Vol.2017-CVIM-207 No.13
2017/5/10

食事レシピ情報を利用した食事画像からのカロリー量推定

會下　拓実[1,†1,a)]　柳井　啓司[1,b)]

概要：食事画像からのカロリー量推定は, 様々な食事管理アプリケーションにとって重要であるが, 現状ではカロリー量推定に人手が必要であったり, 自動であったりしても種類が限定され, 複数視点からの画像が必要であったりと, 完全自動カロリー量推定を実用的な精度で実現できたことはいまだかつてなく, 食事画像からのカロリー量推定は未解決の問題となっている. そこで本研究では, 深層学習を利用したカロリー量および食事カテゴリ, 食材, 調理手順情報の同時学習による食事画像からのカロリー量推定手法を提案する. カロリー量および食事カテゴリ, 食材, 調理手順情報の間には高い相関が存在するため, これらの情報の同時学習による性能は, 各情報を独立に学習した場合の性能を上回ると考えられる. この同時学習を実現するために Multi-task CNN [1] を用いる. また, 本研究では Web 上の日本語のレシピサイトから収集したカロリー量情報付き食事画像のデータセットと, 英語のレシピサイトから収集したデータセットの 2 種類のデータセットを作成した. 複数のタスクを同時に行う Multi-task CNN と単一のタスクを行う Single-task CNN を学習し比較を行った結果, 両方のデータセットにおいて Multi-task CNN の性能は Single-task CNN の性能を上回った. 日本語のデータセットでのカロリー量推定実験では, マルチタスクにより相対誤差が–2.0%, 絶対誤差が–9.5kcal, 相関係数が+0.039, 相対誤差 20%以内の割合が+4.2%となり改善が見られた.

1. はじめに

近年, 健康志向の高まりにより日々の食事を記録し管理する様々なアプリケーションがリリースされている. それらの中には画像認識により食事画像から自動で食品名やカロリー量を推定するものも存在し, 栄養学の知識のない人がカロリー量を記録することが可能となっている. しかしこれらは食事カテゴリや量などの情報をユーザが手動で入力する場合が多いため, リアルタイム性に欠け, 主観による評価であるという問題がある. この問題の解決にはモバイル上での食事画像からの自動認識が有効であり, 様々な研究が存在する [10] [13] [18] [6] [8] [2]. しかし推定されたカロリー量の多くは, 食事カテゴリごとに固定されたカロリー量もしくは各食事カテゴリの基準サイズと比較した相対サイズから算出されたカロリー量である. このように現状では自動で食事画像からカロリー量を推定することが可能なアプリケーションは存在しない. CNN に基づく画像認識手法による進展により, 食事カテゴリ認識を含む画像認識タスクの多くは解決されつつあるが, 食事画像からのカロリー量推定は未解決の問題となっている.

食事画像からのカロリー量推定についてはこれまでにいくつかアプローチが取られている. 主要なアプローチは, 推定された食事カテゴリと食品の面積や体積の情報からカロリー量を推定する手法であり, 標準的な手法である [10] [13] [5] [6] [8] [14]. カロリー量は食事カテゴリと量に強く依存するため, これらの情報からカロリー量を推定することは有効であり, 重要なアプローチである. 食事カテゴリ分類問題はほぼ解決しているため, 現在は食品の面積や体積の推定が解決すべき問題となっている.

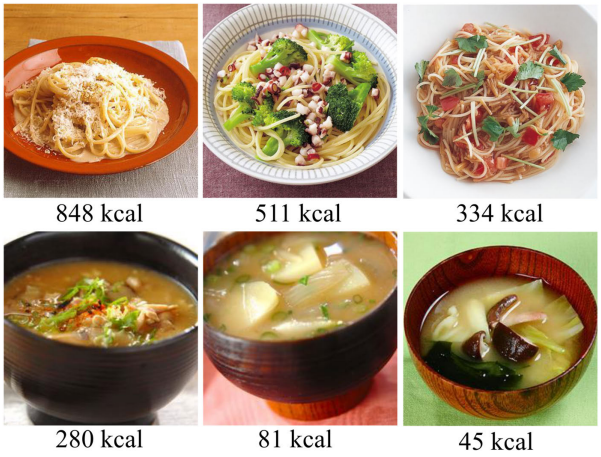


図 1　食事画像とカロリー量. 上段:スパゲッティ, 下段:味噌汁.

別のアプローチとして食品の面積や体積の推定を介さずに食事画像から直接カロリー量を推定する手法があるが, このアプローチを取る研究は少ない [12]. カロリー量は食事カテゴリに強く依存し, さらに量や食材, 調理手順に依存しており, それらは図 1 のように完成した食品の外見に現れる. 同一の食事カテゴリであっても使用される食材や調理手順によってカロリー量は変化するため, これらを考慮することも必要である. 食事画像から直接カロリー量を推

[1] 電気通信大学　総合情報学科
[†1] 現在, 同大学院　情報理工学研究科　情報学専攻所属
[a)] ege-t@mm.inf.uec.ac.jp
[b)] yanai@cs.uec.ac.jp

定する手法は図 1 のような食事カテゴリ内の差異を反映することが可能である.

