

深度付き画像と深層学習による食事カロリー量推定システムの開発

安藤 祥和[†] 會下 拓実^{††} 柳井 啓司^{††}

[†] 電気通信大学 情報理工学部 総合情報学科 メディア情報学専攻

^{††} 電気通信大学 大学院情報理工学研究科 情報学専攻

E-mail: [†]{ando-y, ege-t, yanai}@mm.inf.uec.ac.jp

あらまし 近年、スマートフォンなどの携帯端末の普及と、健康意識の向上により、端末での健康管理の需要は高まっている。端末での食事内容やカロリーの記録もそういった需要の一つであり、様々なアプリが配信されているが、利便性を追求したことで精度が低くなっているものや、利便性に疑問があるものも少なくない。食品のカロリー量を正確に測定するためには食品の体積の推定を行うことが不可欠であるが、この推定のために既知の面積を持つ物体が必要ななどである。一方で、端末の性能や機能は年々向上し続けており、実世界の情報を認識する能力も進化している。iOSにおいては一部の端末に深度の測定が可能なカメラが搭載された。これらの状況を踏まえ、本研究では深度付き画像と深層学習を用いて携帯端末上で容易に食事のカロリー量が推定できるシステム“DepthCalorieCam”の開発を行い、実物の酢豚・唐揚げ・コロッケの3種類の食品に対して実施した評価実験において、食事画像から推定したカロリー量の相対誤差を既存手法に比べ最小で $\frac{1}{100}$ まで抑えることができた。

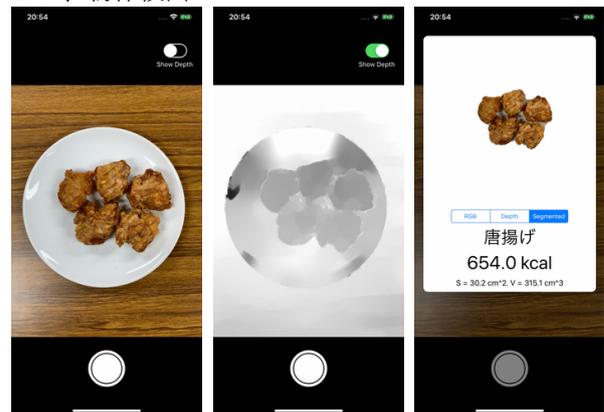
キーワード カロリー量推定、深度、深層学習、アプリケーション、物体検出

1 はじめに

近年、スマートフォンなどの携帯端末の普及と、健康意識の向上により、端末での健康管理の需要は高まっている。端末での食事内容やカロリーの記録もそういった需要の一つであり、これを満たすカロリーの記録ができるアプリには様々なものがあるが、利便性を追求したことで精度が低くなっているものや、利便性に疑問があるものも少なくない。例としては、ユーザーが一品ずつ食品の種類や量を入力しなければならず、食品に関する事前知識が必要なものや、食品の画像を撮影すると食品の種類を自動で認識してくれるものの、カロリーについては食事の量にかかわらず種類ごとに固定されているものなどがある。食品のカロリー量を正確に測定するためには食品の体積の推定を行うことが不可欠であるが、これを行っているものであっても、既知の面積を持つ基準物体が必要で、これと食事を一緒に画像に収める必要があるといった課題点もある。

一方で、端末の性能や機能は年々向上し続けており、実世界の情報を認識する能力も進化している。iOSにおいては、特定のモデルのみではあるが深度の測定が可能なカメラや、また、端末上での機械学習モデルを用いた高速な推論を可能にするNeural Engineなどである。

これらの状況を踏まえ、本研究では最近のスマートフォンに内蔵されている深度カメラで撮影した画像と深層学習を用い、スマートフォン上で基準となる物体を必要とせず体積ベースでの正確な食事のカロリー量が推定できるシステムの開発を目的とする。なお動作環境として、iOSがインストールされた、深度を取得可能なカメラを搭載した端末(iPhone 7/8 Plus, X/Xs/Xs Max)を想定する。



(a) 撮影画面 (RGB) (b) 撮影画面 (深度) (c) 結果画面

図 1: システムのイメージ画像

2 関連研究

深度情報を持たない通常の食事画像からのカロリー量推定については、まずあらかじめカロリー量がわかっている食事画像データに対する検索ベースの手法として、SIFT 特徴量などを用いた検索を行った宮崎らの研究 [1] や、検索に用いる特徴量抽出に距離学習を行った CNN を利用して精度を高めた會下らの研究 [2] がある。また食事画像から直接カロリー量を推定するものとしては、カロリー量の他にレシピ情報などで同時学習した Multi-task CNN を用いる會下らの研究 [3] がある。しかしこの手法では食事の量を考慮していないため、食事の量によってはその精度が大きく低下してしまう。本システムでは食事の体積を考慮したカロリー量の推定を行うことで食事の量によらない高い精度の推定を実現する。

深度を利用した食事画像からのカロリー量推定についてはまず Chen らの研究 [4] がある。Chen らは深度付き画像から食品

のカテゴリおよび食事の体積を推定する手法を提案しているが、文献内ではカテゴリ推定に重点を置いていて、体積の推定に関しては完成しておらず、結果の提示および評価が行われていない。また、赤外線を利用した深度カメラを用いているため、炊いた米などの赤外線の反射および屈折特性に影響を与える材料を用いた食品の深度測定が難しく、文献内では本システムで用いるステレオマッチングなどをベースとした深度カメラが必要であるとされている。

