

iPhone 深度センサを用いた食事カロリー量推定アプリ

藤田 晴斗^{1,a)} 柳井 啓司^{1,b)}

概要: 近年、食事のカロリー管理が健康維持やダイエットにおいて重要な役割を果たしている。特に、食事内容やカロリーの記録に対する関心が高まっており、それをサポートするさまざまなアプリが提供されている。しかし、利便性を追求するあまり、精度が低下しているものや、ユーザビリティに課題があるものも少なくない。例えば、ユーザーが一品ずつ食品の種類や量を手動で入力しなければならず、事前に食品に関する知識が必要なアプリや、食品の画像を撮影すると自動で食品の種類を認識するが、カロリーに関しては量に関係なく固定されているものなどが存在する。

そこで、本研究では、iPhone に搭載されている LiDAR センサを用いて、食事のカロリー量をリアルタイムで推定するシステム「LiDARCaloriCam」を提案する。本システムでは、食事の画像を撮影し、深度センサによって食品の3次元形状を取得する。その3次元形状をもとに、食品の体積を推定し、その体積を元にカロリーを算出する。既存研究から食事カテゴリの増加やカロリー推定の精度向上を図り、より信頼性の高いカロリー推定を実現した。

1. はじめに

従来の深層学習を用いた手法では、食材の種類や量を推定し、カロリーを算出するアプローチが取られてきた。しかし、正確なカロリー推定には食品の体積を考慮することが不可欠であり、これが課題として残っていた。特に、平面画像だけでは食品の高さや奥行きといった3次元的情報を取得することが難しく、体積の推定精度に限界があった。

本研究では、この課題を解決するために LiDAR (Light Detection and Ranging) 技術を応用し、食品の3次元形状を高精度で捉えることを目指している。LiDAR は、レーザー光を用いて物体までの距離を計測する技術であり、これにより食品の立体的な構造を詳細に解析することが可能となる。本手法では、得られた3次元データに基づいて食品の体積を推定し、その体積を元にカロリーを算出することで、従来の手法に比べて高精度なカロリー推定を実現することを目指す。

2. 関連研究

食事画像を用いたカロリー推定は、健康管理や栄養指導で重要な研究領域である。この分野では、食事のカテゴリ分類と体積推定を組み合わせた手法が一般的で、多様なアプローチが提案されている。

2.1 カロリー推定における基礎研究

Akpa ら [1] や丹野ら [16] の研究は、スマートフォンを利用したカロリー推定手法の一例であり、食品の量とカロリーをより簡単かつ精度高く推定するために、身近な物体を基準物体として利用する手法を提案している。具体的には、箸や米粒など、ユーザーが日常的に使用する物体を基準として画像に収め、食品のサイズや体積を相対的に計測する方法である。この手法は、特別な機器や道具を必要としない点で利便性が高いが、ユーザーが撮影時に基準物体を意識して配置する必要がある、また食品の複雑な形状に対する対応が難しいという課題も存在する。

一方、Fang ら [7] は、基準物体に頼らない方法として、食品の幾何学的形状を利用して体積を推定するアプローチを提案している。彼らの研究では、単一の視点から得られる画像データをもとに、食品の幾何学モデルを生成し、それに基づいて体積を推定する。この手法は、画像に含まれる食品の形状を数学的なモデルとして捉えるため、基準物体を用意する必要がなく、より手軽に体積推定が可能となっている。

また、Quin ら [17] の研究では、「Nutrition5k」と呼ばれるデータセットを用いて、食品の画像からカロリーを推定する手法を提案している。このデータセットには、5,000種類以上の食品画像が含まれており、食品の種類とカロリー量、栄養素がラベル付けされている。しかし、データセットの収集対象は西洋料理を中心となっていることや、単一の食品ではなく複数の食品が混在する場合が多く、1つの

¹ 電気通信大学
The University of Electro-Communications
a) fujita-h@mm.inf.uec.ac.jp
b) yanai@uec.ac.jp

