではなくソース URL を送信し、学習・認識処理部にて画像をダウンロードすることで余計なトラ



図 5 画像選択およびラベル付け機能の処理フロー

フィックの発生を避け、処理速度向上を図った。画像選択の際は、全選択ボタンや全解除ボタン、選択反転ボタンを設けることで、作業効率向上を図った。また、ページネーション機能を使って表示画像を切り替えた場合に、遷移前に選択した画像は選択状態を維持するようにし、より多くの画像を一度に画像登録可能にすることで作業効率向上を図った。更に、画像アップロードにはある程度の時間を要するため、登録処理が完了した枚数をリアルタイムで表示するようにし、サーバサイドで処理が進行中であることを明示することでユーザの不快感の低減を図った。

#### 4.2.3 学習および認識テスト機能

図 6 に実装した学習機能の UI と操作イメージを示す。学習開始ボタン押下により表示されるダイアログにて、作成される認識エンジンの名前と使用する学習データセットを選択して OK を押下することで、サーバサイドで学習が開始される (図 6. ①)。本来 Deep Learning による学習を行うためには、ネットワーク設計や学習パラメータの細かなチューニングが必要となるが、知識のないユーザでも利用可能にするため、ネットワークは著者らの推奨モデルを固定で利用することとし、またパラメータについては著者らの経験則に基づき学習データセットの枚数によって動



図 4 画像選択およびラベル付け機能の UI と操作イメージ



図 6 学習機能の UI および操作イメージ





図 7 認識テスト機能の UI と操作イメージ

的に変化させる仕様にする事で、ユーザが本来実施する必要のある作業を隠蔽し、学習を極めて簡易に実施できるようなインターフェースとした。また、学習処理には数分～数時間を要するため、実施状況確認ボタンを用意し、学習終了予定時刻および学習進捗率を表示することで、ユーザが感じる不快感の低減を図った (図 6. ②)。

図 7 は認識テスト機能の UI と操作イメージである。図の通り、認識テストを行うために、まず認識エンジン管理ページにて、エンジンの起動を行う必要がある (図 7. ①)。エンジンの起動とは、エンジンを認識処理可能な状態でメモリ上にロードする処理を指すが、Deep Learning の特性として認識エンジンのデータ容量が数百メガバイトと大きく、メモリ上にロードするためには数秒かかるため、即応性を求める場合には認識リクエストを受け取った時点で既にメモリ上にロードされていることが望ましい。事前にロードしておく場合、メモリの物理的制約上、同時に起動可能なエンジン数には限界があり、生成した全てのエンジンを起動しておくことはできないため、エンジン起動・停止をユーザ自身で実施し、起動可能な最大エンジン数を超過した場合、エラーを出力する仕様とした。エンジン起動後、認識テストページへの遷移ボタンを表示し (図 7. ②)、認識テストページにて認識させたい画像の URL をテキスト入力フォームで指定し (図 7. ③)、Predict ボタン押下によって判定結果を表示する (図 7. ④)。ユーザは本機能によ

て、どの程度判定結果に妥当性があるかを判断でき、不満がある場合はデータセットに画像を追加し、再度学習を行うことで精度改善を図れる。

## 5. 評価と課題

### 5.1 評価

本章では、実装したインターフェースの評価を行い、そこで抽出された課題に対して改善案の検討を行う。

本研究では、評価手法としてウェブユーザビリティ評価スケール[14]を用いた。ウェブユーザビリティ評価スケールは、株式会社富士通と株式会社イードが共同で開発したウェブユーザビリティを定量的に評価するためのアンケート評価手法であり、ウェブユーザビリティに関する 21 項目各々について、5 段階評価質問を行い、その 21 項目の質問から生成される 7 つの評価因子でウェブサイトのユーザビリティを評価する。本評価手法を用いて、Deep Learning ライブラリ等の知識を有さない 3 名に本インターフェースの評価を依頼した。結果を表 1 に示す。

表 1 ウェブユーザビリティ評価スケール

評価因子	ユーザ A	ユーザ B	ユーザ C	平均
①好感度	4.00	3.67	4.33	<b>4.00</b>
②役立ち感	4.33	5.00	4.33	<b>4.56</b>
③内容の信頼性	4.33	4.00	4.33	<b>4.22</b>
④操作のわかりやすさ	4.67	4.67	4.00	<b>4.44</b>
⑤構成のわかりやすさ	4.00	3.00	4.00	<b>3.67</b>
⑥見やすさ	4.33	4.33	4.00	<b>4.22</b>
⑦反応のよさ	3.67	4.00	4.67	<b>4.11</b>

表の通り、全項目について平均 3.5 以上と高い評価を得ることができた。特に、「④操作のわかりやすさ」については、本来必要な Deep Learning ライブラリの複雑な使い方の習熟無しに平均 4.44 と高い評価を得ていることから、本インターフェースの有効性は極めて高いと言える。