また他の深度を利用した食事画像からのカロリー量推定には Myers らの研究 [5] がある。Myers らは食事画像からの体積推定について、非公開の深度付き食事データセットで学習した CNN を用いて色情報のみの食事画像から深度を推定、さらにそこからボクセル化を行って体積を推定している。推定されたボクセルおよび深度については比較的高い精度を実現しているが、この手法では膨大な量の深度付き画像を準備して学習を行っており、そのコストは計り知れない。また文献内では食事画像の領域分割およびカテゴリ認識も行っており、また作成したシステムのモバイルアプリ化について触れているが実現しておらず、端末上ではカテゴリ認識のみにとどまっている。本システムでは実際に端末上で動作可能なシステムを作成し、ユーザーに手軽なカロリーの推定を可能にしている。

さらに本システムとの関連が特に高い、食事の量を考慮したスマートフォン上の食事画像からのカロリー量推定手法について、以下で詳細に比較する。

2.1 CalorieCam

岡元らは、食品のカロリー量を端末のカメラで撮影するだけで推定するシステムである CalorieCam を開発している [6]。CalorieCam はユーザーのカロリーや食事に関する知識に関係なく、食事の画像から自動で食品の表面積の推定と、それを元にしたカロリーの推定を可能にしている。

CalorieCam におけるカロリー量推定の流れは以下のようになっている。

- (1) 画像に Canny フィルタを適用してエッジ画像を抽出
- (2) エッジ画像から食品が乗った食器が含まれる矩形を検出
- (3) 得られた矩形内を $k = 3$ の K-means によって食品、食器、背景の 3 つの領域にクラスタリングし、食品が含まれる矩形を検出
- (4) 食品が含まれる矩形に対して GrabCut [7] を用いて食品が写っているピクセルを検出
- (5) 基準物体が含まれるように指定されている矩形内に対して GrabCut を用いて基準物体が写っているピクセルを検出
- (6) 2 で得られた矩形内から深層学習モデルを用いて食品の種類を推定
- (7) 4、5 で検出したピクセル数の比と既知の基準物体の面積から、以下の式 1 で食品の表面積 S_{food} を計算

$$r_s = \frac{(\text{食品のピクセル数})}{(\text{基準物体のピクセル数})}$$

$$S_{food} = r_s \times (\text{基準物体の表面積}) \quad (1)$$

- (8) 6 で推定した食品の種類、7 で計算した食品の面積、あらかじめ計測されている食品の種類ごとの表面積あたりのカロ

リー量をかけ合わせてカロリー量を計算

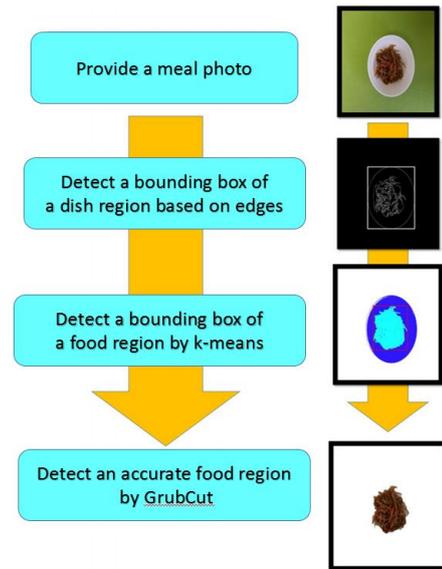


図 2: CalorieCam の食品領域の抽出手順 ([6] から引用)

以上の流れからもわかるように、CalorieCam ではユーザーにカロリーに関する知識を要求しない代わりに、すでに表面積がわかっている基準物体が必要である。また画像中の食事や基準物体の高さを考慮せずにピクセル数のみでその表面積を推定するため、食事が複雑な形状をしていたり、基準物体と食事のカメラからの距離が異なる場合に表面積の推定精度が低下するという問題がある。本システムでは深度付き画像を利用することで、基準物体を必要とせず、また食事の高さや体積を考慮したカロリーの推定を可能にする。

2.2 AR DeepCalorieCam V2

基準物体が必要という点について、丹野らは CalorieCam のこれを不要とし、AR によって測定した物体の実寸に基づくカロリーの測定を可能とした AR DeepCalorieCam V2 を開発している [8]。これによってユーザーは携帯端末のみで食品のカロリーを推定できるようになった。

AR DeepCalorieCam V2 のカロリー量推定の流れは以下のようになっている。

- (1) 画面内に単品の食品を写した状態で撮影を行い、その画像を Inception-v3 [9] で認識し食品カテゴリを推定
- (2) 表面積の計測モードに切り替え、端末を動かしながら AR 空間内で食品にアンカーを設置
- (3) アンカーから食品の表面積を計算し、1 の食品カテゴリごとの事前に計算した回帰式をもとにカロリーを推定
- (4) 推定したカロリーを AR 空間内に表示
- (5) 食品の数だけ 1~4 を繰り返し

前述のように AR を利用することで CalorieCam が抱えていた基準物体が必要である点を解消したが、一方で AR の特性上、システムを起動してから空間の特徴点を検出するために端末を動かす必要があったり、特徴点を認識した後も図 3 のようにユーザーが端末を動かしながら実際の食品上に AR 空間内でアンカーを設置しなければならず、CalorieCam とはまた