食品カテゴリに対してカロリー値を一律に適用するという問題がある。

岡元ら [13] が開発した「CalorieCam」は、クレジットカードなどの既知サイズの基準物体を利用して食事の実面積を算出し、カロリーを推定するシステムである。このアプローチは、Android スマートフォンを用いて手軽に食事の写真撮影し、基準物体と比較して食品の面積を算出することで、その面積を基にカロリーを推定するというものである。

この手法の利点は、基準物体を使用することで簡単に実際の面積を計測できる点であり、ユーザーは日常的なスマートフォンを使用して簡単に操作できる。しかし、ユーザーが常に基準物体を用意しなければならないことや、面積のみの計測では立体的な食品の体積推定が難しいという問題もある。特に、食品の高さや奥行きが大きく影響する場合、面積の計測だけでは正確なカロリー推定が困難となる。

2.2 基準物体を必要としないアプリケーション開発

丹野ら [16] は基準物体を必要としない、AR 技術を組み合わせた「AR DeepCalorieCam V2」を開発している。

このシステムは、スマートフォンカメラと AR (拡張現実) 技術を用いて食品のサイズを推定し、カロリーを計算する。AR 技術を活用することで、よりリアルタイムでのサイズ推定が可能となっており、手動での計測を省くことができる。実行例を図 1 に示す。端末を動かしながら AR 空間内でアンカー (図 1 中の緑色のポイント) を設置しなければならないが、撮影における手間が多いことが問題である。

安藤 [2] が開発した「DepthCalorieCam」は、iPhone のデュアルカメラを利用した深度付き画像と深層学習を用いて、食事のカロリーを推定するシステムである。深度値の算出は、iPhone 等の望遠レンズと広角レンズの 2 つのカメラの視差 (Disparity) により行う。DepthCalorieCam におけるカロリー量推定の流れは下記のようになっている。

- (1) 食品を真上から撮影し、iPhone のデュアルカメラを用いて深度画像を取得
- (2) 領域検出モデル U-Net[15] を用いて食事領域を抽出
- (3) XY 方向の実寸からその体積を推定
- (4) Xception[5] を用いて食品カテゴリごとの体積からカロリー量を回帰式でカロリー量を推定

AR DeepCalorieCam V2 と同様に、基準物体を必要としない手法であるが、AR 技術を用いずに深度画像を取得することで、より簡単に食事のカロリーを推定することが可能となっている。図 2 は、実際の DepthCalorieCam の画面例である。また、表面積ではなく、深度画像を用いた体積を測定することからより誤差の小さい測定が可能になっている。しかし、食事カテゴリが 3 種類に制限されているため、実用的な用途に欠けることや撮影する位置が真上から

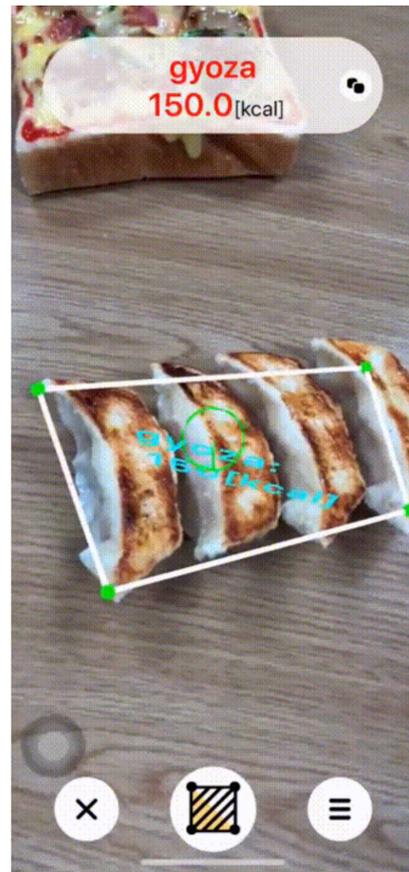


図 1 AR DeepCalorieCam V2 の食品領域指定の様子 ([16] から引用)