本研究は CNN を用いて食事画像からカロリー量を直接推定するため, 後者のアプローチに属する. また我々は, カロリー量および食事カテゴリ, 食材, 調理手順情報の同時学習による食事画像からのカロリー量推定手法を提案する. 食事カロリー量および食事カテゴリ, 食材, 調理手順情報の間には高い相関が存在するため, これらの情報の同時学習による性能は, 各情報を独立に学習した場合の性能を上回ると考えられる. 我々はこの同時学習を実現するために Multi-task CNN [1] を用いる. Multi-task CNN による食事画像認識を行った Chen らの研究 [4] は, 食事カテゴリと食材情報を同時学習することで両方のタスクの精度が向上することを示しいる. 我々は Chen らの研究に刺激され, より困難なタスクであるカロリー量推定に Multi-task CNN を用いることを考える. カロリー量推定は食事画像を入力としてカロリー量を直接出力する回帰問題として扱われる. 1 種類の食品を写したシングルラベル食事画像を入力として, 写真中の食品の 1 人分の量に対応するカロリー量を推定する. 食事カテゴリ分類は通常のカテゴリ分類問題として扱われる. 食材情報に関しては, Word2Vec [11] により食材情報を分散ベクトル表現に変換し, その分散ベクトルを食材情報として学習する. 調理手順情報に関しても食材情報と同様にして調理手順文章を分散ベクトル表現に変換し, その分散ベクトルを調理手順情報として学習する. さらに本研究では Web 上のレシピサイトからカロリー量情報付き食事画像を収集し, 2 種類データセットを構築する. 食事画像データセットには Food-101 [3] や UECFOOD100 [9], VIREO Food-172 [4] などいくつかあるが, 現在, カロリー量がアノテーションされた食事画像データセットは公開されていない.

まとめると本論文は, Multi-task CNN [1] を利用してカロリー量と食事カテゴリ, 食材, 調理手順情報の同時学習を行い, 食事画像からカロリー量を直接推定する. さらに本研究では Web 上のレシピサイトからカロリー量情報付き食事画像データセットを構築する.

2. 関連研究

2.1 食事画像からのカロリー量推定に関する研究

食事画像からのカロリー量推定にはいくつかのアプローチが存在するが, 主要なアプローチは, 推定された食事カテゴリと食品の面積や体積の情報から, 事前に登録された食事カテゴリごとの単位面積当たりもしくは単位体積当たりのカロリー量の値を利用してカロリー量を推定する手法である.

Chen ら [5] は食事カテゴリを推定後, Kinect のような深度カメラにより食品の体積を推定し, 最終的にカロリー量を推定している. 深度カメラによる食品の体積の推定は正確であるが特殊なデバイスであるため, 一般の人が普段使用することは難しいと考えられる.

Kong ら [8] は DietCam という複数枚の画像からカロリー量を推定するアプリケーションを提案している. このアプリケーションは食事カテゴリ認識と領域分割を行い, さらに食品の三次元モデルの再構成を行い, 最終的に推定された体積の値からカロリー量を推定している. 三次元モデルの再構成では局所特徴量に基づくキーポイントマッチングとホモグラフィ推定が行われている. Dehais ら [6] の研究もこれに似ており, 皿の検出と領域分割, 食事カテゴリ分類を行い, 複数枚の画像から三次元モデルの再構成を行い, 最終的に炭水化物の量を推定している. このような複数視点からの画像により体積を推定する方法は, 事前にスマートフォンのカメラの較正を行わなくてはならなかったり, 正確に較正した地点から撮影を行わなくてはならず, ユーザーに対する負担が大きいと考えられる.

Meyer ら [10] は Im2Calories というアプリケーションを提案しており, 食事/非食事の認識, 複数品目の認識, 深度推定, 領域分割などの複数のタスクを CNN により行い, カロリー量を推定している. まず, 食事/非食事認識により画像中に食品が存在するかを判定し, その後マルチラベル認識により画像中の複数の食品を認識する. 次に深度推定と領域分割を行い, オブジェクトの三次元構造と食品の領域を抽出し, これらの情報を統合して食品の量を推定する. 最後に食事カテゴリや量の情報から食品のカロリー量を推定している. この研究では, タスクごとに必要な学習データを独自に作成しているため, かなりのコストがかかると考えられる. また, カロリー量情報付きのデータセットが不足し, 十分に評価が行われていない問題点がある.

Pouladzadeh ら [14] は食品とユーザーの親指を同時に撮影することで指の大きさと比較を行い食品の大きさを求め, カロリー量を推定するシステムを提案している. しかし指の出し方や角度, 映り方などによっては誤差が生じてしまう可能性がある

岡元ら [13] は大きさが既知の基準物体と一緒に食品を撮影することで食品の体積を推定し, 高精度のカロリー量推定を実現した. まず, 基準物体と食品を一緒に撮影し, 基準物体と食品のそれぞれの領域を抽出する. そして基準物体と食品の領域を比較して算出した食品の大きさからカロリー量を計算する. 食品の領域の抽出では, まずエッジにより背景から皿領域を検出し, その皿領域に対して k-means により色情報に基づく領域分割を行い, 最終的に GrabCut により皿領域から食品領域を推定する. 実験には基準物体と食品が一緒に写った画像が必要であり, データセットは手作業で作成された.