違う課題点が生じている。また、AR DeepCalorieCam V2 についても、食品の体積ではなく、ユーザーが指定した食品のアンカーから計算した表面積を元にカロリーの推定を行っているため、精度は体積ベースの推定より劣る。本システムでは、深度付き画像のみで画像中の食品の形状を認識することで、ユーザーが空間の特徴点が検出されるまで端末を動かす必要をなくし、またカロリーの推定の精度についてもユーザーの操作に依存しない推定を可能とする。



図 3: AR DeepCalorieCam V2 の食品領域指定の様子 ([8] から引用)

3 深度カメラ

まず、本システムで利用する深度カメラの原理について説明する。先述の通り、本システムでは以下に挙げた端末での利用を想定している。

- iPhone 7 Plus
- iPhone 8 Plus
- iPhone X
- iPhone XS
- iPhone XS Max

これらの端末は望遠レンズと広角レンズを備えた 2 つのバックカメラを搭載している。本システムではこの 2 つのカメラの情報からなる視差 (Disparity) ベースの深度カメラを利用する。

視差ベースの深度カメラの原理を説明するに先立って、まず簡単なピンホールカメラの原理について説明する。図 4 にピンホールの簡単な図を示した。ピンホールカメラは箱の一箇所に小さな穴 (ピンホール) を開け、その反対側に光を感知するセンサー (ピンホールカメラの場合はフィルムなど) を設置し、ピンホールから直進してきた光をセンサーで記録することで撮影を行うカメラである。実際のカメラにはレンズなどの部品が付くことでカメラの構造が複雑になるが、深度カメラの説明のために一般のカメラもピンホールカメラのような原理で撮影を行うとする。ピンホールを通過した光は直進してセンサーにぶつかるため、光が入ってきた方向はわかるが、1 つのカメラのみではその先にある物体までの距離はわからない。

ここで焦点距離が等しいカメラが 2 つある場合を考える。このとき図 5 のように、基準となるカメラから物体までの距離を深度 z 、2 つのカメラの中心間の距離をカメラ間距離 c 、焦点距離を f 、各センサ上での物体間の距離を視差 d と呼ぶ。

そして図中に赤線で示した部分に注目すると高さ z で幅 c の三角形と高さ f で幅 d の三角形が相似であることがわかる。こ

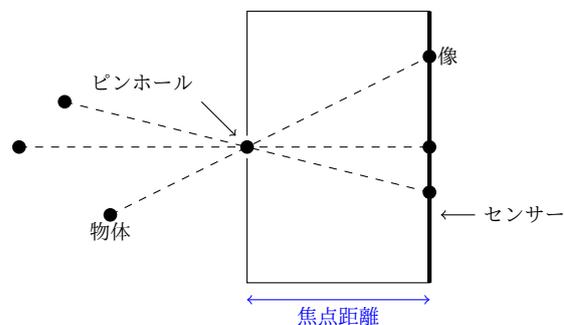


図 4: ピンホールカメラ

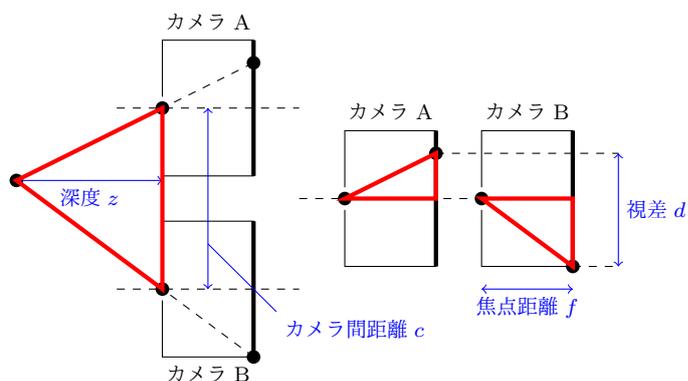


図 5: デュアルカメラによる深度の計測

のことより深度 z は c 、 f 、 d を用いて以下の式 2 で表される。

$$z = \frac{fc}{d} \quad (2)$$

以上が本システムで利用する深度カメラの原理である。

また、このようなカメラとは別に iPhone X/XR/XS/XS Max には TrueDepth カメラと呼ばれる、光を物体に照射して跳ね返ってくるまでの時間を計測することで物体までの距離を高精度に測ることのできるカメラが前面に搭載されているが、

- 前面のカメラで撮影を行うと、撮影中にディスプレイが見にくく現実的でないこと
- 動作する端末が比較的新しい機種のみ絞られてしまうこと

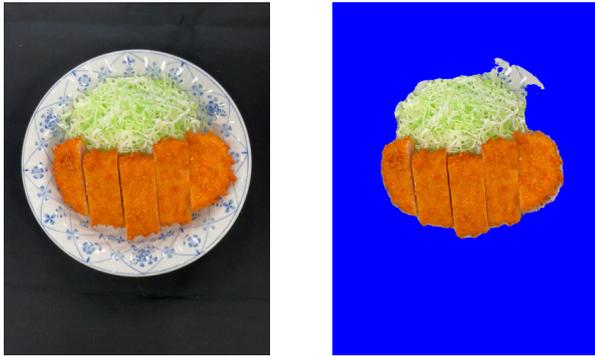
などから今回は対象としない。しかし本システムの構造上、カロリーの推定は可能である。