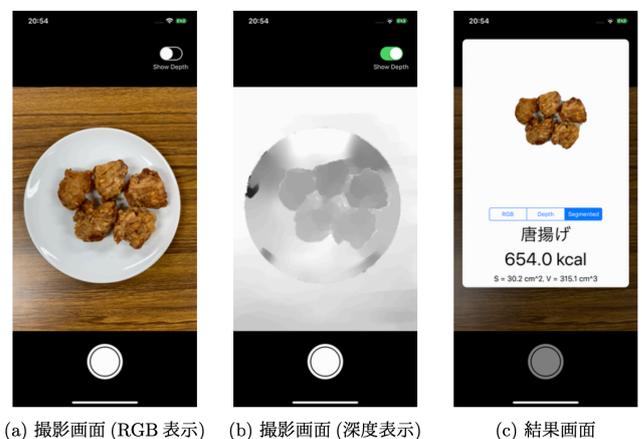


図 2 DepthCalorieCam の画面例 ([2] から引用)

に限定される点など使用が制限されてしまっている点の問題となっている。

2.3 本研究の位置付けと LiDAR 技術の活用

これらの既存の手法は、それぞれに特徴があり、一定の成果を上げているが、基準物体の準備や食品の形状の複雑さに起因する精度の限界が指摘されている。また、単一画像や 2 次元の情報に依存する手法が多いため、食品の正確な体積を推定するためには、さらなる手法を検討する必

要がある。

本研究では、これらの課題を克服するために LiDAR 技術を用いた新たなアプローチを提案する。

LiDAR は、レーザー光を使用して物体までの距離をより正確に計測し、3 次元的な形状データを取得することができる。この技術を活用することで、食品の立体的な体積を正確に測定し、基準物体を必要としない体積推定が可能となる。

実際に iPhone の LiDAR センサを利用した研究「NutritionVerse」[4] では、iPhone 13 Pro Max の LiDAR センサを用いて食品の形状を取得している。251 の料理の 889 枚の画像データセット NutritionVerse-Real を作成し、間接セグメンテーションおよび直接予測ネットワークを含む深層学習モデルのトレーニングおよび評価を行っている。

本研究では、LiDAR 技術を活用してモバイル端末上での食事のカロリー推定を実現するためのシステムを提案し、その有効性を検証する。

3. システム全体構成

本システムは、iOS 端末上のクライアントアプリケーションと、サーバーサイドの体積計算処理で構成される。クライアントアプリケーションでは、ARKit により深度推定を行い、Core ML を用いて食品領域のセグメンテーションと食品カテゴリの分類を実行する。これらの情報を基に三次元点群を生成し、サーバーサイドに送信する。サーバー側では受信した点群に対して複数の手法で体積を算出し、推定結果をクライアントに返却する。

3.1 システムの処理フロー

本システムの処理フローを図 3 に示す。スマートフォンから得られる RGB 画像と深度情報を用いて食品領域を抽出し、三次元点群を生成する。その後、サーバーサイドで体積を推定し、食品カテゴリごとの情報と照合してカロリー量を算出する流れとなる。

- (1) **深度付き画像撮影**: ARKit を用いてスマートフォンで食品を撮影し、RGB 画像と深度マップを取得する
- (2) **食品領域抽出**: セグメンテーションモデル (DeepLabV3[3]) により食品部分を切り出し、背景を除去する
- (3) **食品分類予測**: 分類モデル (Image Print Feature V1) を用いて食品カテゴリを推定する
- (4) **点群生成**: カメラ内部パラメータと LiDAR による深度マップを用い、三次元座標を求める。同時に RGB 値や食品クラスラベルを付与して点群を構築する
- (5) **点群情報送信**: 生成した点群をサーバーに送信し、解析要求を行う
- (6) **点群前処理**: 外れ値除去や品質チェックを行い、体積計算に適した点群データを生成する

- (7) **体積計算**: サーバー側で複数の手法 (Convex Hull 法, Delaunay 三角分割法, メッシュベース法) を用いて体積を算出する
- (8) **推奨体積評価**: 信頼度スコアを計算し、最適な体積を選択する
- (9) **体積情報送信**: 体積計算結果をクライアントに送信し、カロリー推定に利用する
- (10) **カロリー量推定**: 得られた体積と食品カテゴリの情報を基にカロリー量を算出し表示する