以上のように食事カテゴリと食品の量を推定するのが標準的なアプローチである. 本研究はこれとは異なり, 食品の量を介さず食事画像からカロリー量を直接推定する. 同様に食事画像からカロリー量を直接推定する研究として宮崎らの研究 [12] が存在する. 宮崎らは色ヒストグラムや SURF などの低レベル特徴量に基づいて, データベース上の類似画像を検索し, 特徴量ごとに類似度の高い上位 n 枚のカロリー量の平均値を計算し, それらの値から最終的にカロリー量を推定している. データセットには Web サービスである FoodLog[*1] に投稿された食事画像 6512 枚を使

[*1] http://www.foodlog.jp/

用し, 栄養学の知識を持った複数の専門家が食事画像にカロリー量をアノテーションしている．データセットには複数品目の画像も含まれ, 1 人分のカロリー量がアノテーションされている．この手法は色特徴や局所特徴量に基づく Bag-of-Features 特徴などの hand-crafted features のみを用いているため, 高精度の推定を行うことは困難であると考えられる．それに対して本研究では画像認識において成功を収めている CNN を利用するため，大幅な精度向上が期待できる．

2.2 Multi-task CNN と食事画像に関する研究

複数のタスクを同時に学習するために, これまでに Multi-task CNN [1] が提案されている．この研究では顔属性の認識を行っており, 複数の属性を同時に学習するために Multi-task CNN が提案されている．

Multi-task CNN に食事画像を適用した研究として Chen らの研究 [4] が存在する．Chen らは, Multi-task CNN により食事カテゴリと食材情報を同時に学習することで, 両方のタスクの精度が向上することを示している．これらは異なるタスクであるが, 食事カテゴリと食材には高い相関があるため, 両タスクに共通する特徴が学習され, 性能が向上したと考えられる．本研究は Chen らの研究に刺激され, より困難なタスクと考えられるカロリー量推定に対して Multi-task CNN を用いることを考える．

3. 手法

本研究では CNN を用いて食事画像からのカロリー量推定を行う．CNN の学習には国産の Deep Learning 用フレームワークである Chainer[*2] [19] を使用する．学習する食事画像は 1 種類の食品を写したシングルラベル画像であり, 写真中の食品の 1 人分の量に対応するカロリー量を推定する．カロリー量の推定は食事画像を入力としてカロリー量を直接出力するため回帰問題として扱われる．さらに本研究では Multi-task CNN によりカロリー量に加えて食事カテゴリや食材, 調理手順の情報を同時に学習する．食事カテゴリを推定する問題はマルチクラス分類問題として扱われる．食材情報に関しては, 認識対象とする食材の選定や, 表記ゆれの問題の解消のために, Word2Vec [11] により食材情報を分散ベクトル表現に変換し, その分散ベクトルを食材情報として学習する．調理手順情報に関しても食材情報と同様にして調理手順文章を分散ベクトル表現に変換し, その分散ベクトルを調理手順情報として学習する．

3.1 Multi-task CNN の概要

本研究で用いる Multi-task CNN のアーキテクチャは VGG16 [16] に基づく．VGG16 は畳み込み層が 13 層, 全結合層が 2 層, 出力層が 1 層の合計 16 層の深いネットワークであり, その性能と汎用性の高さから様々な研究に適用されている．この VGG16 [16] を拡張し, Multi-task CNN を実装する．本研究で使用する Multi-task CNN は図 2 のように, fc6 層までの層が全てのタスクで共有され, fc7 層以降の層が各タスクで独自に学習される．したがって各タスクは独立した fc7 層と出力層を有する．また, 入力はシングルラベルの食事画像であり, 同時に各タスクの推定値が出力される．Multi-task CNN を用いて食事カテゴリと食材情報を学習した Chen らの研究 [4] は, それぞれのタスクが独自の中間層を持っているとき性能が向上したと述べており, 本研究はそれに従う．

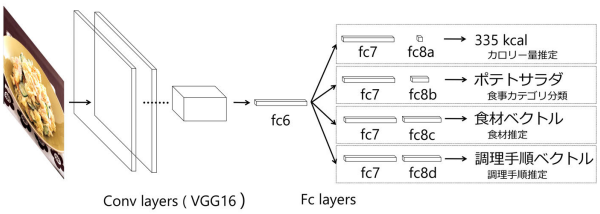


図 2 本研究で使用する Multi-task CNN のアーキテクチャ．

本研究ではカロリー量に加えて食事カテゴリ, 食材, 調理手順の情報を同時に学習する．各タスクの損失関数を L_{cal}, L_{cat}, L_{ing}, L_{dir} とし学習データの総数を N とすると, 全体損失関数 L は次のように定義される．

$$L = -\frac{1}{N}\sum_{n=0}^{N}(\lambda_{cal}L_{cal} + \lambda_{cat}L_{cat} + \lambda_{ing}L_{ing} + \lambda_{dir}L_{dir}) \quad (1)$$

ただし λ は各タスクの損失関数にかかる重みであり, 各 λ の値はすべての損失項が同程度の値に収束するように決定される場合が多いが, 最高性能を引き出すためには各損失項において微調整が必要になる場合が多い．