4 提案手法

4.1 食品領域分割

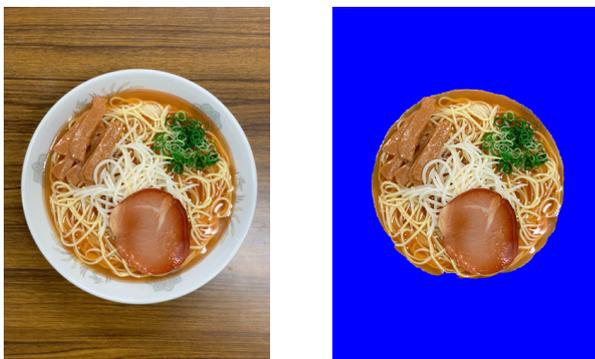
食事画像からのカロリー量推定のためには、ピクセル単位での食品領域の分割が不可欠であるが、本システムでは深層学習の領域分割モデルを用いて分割を行った。モデルには基本的な領域分割モデルである U-Net [10] を使い、データセットには UECFOOD-100 [11] に研究室内で追加で食事・非食事の領域アノテーションをつけた画像 5301 枚を利用した。学習時には 4771 枚を学習画像、530 枚をテスト画像とし、最終的にテスト画像での分割性能が最も高かった時点の重みをシステムで利用している。このモデルにおける、領域分割タスクで一般的に用いられる評価指標である mIoU (mean Intersection over

Union) をテスト画像について算出したスコアは 0.800 だった。このモデルで食品サンプルを撮影した画像を領域分割した例を図 6、7 に示す。複雑な形状の食品でも高精度に領域分割できていることがわかる。



(a) 元画像 (b) 領域分割結果

図 6: U-Net による領域分割例



(a) 元画像 (b) 領域分割結果

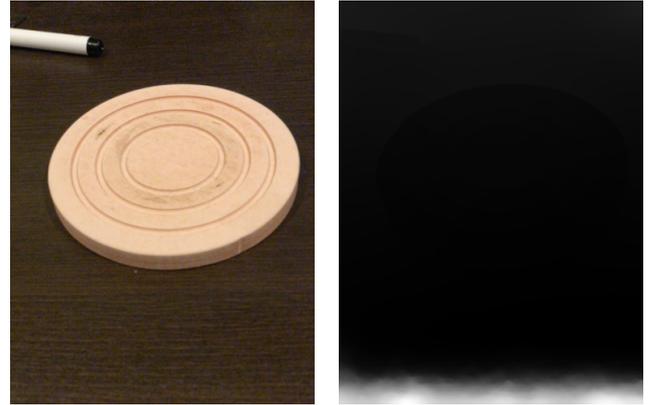
図 7: U-Net による領域分割例

4.2 深度情報の較正

深度カメラから取得した深度画像はカメラから対象とする各点までの距離を格納した実数値からなる画像だが、その深度情報は異常値や全体的な誤差を含んでいる。そのため、次章で説明する体積推定の前段階として、深度情報に対する較正を行う。

まず異常値について、具体的には画像周辺部の深度が異常に大きな値を示すことがあった。例を図 8 に示す。深度画像では大きな異常値によって本来の物体の形状が確認できないほどに深度差が潰れてしまっていることがわかる。これについては 3 章で示した今回利用している深度カメラの仕組み上、画像周辺部では 2 つのカメラ間で対応が取れないことがあり、それがこのような異常値の原因となっていると考えられる。ここで図 8 の深度画像の深度の値を昇順に並べてプロットしたものを図 9 に示す。図 9 を見ると、図 8 の場合は 90 パーセントイルを超えたあたりから深度の値が急に大きくなっていることがわかる。例示した場合以外も大方 80 パーセントイル以上の範囲で値に以上が見られることが多かったため、深度の 80 パーセントイル値以上の値をすべて 80 パーセントイル値に置き換えることで異常値の補正とした。図 8 の深度画像に同様の補正を施した

ものを図 10 に示す。補正後は画像中の物体の形が目視でも確認できるようになったことがわかる。



(a) 色情報画像 (b) 深度画像 (値が 0~255 になるように補正)

図 8: 画像周辺部の異常値

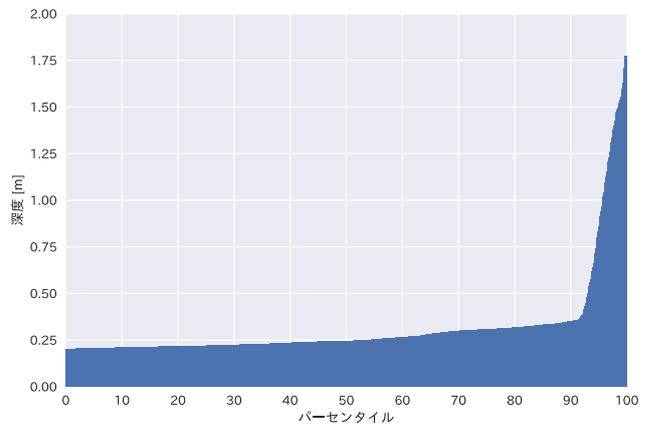


図 9: 図 8 の場合の端末から得られた深度値の分布

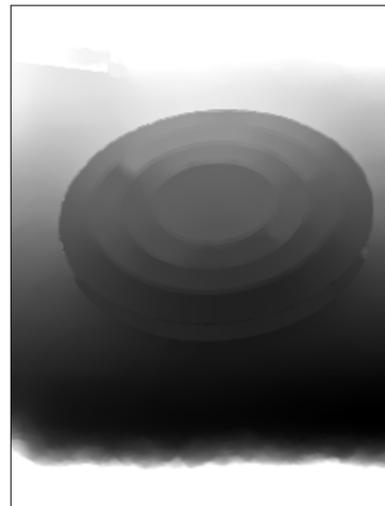


図 10: 異常値を補正した深度画像 (値が 0~255 になるように補正。黒棒は見やすさのため)