3.2 食品領域の抽出

3.2.1 セグメンテーション

食品領域を正確に切り出すため、本研究では DeepLabV3 を Core ML 形式で運用する方針を取った。UECFOOD-Pix[12] の食事画像データセットで fine-tuning したモデルを iOS 端末に組み込み、リアルタイム推論を行う。背景をマスクし、食品に特化した三次元点群を得るための基礎情報を生成する。実際のセグメンテーション結果の例を図 4 に示す。(a) の入力画像を与えると、(b) のような食事領域に対するセグメンテーションマスクが出力される。

また、DeepLabV3 は、のトランスフォーマーベースの手法 (例えば、SegFormer) などと比べると、セマンティックセグメンテーションタスクにおいてやや古いモデルとされる。しかし、実際の iOS アプリケーションへの組み込みを考慮した場合、以下の理由から本研究では DeepLabV3 を採用した。

- (1) **iOS 端末でのリアルタイム推論が可能**: iOS デバイス上で Core ML を用いて推論を行う際、モデルサイズや演算量が増えすぎると推論速度が低下し、ユーザーエクスペリエンスを損ねる恐れがある。DeepLabV3 は推論速度と精度のバランスが良好であり、実機上での計測でもリアルタイムセグメンテーションが十分実用的な速度で実行できることが確認された。
- (2) **実装の容易さ**: DeepLabV3 は GitHub 上などに Core ML 変換の例や導入事例が多く、既存のコードを参考に効率的に実装できる。DeepLabV3+ に比べてネットワークの構造が単純であるため、移植時の互換性問題やデバッグコストを低減できるメリットもある。

これらの理由から、本研究では DeepLabV3 をベースにしたモデルを採用し、UECFOODPix データセットで学習させたモデルを用いて iOS 端末上のリアルタイム推論を実装した。

3.2.2 食品分類

セグメンテーションで抽出された画像を、Apple が提供する iOS の機械学習ライブラリである Create ML[8] の Image Classification モデルに入力し、食品カテゴリを判定する。Image Classification のモデルは、Image Feature Print V1 (IFP) と呼ばれる特徴量を抽出するための量