3.1.1 カロリー量の学習

カロリー量推定タスクは 4096 次元の fc7 層と, カロリー量を出力する単一のユニットで構成される出力層を有し, 1 人分のカロリー量の値を出力する．このような回帰問題においては, 一般的に損失関数として 2 乗和誤差が用いられるが, 本研究では次のような損失関数を使用する．絶対誤差を L_{ab}, 相対誤差を L_{re} とすると, カロリー量推定タスクの損失関数 L_{cal} は下のように定義される．

$$L_{cal} = \lambda_{re}L_{re} + \lambda_{ab}L_{ab} \quad (2)$$

ただし λ は各損失にかかる重みである．絶対誤差は推定値と正解値の差の絶対値であり, 相対誤差は絶対誤差と正解値の比である．どちらの誤差も重要な指標であるため, 両方とも考慮することが望ましいと考えられる．式 (2) のように絶対誤差と相対誤差を組み合わせた損失関数を使用することで, 両方の誤差が減少する．ある画像 x を入力したときの推定値を y, y に対する正解値を g とすると, 絶対誤差 L_{ab} と相対誤差 L_{re} は下のように定義される．

$$L_{ab} = |y - g| \quad (3)$$

$$L_{re} = \frac{|y - g|}{g} \quad (4)$$

[*2] http://chainer.org/

3.1.2 食事カテゴリの学習

食事カテゴリ分類タスクは 4096 次元の fc7 層と, カテゴリ数分のユニットで構成される出力層を有し, 食事カテゴリのクラス確率を出力する. 損失関数として交差エントロピー誤差を使用する. ある画像 x を入力したときの出力層のユニット i の出力値を y_i, y_i に対する教師データの値を g_i とすると, 食事カテゴリ分類タスクの損失関数 L_{cat} は次のように定義される.

$$L_{cat} = -\sum_{k=1}^{n} g_k \log y_k \tag{5}$$

ただし g_k はバイナリ値であり, ユニット k が正解のユニットであれば $g_k = 1$ となり, ユニット k が正解のユニットではなければ $g_k = 0$ となる. n は食事カテゴリが 20 種類の場合は $n = 20$ となる.

3.2 食材情報の学習

本研究では Word2Vec [11] による単語の分散表現を用いることで, 各レシピデータの食材名の単語を低次元実数ベクトルに変換し, それを食材情報の学習のための教師データの作成に利用する. 最終的に各レシピの食材情報は実数ベクトルに変換され, 教師データとして学習に利用される. この方法では個々の食材の有無などを推定することはできないが, Multi-task CNN による食材情報の同時学習の効果を得る上では問題はないためこの方法を採用する.

Word2Vec の学習に使用する文章は予め低頻度の除去や高頻度語のサブサンプリングなどの処理を行う. モデルには Skip-gram [11] を使用し, 学習時にネガティブサンプリング [11] を行う. これにより学習された Word2Vec から得られる単語の分散表現を使用するため, 辞書に含まれない単語は無視される. また, 本実験では各レシピデータにおいて tf-idf 値の上位 N_{max} 個までの食材名の単語のみを利用する. N_{max} は 1 レシピデータから抽出される食材名の単語の数の平均値とする. こうして得られた単語の分散表現と tf-idf 値から各レシピデータの食材ベクトルを生成する. あるレシピデータ r_j の食材情報を食材名の単語 w_i とすると, レシピデータ r_j の食材ベクトル v_j は次のように表される.

$$v_j = \sum_{k=1}^{N} tfidf_{k,j} * word2vec(w_k) \tag{6}$$

ただし N は各レシピデータで使用する単語の数である. $word2vec(w_k)$ は Word2Vec による w_k の分散表現であり, $tfidf_{k,j}$ はレシピデータ r_j から抽出された単語 w_k の tf-idf 値である. 食材情報の学習は, この食材ベクトルを推定するタスクとして実現される. この食材ベクトル推定タスクでは 4096 次元の fc7 層と, 食材ベクトルの次元数のユニットで構成される出力層をもつ. ある画像 x を入力したときの出力層のユニット i の出力値を y_i, y_i に対する正解値を g_i とすると, 食材ベクトル推定タスクの損失関数 L_{ing} は次のように表される.

$$L_{ing} = -\frac{1}{2}\sum_{k=1}^{n}(g_k - y_k)^2 \tag{7}$$

3.3 調理手順情報の学習

調理手順に関しても食材情報の学習と同様に, Word2Vec [11] による単語の分散表現を使用して各レシピデータの調理手順文章中の単語を低次元実数ベクトルに変換し, それを調理手順情報の学習のための教師データの作成に利用する. 最終的に各レシピの調理手順文章は実数ベクトルに変換され, 教師データとして学習に利用される. 調理手順文章中の名詞, 動詞, 形容詞のみを使用し, tf-idf 値の高い単語を積極的に使用する. 本実験では各レシピデータの調理手順文章において tf-idf 値の上位 N_{max} 個までの単語のみを利用する. N_{max} は 1 レシピデータの調理手順文章から抽出される単語の数の平均値とする. こうして得られた単語の分散表現と tf-idf 値から各レシピデータの調理手順ベクトルを生成する. あるレシピデータ r_j の調理手順文章中の単語を w_i とすると, レシピデータ r_j の調理手順ベクトル v_j は式 (6) により得られる. 調理手順の学習は, この調理手順ベクトルを推定するタスクとして実現される. この調理手順ベクトル推定タスクは 4096 次元の fc7 層と, 調理ベクトルの次元数のユニットで構成される出力層をもつ. 損失関数 L_{dir} として式 (7) の 2 乗和誤差を使用する.

4. カロリー量情報付き食事画像データセットの構築

現時点ではカロリー量情報付きの大規模な食事画像データセットは公開されておらず, 手作業での収集やクラウドソーシングの使用はコストがかかるため, Web 上のカロリー量情報付きレシピサイトからカロリー量情報付き食事画像を収集する. 本実験では日本語と英語の 2 種類のデータセットを構築する.