また全体的な誤差については 5.1 章の実験で作成した回帰式を用いて深度画像全体の深度を回帰することで較正を行った。

4.3 深度付き画像からの実寸の推定

深度付き画像から物体の実寸を推定する前段階として、カメラから物体が設置されている基準面までの距離を推定する。なお、実寸の推定にあたって、本システムは以下のような環境での動作を想定している。

- 基準面は平ら
- 基準面と端末が並行
- 画像に写っている範囲内で基準面の深度が最大

しかし、図 11 のように物体が基準面を覆っている場合、目的の距離 CR は深度付き画像からは読み取ることができない。そこで本システムでは、深度付き内の上下左右の各辺の中点 R_N 、 R_S 、 R_E 、 R_W の深度とカメラから取得できるその画角 (図 11 中の $\angle R_NCR_S$ 、 $\angle R_ECR_W$ にあたる) を元に、各中点 R_X ごとに以下の式 3 で計算した距離の平均を基準面までの距離として扱う。

$$CR = CR_X \cos \angle RCR_X \quad (3)$$

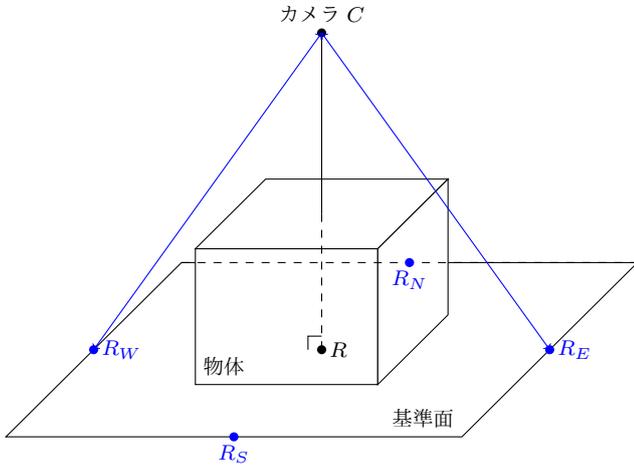


図 11: 基準面までの距離の推定

次に深度付き画像に写っている基準面での XY 方向の実寸 (図 11 中の R_NR_S 、 R_ER_W) と画像内での 1 ピクセルあたりの実面積 S_{pixel} を求める。これらはカメラの画角と先ほど計算したカメラから基準面まで距離 CR 、深度付き画像の全ピクセル数 N_{pixel} を用いて以下の式 4~6 で表される。

$$R_NR_S = 2CR \tan \frac{\angle R_NCR_S}{2} \quad (4)$$

$$R_ER_W = 2CR \tan \frac{\angle R_ECR_W}{2} \quad (5)$$

$$S_{pixel} = \frac{R_NR_S \times R_ER_W}{N_{pixel}} \quad (6)$$

そして最後に 4.1 章で求めた食事領域 P と組み合わせ、その体積を計算する。それぞれの値の関係は図 12 のようになる。まずそのピクセルに写っている面が基準面に比べ、どれだけ大きく写っているかを表す拡大率 r_p を、食事領域の各ピクセルの深度 z_p と基準面の深度 $z_{ref} = CR$ を用いて以下の式 7 のように定める。

$$r_p = \left(\frac{z_{ref}}{z_p} \right)^2 \quad (p \in P) \quad (7)$$

これを用いて最終的な体積 V を以下の式 8 で求める。

$$V = \sum_{p \in P} \frac{S_{pixel}}{r_p} (z_{ref} - z_p) \quad (8)$$

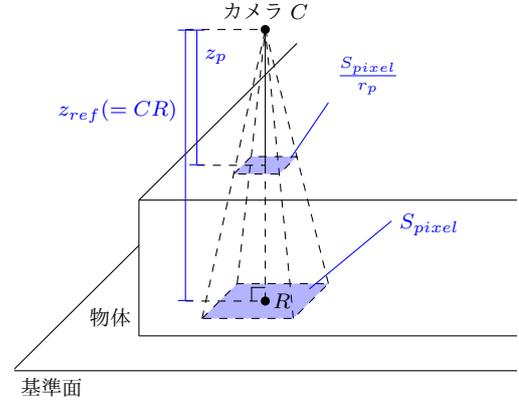


図 12: 体積の推定

なお、この方法で計測できる体積は物体がカメラから見える表面を天面とし、基準面までのどの高さにおいても同じ形状の断面を持つと仮定した際の体積である。例えば、図 13 のような形状の物体については同図の青い部分の体積となる。食品のような複雑な形状をもつ物体を対象にした場合は真の体積との誤差が大きくなることが予想されるが、後述する回帰などにおいてこの方法で計測された体積を元にしてカロリー量推定の精度を確保する。

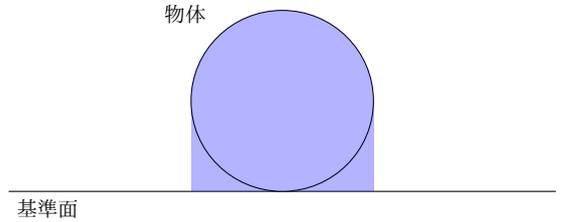


図 13: 本手法で計測する体積 (真横からの図)