### 5.2 評価者からの要望

前節にて高い評価を受けた一方で、評価者 3 名全員から出た機能追加要望として、学習データセットに登録された画像の確認機能および画像単位の削除機能がある。前述したとおり、本インターフェースによって画像選択およびラベル付け機能により、データセットへ画像を登録できるが、どのような画像を登録したか把握することはできない。また、確認不足や操作ミス等により誤って画像を登録してし

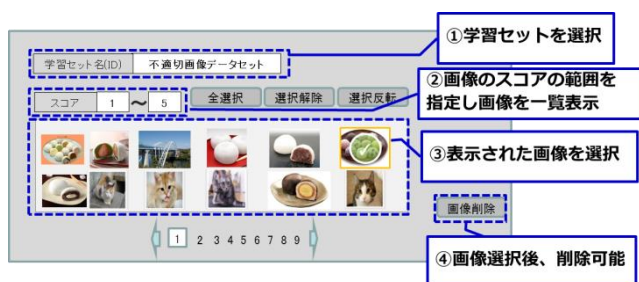


図 8 学習データ確認および削除機能の画面イメージ

まった場合に、後から画像を削除することができない。そのため、改善案として、図 8 に示す画面イメージを検討した。本画面では、まず対象となるデータセットを選択し、該当データセットがスコア型の場合は、表示するスコアの範囲を指定する。その後、指定された範囲の画像を一覧表示し、削除したい画像がある場合は、画像選択後に削除ボタンを押下する。

今後もユーザ要望に応じていく過程で、より優れたインターフェースの検討を図る。

## 6. おわりに

本研究では、機械学習技術やツール開発に関する知識を持っていない人をメインターゲットとし、ユーザ自身で画像収集・ラベル付け・認識エンジン作成を効率的に実施可能なインターフェースの開発を実施した。開発したインターフェースは、画像収集部、学習・認識処理部、ユーザインターフェース部の 3 つの機能部から構成され、それぞれの機能部間での通信コスト削減によるレスポンス性能の向上を図るとともに、ユーザインターフェース部において様々なユースケースを想定した機能を実装することで作業効率向上を図った。ウェブユーザビリティ評価スケールを用いた評価においては、全項目高い評価を得ることが出来たとともに、評価者から出た機能追加要望による改善案の検討を行った。

今後の課題としては、既存のデータセット構築手法との比較が挙げられる。例えば、ある特定ユーザの属人的な判定を実施するエンジン開発をクラウドソーシングにより実現しようとする場合、サンプル画像等の提示により、ユーザの判定基準をワーカーと共有することで、ある程度信頼性のあるデータセットの構築が可能と考える。ユーザ自身がラベル付けしたデータセットとクラウドソーシングによるデータセットで認識精度、作業時間、コスト等にどの程度差が現れるか検証することで、それぞれの手法の優位性について評価でき、状況に応じた手法を利用することが可能となる。

**謝辞** 本インターフェースの開発にあたって、インターンシップ生として協力いただいた千葉工業大学 大西宇宙氏に、謹んで感謝の意を表する。

## 参考文献

- 1) Krizhevsky, A., Sutskever, I., Hinton, G.: Imagenet classification with deep convolutional neural networks, *Advance in Neural Information Processing Systems* (2012).
- 2) Caltech 101 Image Dataset  
[http://www.vision.caltech.edu/Image\\_Datasets/Caltech101/](http://www.vision.caltech.edu/Image_Datasets/Caltech101/)
- 3) Caltech 256 Image Dataset  
[http://www.vision.caltech.edu/Image\\_Datasets/Caltech256/](http://www.vision.caltech.edu/Image_Datasets/Caltech256/)
- 4) Wang, X.-J., Zhang, L., Feng, J., Ma, W.-Y.: AnnoSearch: Image Auto-Annotation by Search, *IEEE Computer Vision and Pattern Recognition*, pp.1483-1490 (2006).
- 5) 柳井啓司: 確率的 Web 画像収集, *人工知能学会論文誌*, Vol.21, No.1, pp.10-18 (2007).
- 6) ESP game. <http://espgame.org/>
- 7) Amazon Mechanical Turk  
<https://www.mturk.com/mturk/welcome>
- 8) Torralba, A., Fergus, R., Freeman, W.T.: 80 Million Tiny Images: A Large Data Set for Nonparametric Object and Scene Recognition, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp.1958-1970 (2008).
- 9) 井前麻理子, 中江俊博, 長森藤江, 長谷川将平, 池松大志, 宮下直也, 木全英明: 畳み込みニューラルネットワークを用いた画像コンテンツフィルタリングにおける回帰分析の評価, *SSII2015, IS2-15*.
- 10) Bing Search API  
<https://datamarket.azure.com/dataset/bing/search>.
- 11) Caffe | Deep Learning Framework  
<http://caffe.berkeleyvision.org/>
- 12) Welcome | Flask (A Python Microframework)  
<http://flask.pocoo.org/>
- 13) Tornado Web Server  
<http://www.tornadoweb.org/en/stable/>
- 14) ウェブサイトユーザビリティアンケート評価手法の開発  
[http://u-site.jp/wp-content/uploads/his\\_10th\\_paper.pdf](http://u-site.jp/wp-content/uploads/his_10th_paper.pdf)