4.1 日本語のカロリー量情報付き食事画像データセット

日本語のデータセットではカロリー量情報を提供する 6 つのレシピ情報サイト (レシピ大百科 [*3], E・レシピ [*4], ホームクッキング [*5], みんなのきょうの料理 [*6], オレンジページ net[*7], レタスクラブニュース [*8]) から合わせて約 83,000 件のレシピ情報を収集した. これらのサイトで公開されているレシピ情報は, 調理師や料理研究家などの専門家が提供したものとなっている. 図 3 のようにレシピ情報ページには食事画像, カロリー量に加えて, 必ず食材情報と調理手順情報が含まれている. 収集したデータを観察すると, 食事画像の多くは 1 種類の食品の画像であり, カロリー量情報の多くは 1 人分の値であることがわかった. したがって本研究では, 1 種類の食品が写ったシングルラベルの食事画像を入力とし, 1 人分のカロリー量の値を推定

[*3] http://park.ajinomoto.co.jp/
[*4] http://erecipe.woman.excite.co.jp/
[*5] https://www.kikkoman.co.jp/homecook/
[*6] http://www.kyounoryouri.jp/
[*7] http://www.orangepage.net/
[*8] http://www.lettuceclub.net/recipe/

する．

図 3　レシピ情報ページの例．

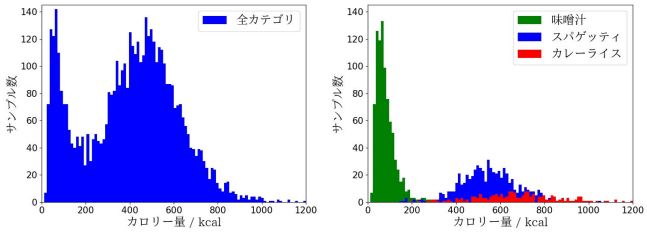
本研究では収集した画像に食事カテゴリの情報をアノテーションする必要があるため，今回は UEC Food-100 食事画像データセット [9] の 100 種類の食事カテゴリについてラベリングを行う．UEC Food-100 食事画像データセットは主に日本の食品に関するデータセットであり，カロリー量の情報はアノテーションされていない．図 4 に UEC Food-100 食事画像データセットの食事 100 カテゴリの一覧を示す．

図 4　UEC Food-100[9] の食事 100 カテゴリ．

また，本実験では低解像度画像，複数の種類の食品が写る画像，付随するカロリー量が 1 人分の値であると断定できない画像をノイズとして除去し，その後サンプル数が 100 枚以下になった食事カテゴリを除いた．最終的に総画像枚数 4877 枚，食事 15 カテゴリのカロリー量情報付き食事画像データセットが完成した．図 5 にカロリー量情報付き食事画像データセットの食事 15 カテゴリを示す．図 6 (a) にデータセット全体のカロリー量の分布を示す．味噌汁の画像が多いために 100kcal 以下のサンプル数が多くなっているが，おおよそ 500kcal 付近の食品が多いことがわかった．また，1000kcal 以上のサンプルはほとんどなかった．図 6 (b) に味噌汁，スパゲッティ，カレーライスに関するカロリー量の分布を示す．食事カテゴリとカロリー量との間に相関があり，また，同じ食事カテゴリであってもばらつきがあることがわかる．

*10 http://allrecipes.com/

図 5　日本語のカロリー量情報付き食事画像データセットの食事 15 カテゴリ．



(a) 全 15 カテゴリ．　(b) 味噌汁，スパゲッティ，カレーライス．

図 6　日本語のカロリー量情報付き食事画像データセットのカロリー量の分布．

4.2　英語のカロリー量情報付き食事画像データセット

英語のデータセットでは Allrecipes*10 から約 24,000 件のレシピ情報を収集した．Allrecipes はユーザ投稿型のレシピサイトであり，各レシピから 1 人分のカロリー量が得られる．食事カテゴリのアノテーションでは，Allrecipes で使用されているカテゴリを使用した．また，低解像度画像，複数の種類の食品が写る画像をノイズとして除去し，最終的に総画像枚数 2484 枚，図 7 の食事 21 カテゴリのカロリー量情報付き食事画像データセットが完成した．日本語のカロリー量情報付き食事画像データセットと比較して，視覚的に類似する食品が多く含まれる．

図 7　英語のカロリー量情報付き食事画像データセットの食事 21 カテゴリ．

5. 実験

本研究では VGG16 [16] を拡張し, 図 2 のような Multi-task CNN を使用する. Dropout [17] は用いず, fc6 層と fc7 層において Batch Normalization [7] を使用し, すべての層が畳み込み層により実装される. Batch Normalization 層と出力層以外の層では ImageNet の 1000 種類分類タスクの pre-train モデルを初期値として利用する. バッチサイズは 8 とし, 最適化手法として SGD を使用し, Momentum 値は 0.9 とする. 式 1 と式 2 の損失項にかかる重みは事前に決定する必要があるが, 本実験では同時に行う全てのタスクの損失項にかかる重みを 1 に設定した状態で一度学習を行い, そのとき各イテレーションで得られる損失の値をタスクごとに保持しておき, 最終的に全イテレーションにおける損失の値の平均値の逆数を各タスクの損失項にかかる重みとして使用する. ただし本実験では, $\lambda_{re} = 1$ と固定した.

5.1 カロリー量推定の損失関数の決定

式 (2) の絶対誤差と相対誤差を組み合わせた損失関数の有効性を検証するために, 損失関数としてそれぞれの誤差を単独で用いた場合との比較を行った. 実験には 4.1 章の日本語のレシピサイトから収集したカロリー量情報付き食事画像データセットを使用した. 学習に 70%を使用し, テストには残りの 30%を使用した. 学習率 0.001 において 50k イテレーション, さらに 0.0001 において 20k イテレーション学習した.