4.4 食事カテゴリーの推定

本システムにおいてはカロリー量は後述するカテゴリーごとの事前に計算した回帰式をもとに推定するため CNN を用いた食事カテゴリー認識を行う。モデルは一般的にはモバイル端末上で動作させるにはモデルサイズが大きいのが、認識精度が優れている Xception [12] を利用した。重みは ImageNet で学習したものを 101 種類の食事画像データセットである Food-101 [13] で Fine-tuning し、さらに後述する実験で利用した食品カテゴリーの画像を UECFOOD-100 [11] から抽出して Fine-tuning したものを利用している。

4.5 食品のカロリー量の推定の流れ

最終的なカロリー量は以下の流れで推定する。

- (1) 4.1 章の手法で食品領域を抽出
- (2) 抽出された領域の各ピクセルごとにその深度と 4.3 章の手法で推定した XY 方向の実寸からその体積を推定
- (3) 4.4 章の手法で推定した食品カテゴリごとの体積からカロリー量の回帰式でカロリー量を推定

5 実験

5.1 深度カメラの精度に関する実験

4.2 章で述べたように、深度カメラから取得できる深度情報には全体的に誤差を含んでおり、その較正のために深度カメラの精度に関する実験を行った。実験はスマートフォン本体を水平に保った状態で同じく水平な基準面を深度カメラで撮影し、実際の本体と基準面の距離と深度付き画像の中央に記録された深度を比較することでその精度の実験とした。撮影はシステムが実際に利用されると想定される、基準面との距離が 20~70cm ほどの範囲内の複数の位置で行った。

実験で得られた深度カメラによる基準面との距離と実際に測定した基準面との距離のグラフを図 14 に示した。深度カメラによる測定値との誤差の平均は 5.684 [cm]、標準偏差は 4.543 [cm] であった。較正に利用する深度カメラによる基準面との距離を説明変数 $d_{measured}$ 、実際の基準面との距離を目的変数 d_{true} とした回帰式は以下の式 9 のようになった。この回帰式の決定係数は $R^2 = 0.8789$ であった。図 14 中の直線はこの回帰式で表される直線である。

$$d_{true} = 1.063 \times d_{measured} + 3.243 \quad (9)$$

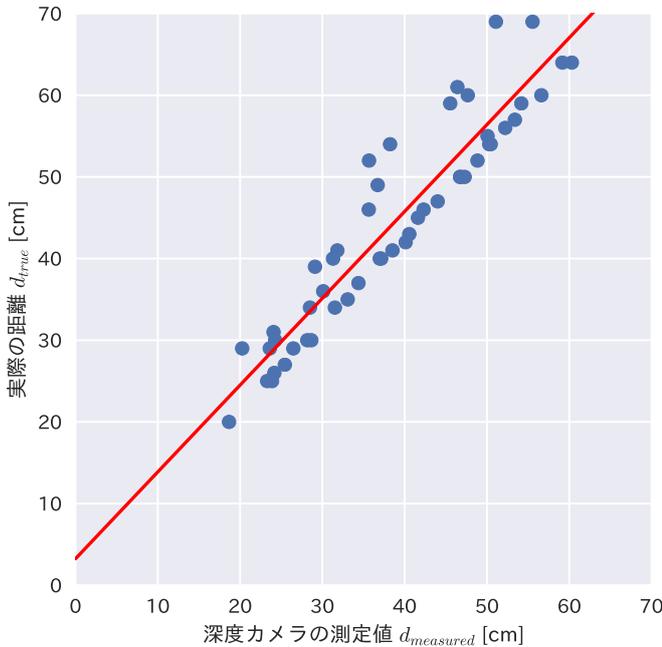


図 14: 深度カメラの精度

以降、システムでは深度カメラの出力を式 9 を用いて較正しているものとする。

5.2 食品体積からのカロリー量の回帰実験

本システムのカロリー量推定は、深度付き画像を用いて推定した食品の体積からの回帰で求める。この実験ではカロリーがわかっている食品を用いて回帰式を作成した。実験に用いる食品のカテゴリは CalorieCam [6] および AR DeepCalorieCam V2 [8] においてカロリー量の推定に対応しているものの中で、実験時に実物が用意可能であった 3 カテゴリを選択した。これらのカテゴリの食品をそれぞれ S・M・L の 3 サイズ用意したものを各サイズごとに様々な距離から 10 回ずつ撮影し、推定された体積と基地のカロリーから回帰式を作成した。なお、実験時の基準面は図 15 に写っているものと同様のテクスチャのある面であった。選択したカテゴリと例として M サイズの画像の一覧を図 15 に、各食品カテゴリのサイズごとのカロリーを表 1 に示す。

表 1: 各食品カテゴリのカロリー量

カテゴリ	カロリー [kcal]		
	S	M	L
酢豚	220	368	463
唐揚げ	199	377	583
コロッケ	84	168	252



(a) 酢豚 (b) 唐揚げ (c) コロッケ

図 15: 回帰式の作成に使用した食品カテゴリ

推定された体積とカロリー量のグラフを図 16 に示す。またカテゴリごとに推定された体積を説明変数 x_{vol} 、カロリー量を目的変数 y_{cal} として線形回帰を行って得た回帰式およびその決定係数を表 2 に示す。図 16 中の直線はこの回帰式で表される直線である。