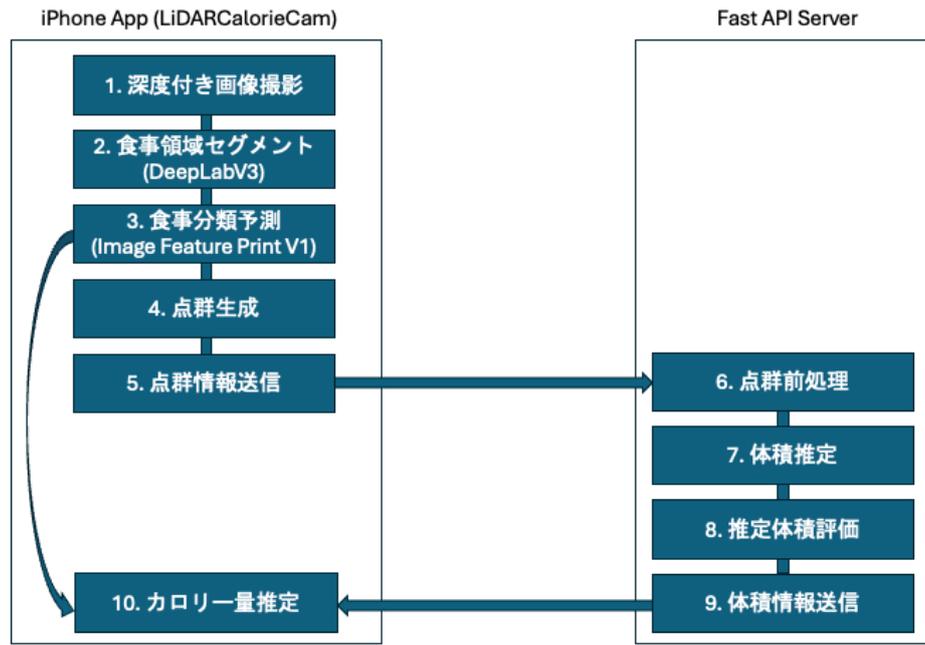


図 3 システムの処理フロー

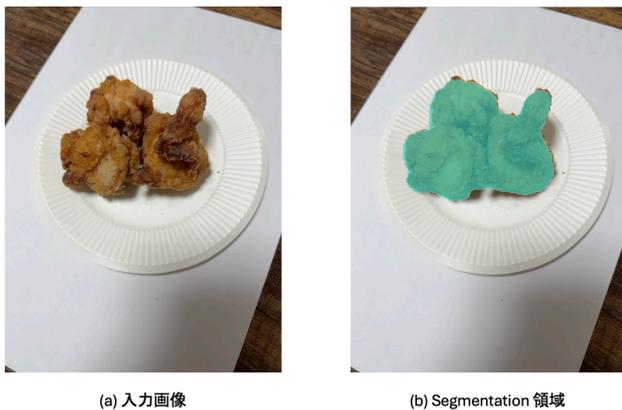


図 4 セグメンテーション結果の例

み込みニューラルネットワークを使用している。IFPv1では、 299×299 ピクセルの画像を入力として受け取り、 2048 次元の特徴量を出力する。UEC Food-100[9] のデータセットを用いて学習したモデルを適用し、後に各カテゴリに対応するカロリー密度や体積-重量変換用の回帰式を後段で参照できるようにする。

3.3 深度推定と点群生成

3.3.1 深度推定 (ARKit の sceneDepth)

本研究では、iOS 端末上の拡張現実 (AR) 開発フレームワークである ARKit を用いて食品の三次元情報を取得する。ARKit は、カメラ映像、IMU (加速度センサー、ジャイロスコープ)、LiDAR センサーなどの複数のセンサー情報を統合し、デバイスの姿勢 (位置・向き) 推定や周囲環境のマッピングを行うフレームワークである。ARKit を用いることで、座標変換や空間認識のための複雑な低レベル

処理を意識することなく、三次元的な位置や姿勢を扱う機能を比較的容易に実装できることが特徴である。

ARKit は、以下のような手順で空間認識を行う。

- **センサーデータの統合**: デバイス内蔵のカメラ映像 (RGB 画像)、IMU (加速度センサー・ジャイロスコープ)、および (対応デバイスの場合) LiDAR センサーからのデプス情報を取得し、カメラの姿勢推定と空間マッピングをリアルタイムで行う。
- **画像特徴点のトラッキング**: カメラ映像中に見られる特徴点 (コーナーやエッジなど) を抽出し、フレーム間で追跡することでデバイスの移動量や回転量を推定する (VIO: Visual Inertial Odometry)。

本研究では、LiDAR センサからの深度情報をリアルタイムで取得できるライブラリを用い、各フレームに対して、カメラの RGB 画像と対応する深度マップを同時に受け取り、奥行き方向の情報を利用している。

ARKit が提供する `ARFrame` には、カメラの姿勢だけでなく、カメラ内部パラメータ (K 行列) も含まれている。カメラ内部パラメータを用いて、画像座標系から実世界座標系への変換を行うことが可能である。図 5 のようにピンホールカメラモデルを仮定し、カメラの焦点距離や画像中心座標などのパラメータを用いて、画像座標 (u, v) から実世界座標 (X, Y, Z) への変換を行う。

たとえば、焦点距離 f_x, f_y や画像中心座標 (c_x, c_y) といった情報は以下のような行列で表される。

$$K = \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix}$$

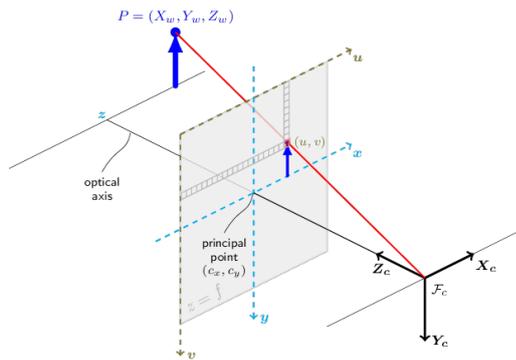


図 5 ピンホールカメラモデル (OpenCV[14] 公式ドキュメントより引用)