テストデータを用いてカロリー量の推定を行った結果を表 1 にまとめた. テストには, 学習時に最後の 1k イテレーションから 100 イテレーション間隔で得られた 10 個のモデルを使用し, 各モデルから得られた推定値の平均値を最終的な推定値とした. 評価指標として相対誤差, 絶対誤差, 相関係数, 相対誤差 20%以内の推定値の割合を用いた. 絶対誤差や相対誤差 20%以内の推定値の割合を見ると, 相対誤差と絶対誤差の両方を使用することで精度が向上したことがわかる. したがって本研究ではカロリー量推定の損失関数として式 (2) を使用する.

表 1 カロリー量推定の損失関数の比較.

	相対誤差 (%)	絶対誤差 (kcal)	相関係数	誤差 20%以内 (%)
相対誤差	**29.4**	105.9	0.776	42.0
絶対誤差	59.8	134.9	0.589	36.7
相対誤差 + 絶対誤差	**29.4**	**100.7**	**0.778**	**45.9**

5.2 日本語のレシピサイトから収集したデータセットでのカロリー量推定

実験には 4.1 章の日本語のレシピサイトから収集したカロリー量情報付き食事画像データセットを使用した. 学習に 70%を使用し, テストには残りの 30%を使用した. 学習率 0.001 において 50k イテレーション, さらに 0.0001 において 20k イテレーション学習した. 食材ベクトルと調理手順ベクトルの作成のために, クックパッドのレシピデータセット [*9] の調理手順文章約 8,710,000 文を使用して Word2Vec の学習を行った. 単語の分散ベクトルの次元は $n = 500$ とした. 学習に使用するレシピデータの食材情報に関しては $N_{max} = 12$ であったため, 各レシピデータにおいて tf-idf 値の上位 12 個までの食材名の単語のみを利用し, 式 (6) により食材ベクトルを作成した. 調理手順情報に関しては $N_{max} = 44$ であったため, 各レシピデータの調理手順文章において tf-idf 値の上位 44 個までの単語のみを利用し, 式 (6) により調理手順ベクトルを作成した.

テストデータを用いてカロリー量の推定を行った結果を表 2 にまとめた. テストには, 学習時に最後の 1k イテレーションから 100 イテレーション間隔で得られた 10 個のモデルを使用し, 各モデルから得られた推定値の平均値を最終的な推定値とした. カロリー量推定では評価指標として相対誤差, 絶対誤差, 相関係数, 相対誤差 20%以内の推定値の割合を用い, 食事カテゴリを同時に学習する Multi-task CNN では正解分類率 Top-1 を示した. シングルタスクとマルチタスクとを比較すると, どの評価指標においてもマルチタスクにより性能が向上したことがわかる. 全タスクでのマルチタスクの場合では, シングルタスクに比べて相対誤差が–2.0%, 絶対誤差が–9.5kcal, 相関係数が+0.039, 相対誤差 20%以内の割合が+4.2%となり, 食事カテゴリ分類においては正解分類率が+2.9%となり改善が見られた. 図 8, 図 9 にカロリー量推定の推定値と正解値の相関関係を示す. 図 9 は全タスクでのマルチタスクの結果である. 図 8 と図 9 を比較すると, 95%信頼楕円などからマルチタスクにより精度が向上していることがわかる. 図 10, 図 11 に成功例と失敗例を示す. 失敗例をいくつか見ると, 比較的に高いカロリー量の食品を低く推定してしまうものが多かった. これは学習データに比較的に高カロリー量のサンプルが不足していることが原因ではないかと考えられる.

表 2 日本語のカロリー量情報付き食事画像データセットでの推定結果.

	相対誤差 (%)	絶対誤差 (kcal)	相関係数	誤差 20%以内 (%)	Top-1(%)
calorie(single)	29.4	100.7	0.778	45.9	—
+categories	27.9	95.2	0.802	48.8	82.8
++ingredients	27.6	94.4	0.811	49.5	**85.2**
+++directions	27.4	**91.2**	**0.817**	**50.1**	84.1
+ingredients	29.2	96.8	0.795	46.8	—
++directions	28.0	97.9	0.806	47.2	—
+directions	28.2	95.5	0.808	48.1	—
++categories	**27.3**	96.0	0.808	48.8	84.8
categories(single)	—	—	—	—	81.2

[*9] http://www.nii.ac.jp/dsc/idr/cookpad/cookpad.html

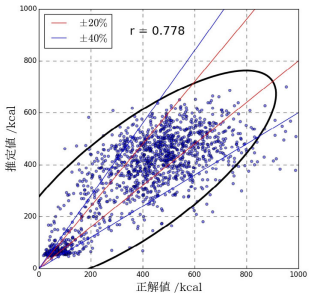


図 8　推定値と正解値の相関関係 (single-task).

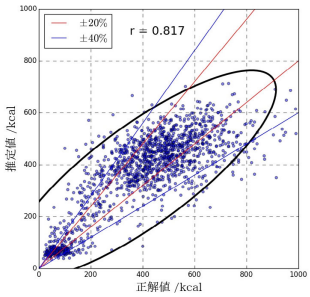


図 9　推定値と正解値の相関関係 (multi-task).