表 2: 食品カテゴリごとの回帰式

食品カテゴリ	回帰式	決定係数 R^2
酢豚	$y_{cal} = 1.50x_{vol} + 33.1$	0.8959
唐揚げ	$y_{cal} = 1.91x_{vol} + 53.4$	0.9769
コロッケ	$y_{cal} = 1.43x_{vol} + 7.30$	0.8915

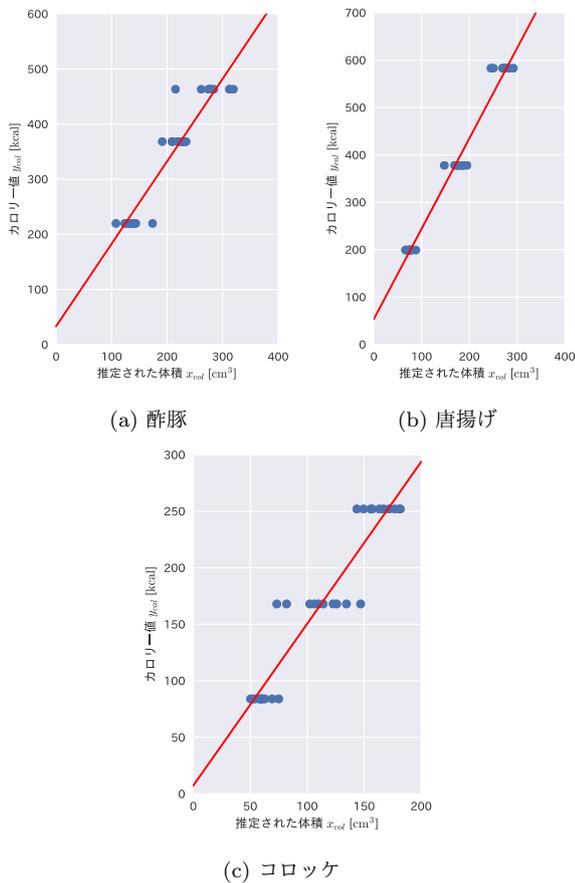


図 16: 各食品カテゴリーの推定体積とカロリー量のグラフ

5.3 カロリー量の推定精度に関する実験

この実験では最終的な本システムの評価のために、既存手法である CalorieCam [6] および AR DeepCalorieCam V2 [8] と実物の食品のカロリー量推定の精度をユーザーテストを行って比較した。実験には 5.2 章で挙げたカテゴリーの食品について、5.2 章の実験で利用したものとは別の店舗で同じカテゴリーかつカロリー量の記載のあるものを利用した。画像およびカロリー量を図 17 と表 3 に示す。



図 17: システムの評価に用いた食品

表 3: 各食品カテゴリーのカロリー量

カテゴリー	カロリー [kcal]
酢豚	500
唐揚げ	655
コロッケ	246

実験はこの 3 カテゴリーの食品を実際の食事の場面を想定して食器に盛り付けたものを対象とし、各カテゴリーそれぞれにつ

いて基準面のテクスチャの有無を切り替えた計 6 パターンにて実施した。基準面のテクスチャの有無について、例を図 18 に示す。各パターンについて、食事のカロリー量について特別な知識がない被験者 3 名がそれぞれのシステムで 5 回ずつカロリー量を測定し、計 270 回の測定を行っている。



(a) テクスチャあり (b) テクスチャなし

図 18: 基準面のテクスチャの有無

まず各食品カテゴリー・システムごとのカロリー量の推定値の誤差の平均と標準偏差を表 4 に示す。

表 4: カロリー量の推定結果の誤差の平均 ± 標準偏差 (単位: kcal)

カテゴリー	CalorieCam	AR CalorieCam V2	DepthCalorieCam
酢豚	364±552	-112±163	2±52
唐揚げ	-123±171	343±51	-5±64
コロッケ	-48±16	-104±12	-35±22

さらに表 4 に被験者および基準面のテクスチャの有無ごとの分類を追加したものを表 5 と表 6 に示す。

表 5: 被験者別のカロリー量の推定結果の誤差の平均 ± 標準偏差 (単位: kcal)

カテゴリー	被験者	CalorieCam	AR CalorieCam V2	DepthCalorieCam
酢豚	A	315±155	-58±269	33±67
	B	79±187	-118±68	-18±37
	C	697±841	-160±52	-10±35
唐揚げ	A	-149±77	-308±16	6±73
	B	-156±164	-390±63	-19±32
	C	-62±235	-331±14	-2±80
コロッケ	A	-46±11	-100±5	-34±18
	B	-56±8	-112±13	-31±24
	C	-43±24	-99±13	-40±26

表 6: 基準面のテクスチャの有無の別カロリー量の推定結果の誤差の平均 ± 標準偏差 (単位: kcal)

カテゴリー	テクスチャ	CalorieCam	AR CalorieCam V2	DepthCalorieCam
酢豚	あり	571±713	-130±224	1±28
	なし	156±177	-93±65	2±69
唐揚げ	あり	15±129	-364±64	17±68
	なし	-260±58	-321±19	-27±52
コロッケ	あり	-46±21	-111±10	-28±19
	なし	-50±11	-96±9	-43±24