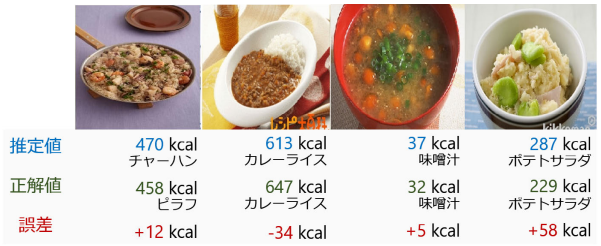


図 10　カロリー量推定成功例.

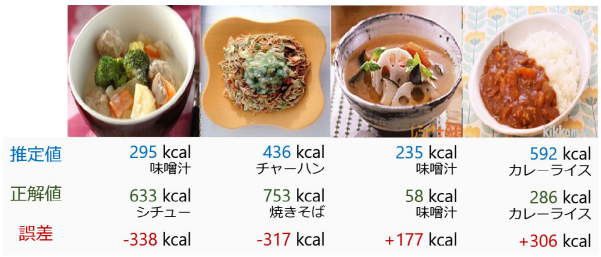


図 11　カロリー量推定失敗例.

5.3　英語のレシピサイトから収集したデータセットでのカロリー量推定

4.2 章の英語のレシピサイトから収集したカロリー量情報付き食事画像データセットを使用し, 5.2 章と同様の実験を行った. 学習に 80%, テストに残りの 20%を使用した. 学習率 0.001 において 30k イテレーション, さらに 0.0001 において 10k イテレーション学習した. 食材ベクトルと調理手順ベクトルの作成のために, 4.2 章のデータセットの調理手順文章約 82,000 文を使用して Word2Vec の学習を行った. 単語の分散ベクトルの次元は $n = 500$ とした. 学習に使用するレシピデータの食材情報に関しては $N_{max} = 26$ であったため, 各レシピデータにおいて tf-idf 値の上位 26 個までの食材名の単語のみを利用し, 式 (6) により食材ベクトルを作成した. 調理手順情報に関しては $N_{max} = 66$ であったため, 各レシピデータの調理手順文章において tf-idf 値の上位 66 個までの単語のみを利用し, 式 (6) により調理手順ベクトルを作成した.

5-fold cross-validation を行い, 各評価指標の値の平均値を表 3 にまとめた. シングルタスクとマルチタスクとを比較すると, どの評価指標においてもマルチタスクにより性能が向上したことがわかる. 食事カテゴリと食材情報のマルチタスクの場合では, シングルタスクに比べて相対誤差が–1.6%, 絶対誤差が–8.9kcal, 相関係数が+0.09, 相対誤差 20%以内の割合が+2.9%となり, 食事カテゴリ分類においては正解分類率が+6.7%となり改善が見られた. 5.2 章の結果と同様に, マルチタスクの優位性を確認することができた.

表 3　英語のカロリー量情報付き食事画像データセットでの推定結果.

	相対誤差 (%)	絶対誤差 (kcal)	相関係数	誤差 20%以内 (%)	Top-1(%)
calorie(single)	43.3	128.5	0.293	32.2	—
+categories	42.9	120.5	0.361	34.3	58.7
++ingredients	**41.7**	**119.6**	**0.383**	**35.1**	61.1
+++directions	42.9	120.6	0.369	33.7	**61.3**
+ingredients	43.0	124.2	0.335	32.5	—
++directions	42.1	122.5	0.351	32.5	—
+directions	42.0	123.0	0.349	33.6	—
++categories	42.4	120.8	0.365	33.5	59.3
categories(single)	—	—	—	—	54.4

5.4　既存手法との参考比較

次に宮崎らの研究 [12] との比較を行った. ただし使用したデータセットが異なるため参考比較である. 宮崎ら [12] は本論文と同様に食事画像からカロリー量を直接推定した. まず食事画像から color histogram や SURF 特徴量を抽出し, 各特徴量に基づいて, 辞書データから類似画像上位 5 位のカロリー量の平均値を計算する. 最後に計算した平均値に基づき, 線形予測によりカロリー量を推定する. 食品の量は考慮しておらず, 本論文と同様に 1 人分のカロリー量を推定している. データセットには Web サービスである FoodLog[*10] に投稿された食事画像 6512 枚を使用し, 栄養学の知識を持った複数の専門家が食事画像にカロリー量をアノテーションしている. 注意したいのは, 本論文では 1 種類の食品を写したシングルラベル画像を対象にしているが, [12] では複数の食品を含む画像もあるということである. さらに本論文では 15 カテゴリを対象としているが, [12] ではそのような制限はないことも考慮しなければいけない.

Baseline と比較すると, 日本語のデータセットでの全タスクでのマルチタスクの結果では, 相対誤差 40%以内に含まれる推定値の割合はほぼ同じであるが, 相対誤差 20%以内に含まれる推定値の割合は+13%向上し, 相関係数に関しても+0.5 と大幅な改善が見られた. 英語のデータセットでの食事カテゴリと食材情報のマルチタスクの結果では, Baseline と比較して相関係数が+0.06 となった.

6.　おわりに

本研究では, 入力として食事画像を受け取りカロリー量の値を直接出力する CNN を学習することで, 食事画像中

[*10] http://www.foodlog.jp/

表 4　宮崎ら [12] の手法との比較

	相対誤差	誤差 20%以内 (%)	誤差 40%以内 (%)
Baseline	0.32	35	79
Multi-task (日本語)	0.82	49	80
Multi-task (英語)	0.38	35	65

の食品の外見を直接反映するようなカロリー量の推定を行った. さらに Multi-task CNN により, カロリー量に加えて食事カテゴリや食材, 調理手順などのカロリー量と相関がある情報を同時に学習することで, より高精度のカロリー量推定の実現を試みた. また, Web 上のレシピサイトからカロリー量情報付き食事画像を収集し, 日本語のレシピサイトから収集したデータセットと, 英語のレシピサイトから収集したデータセットを作成し実験を行った. 実験ではシングルタスクとマルチタスクとの比較を行い, 両方のデータセットにおいてマルチタスクによりカロリー量推定と食事カテゴリ分類の性能が向上することを確認した.

本研究では Web 上のレシピサイトのカロリー量情報を正解値として学習とテストを行ったが, このカロリー量の正確性を保証することはできず, 誤った値も多く含まれると考えられるため, このデータセットをもとに高精度なカロリー量推定を行うことは困難であると考えられる. そんな中, マルチタスクによりカロリー量推定と食事カテゴリ分類の両方のタスクの性能が向上するという結果が得られたことはデータセットに関係なく有益であったと考えられる. 高精度のカロリー量推定の実現のためには, 良質なデータセットの作成が急務であると考えられる.

今後の課題としては, 食品の量を考慮したカロリー量推定とデータセットの構築などがある. 食品領域の検出や領域分割, さらに [13] や [15] のように予め基準物体を設けるなどして, 写真中の食品の量を推定することなどが考えられる. また, [8] や [6] のように複数視点からの画像や奥行き推定を行うことで三次元情報を考慮することも考えられる.

謝辞 本研究は科研費 (17H01745,17H06026) の助成を受けたものである.

参考文献

[1] H. A. Abrar, W. Gang, L. Jiwen, and J. Kui. Multi-task CNN model for attribute prediction. *IEEE Transactions on Multimedia*, 17(11):1949–1959, 2015.

[2] V. Bettadapura, E. Thomaz, A. Parnami, D. G. Abowd, and A. Essa. Leveraging context to support automated food recognition in restaurant. In *Proc. of the 2015 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2015.

[3] L. Bossard, M. Guillaumin, and L. Van Gool. Food-101 – mining discriminative components with random forests. In *Proc. of European Conference on Computer Vision*, 2014.

[4] J. J. Chen and C. W. Ngo. Deep-based ingredient recognition for cooking recipe retrival. In *Proc. of ACM International Conference Multimedia*, 2016.

[5] M. Chen, Y. Yang, C. Ho, S. Wang, E. Liu, E. Chang, C. Yeh, and M. Ouhyoung. Automatic chinese food identification and quantity estimation. In *Proc. of SIGGRAPH Asia Technical Briefs*, page 29, 2012.

[6] J. Dehais, M. Anthimopoulos, and S. Mougiakakou. Go-carb: A smartphone application for automatic assessment of carbohydrate intake. In *Proc. of ACM MM Workshop on Multimedia Assisted Dietary Management*, 2016.

[7] S. Ioffe and C. Szegedy. Batch Normalization: Accelerating deep network training by reducing internal covariate shift. In *Proc. of International Conference on Machine Learning*, 2015.

[8] F. Kong and J. Tan. Dietcam: Automatic dietary assessment with mobile camera phones. In *Proc. of Pervasive and Mobile Computin*, pages 147–163, 2012.

[9] Y. Matsuda, H. Hajime, and K. Yanai. Recognition of multiple-food images by detecting candidate regions. In *Proc. of IEEE International Conference on Multimedia and Expo*, 2012.

[10] A. Meyers, N. Johnston, V. Rathod, A. Korattikara, A. Gorban, N. Silberman, S. Guadarrama, G. Papandreou, J. Huang, and P. K. Murphy. Im2calories: towards an automated mobile vision food diary. In *Proc. of IEEE International Conference on Computer Vision*, 2015.

[11] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, 2013.

[12] T. Miyazaki, G. Chaminda, D. Silva, and K. Aizawa. Image - based calorie content estimation for dietary assessment. In *Proc. of IEEE ISM Workshop on Multimedia for Cooking and Eating Activities*, 2011.

[13] K. Okamoto and K. Yanai. An automatic calorie estimation system of food images on a smartphone. In *Proc. of ACM MM Workshop on Multimedia Assisted Dietary Management*, 2016.

[14] P. Pouladzadeh, S. Shirmohammadi, and R. Almaghrabi. Measuring calorie and nutrition from food image. In *IEEE Transactions on Instrumentation and Measurement*, pages 1947–1956, 2014.

[15] W. Shimoda and K. Yanai. CNN-based food image segmentation without pixel-wise annotation. In *Proc. of IAPR International Conference on Image Analysis and Processing*, 2015.

[16] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *arXiv preprint arXiv:1409.1556*, 2014.

[17] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958, 2014.

[18] R. Tanno, K. Okamoto, and K. Yanai. Deepfoodcam: A dcnn-based real-time mobile food recognition system. In *Proc. of ACM MM Workshop on Multimedia Assisted Dietary Management*, 2016.

[19] S. Tokui, K. Oono, S. Hido, and J. Clayton. Chainer: a next-generation open source framework for deep learning. In *Proceedings of Workshop on Machine Learning Systems (LearningSys) in The Twenty-ninth Annual Conference on Neural Information Processing Systems (NIPS)*, 2015.