これらの結果について 6 章で評価・考察を行う。

6 考 察

実験 5.3 の結果より、まず本システムの全体的な評価について、表 1 から今回実験に用いた 3 カテゴリーの食品については既存手法を上回る精度でのカロリー量の推定を可能にしたことがわかる。相対誤差について考えると酢豚においては CalorieCam [6] 比で 42.1%→0.4%で約 $\frac{1}{100}$ 、唐揚げにおいては AR DeepCalorieCam V2 [8] 比で 34.4%→0.8%で約 $\frac{1}{43}$ という高精度を実現した。これは既存手法が表面積を元にカロリー量を 2 次の回帰式を用いて回帰しており、推定された表面積の誤差の影響を受けやすいのに対し、本システムでは推定した体積を元に線形の回帰式を用いて回帰していることで、カロリー量が体積の誤差の影響を受けにくく、また食品のカロリー値はその体積に比例するという原則に近い回帰結果を得られることが理由だと考えられる。

また表 5、6 より、ユーザーごとの撮影方法の細かな違いや撮影時の環境の影響を受けにくく、安定して高い精度を実現している事がわかる。そして、既存手法についても、表 6 から CalorieCam が基準面にテクスチャがある場合に標準偏差が大きくなっている点から、テクスチャの影響を受けやすいことや、実験 5.3 の結果全体で AR DeepCalorieCam V2 のコロケの標準偏差が他の食品カテゴリーに比べ小さい点から、コロケのような形が単純なものほどユーザーが AR 空間で設置するアンカーの位置のばらつきが少なくなることなどもわかった。

7 まとめと今後の課題

iPhone の特定のモデルに搭載されている深度カメラから取得できる深度付き画像と、食事カテゴリー認識や食品領域分割タスクをこなす深層学習モデルを用いることで、携帯端末のみで食品のカロリー量を推定するシステム “DepthCalorieCam” を開発した。開発したシステムでは、深度の適切な較正に加え、食品のカロリー量を表面積からの回帰ではなく、体積からの回帰で求めることで既存手法に比べ推定カロリー量の相対誤差を最小で $\frac{1}{100}$ に抑えることが可能となった。また最良の精度だけでなく、食事が置かれている基準面やユーザーの撮影方法による誤差の影響を受けづらく、カロリー量推定の安定性についても既存手法から大きく向上させることができた。

一方で、現在のシステムでは対応している食品カテゴリーが 3 つのみで、その数は既存手法に比べると非常に少ない。本システムの構成上、新たなカテゴリーに対応するためには、カロリー量がわかっている実物の食品が必要であり、そのコストは既存手法に比べると高くなってしまっている。しかし、今回高い精度でカロリー量が推定できたことで、深度値の較正および体積推定の精度については一定の水準以上であるとみなせるため、実物の食品の撮影のみで対応カテゴリーを増やすことが可能であると考えられる。また、今回は既存手法との比較評価のために実験に利用する食品カテゴリーが制限されていたが、対応カテゴリーを増やすという点のみを考えるとその制限もなくなるため、より容易に追加できるだろう。このような背景もあり、今後についてはまずシステムの実用性を向上させる事を考え、対応カテゴリーを増加させたいと考えている。

- [1] T. Miyazaki, Gamhewage C. De Silva, and K. Aizawa. Image-based Calorie Content Estimation for Dietary Assessment. In *Proc. of IEEE International Symposium on Multimedia (ISM)*, pp. 363–368, 2011.
- [2] 會下拓実, 下田和, 柳井啓司. CNN 特徴量学習に基づく画像検索による食事画像カロリー量推定. 電子情報通信学会論文誌 D, Vol. J101-D, No. 8, pp. 1099–1109, 2018.
- [3] T. Ege and K. Yanai. Image-Based Food Calorie Estimation Using Recipe Information. *IEICE Transactions on Information and Systems*, Vol. E101-D, No. 5, pp. 1333–1341, 2018.
- [4] M. Chen, Y. Yang, C. Ho, S. Wang, S. Liu, E. Chang, C. Yeh, and M. Ouhyoung. Automatic Chinese Food Identification and Quantity Estimation. In *Proc. of SIGGRAPH Asia Technical Briefs*, p. 29, 2012.
- [5] A. Myers, N. Johnston, V. Rathod, A. Korattikara, A. Gorban, N. Silberman, S. Guadarrama, G. Papandreou, J. Huang, and K. P. Murphy. Im2calories: towards an automated mobile vision food diary. In *Proc. of IEEE International Conference on Computer Vision (ICCV)*, pp. 1233–1241, 2015.
- [6] 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 (MADiMa)*, 2016.
- [7] C. Rother, V. Kolmogorov, and A. Blake. Grabcut: Interactive foreground extraction using iterated graph cuts. In *ACM transactions on graphics (TOG)*, Vol. 23, pp. 309–314, 2004.
- [8] R. Tanno, T. Ege, and K. Yanai. Ar deepcaloriecam v2: Food calorie estimation with cnn and ar-based actual size estimation. In *Proc. of ACM Symposium on Virtual Reality Software and Technology (VRST)*, 2018.
- [9] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In *Proc. of IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2818–2826, 2016.
- [10] O Ronneberger, P Fischer, and T Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *Proc. of Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pp. 234–241, 2015.
- [11] Y. Matsuda, H. Hoashi, and K. Yanai. Recognition of multiple-food images by detecting candidate regions. In *Proc. of IEEE International Conference on Multimedia and Expo (ICME)*, 2012.
- [12] F. Chollet. Xception: Deep Learning with Depthwise Separable Convolutions. In *Proc. of IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1800–1807, 2017.
- [13] L. Bossard, M. Guillaumin, and L. Van Gool. Food-101 – Mining Discriminative Components with Random Forests. In *Proc. of European Conference on Computer Vision (ECCV)*, pp. 446–461, 2014.