各ピクセル座標 (u, v) と深度値 d が与えられたとき、実世界座標 (X, Y, Z) を次式で求める。

$$X = \frac{(u - c_x) \cdot d}{f_x}, \quad Y = \frac{(v - c_y) \cdot d}{f_y}, \quad Z = d$$

ここで、 d は sceneDepth から得られる深度マップ上の値 (奥行き) であり、一般にメートル単位となる。得られた (X, Y, Z) は、カメラ座標系 (左手系または右手系) を基準とする実空間での三次元位置を意味する。

3.4 点群生成

セグメンテーションマスクを用いて食品画素のみを抽出し、RGB 値と座標情報を組み合わせる。深度マップ (256 × 192) と RGB 画像 (1920 × 1440) と解像度が異なるためスケーリングを考慮した点群を作成する。各点には (X, Y, Z) と (R, G, B) の情報が付与され、サーバーへ送信される。実際にサーバー側で受信されたコロッケの点群データの例を図 6 に示す。

3.5 体積計算

サーバー側に受信した点群に対して点群中の密度の低い領域や孤立した点を外れ値として検出・除去できるよう、DBSCAN[6] を用いた外れ値の除去を行う。なお外れ値除去により半数以上の点が失われた場合は元の点群にロールバックするように設計している。

本システムでは、3つの手法を組み合わせることで体積を推定する。点群から体積を計算する代表的な手法として、Convex Hull 法、Delaunay 三角分割法、メッシュベース法を採用している。

各手法から得られた体積値の平均 μ と標準偏差 σ を計算し、変動係数 (CV: Coefficient of Variation) $CV = \frac{\sigma}{\mu}$ を求める。信頼度を $confidence = e^{-CV}$ で定義し、confidence が高いほど精度が高いとみなす。

推奨体積の選択は信頼度が 0.8 以上の場合はメッシュベース法の結果を採用し、それ以外については Convex Hull 法の結果を採用している。

表 1 食品ごとの体積推定誤差

食品	MAE (cm ³)	MAPE (%)
唐揚げ	32.5	64.5
コロッケ	58.0	31.0
ホットドッグ	28.5	8.8
焼き鳥	37.4	39.5
トースト	61.3	13.9

3.6 カロリー推定

3.6.1 体積から重量への変換

食品カテゴリごとに事前に測定した体積-重量データを用いて回帰式を作成する。回帰式を $W = aV + b$ とし、 V を体積、 W を重量とする。パラメータ a と b はカテゴリごとに異なる値を設定し、推定された体積を入力として重量を推定する。

3.6.2 重量からカロリー量への変換

カテゴリごとのカロリー密度 [kcal/g] を食品成分表や既存データベースから取得し、 $\hat{C} = D \times \hat{W}$ よりカロリー量 \hat{C} を求める。ここで D はカテゴリごとに定義されるカロリー密度、 \hat{W} は推定重量である。

4. 実験

本システムの性能を総合的に評価するための実験を行う。また、実際のアプリケーションの動作例を図 7 と図 8 に示す。

4.1 体積推定の精度評価

実験方法

複数の食品 (唐揚げ、コロッケ、ホットドッグ、焼き鳥、トースト、ピザトーストのサンプル) を 30cm 程度の距離から斜め撮影し、取得した三次元点群に対して体積を推定する。推定体積と実体積の差を評価するため、実体積の測定には図 9 のように計量カップや水置換法などを活用し、吸水性がある食品にはラップを巻くなどの対応を行った。

実験結果

表 1 に、食品ごとの推定体積と実体積の誤差を MAE と MAPE で示す。なお、今回はすべての測定データが信頼度を 0.8 を超えていたためメッシュベースの体積値を使用している。

撮影角度や食品形状に起因するバラツキはあるが、大きな外れ値は少なく、概ね許容範囲内に収まる結果が得られた。一部の食品では誤差が大きくなる傾向が見られたが、これは撮影角度や形状の特性によるものと考えられる。5.1 節で詳細に議論する。

なお、図 10 に、唐揚げ (推定体積値: 96cm³) の Convex Hull 法による体積推定結果例を示す。

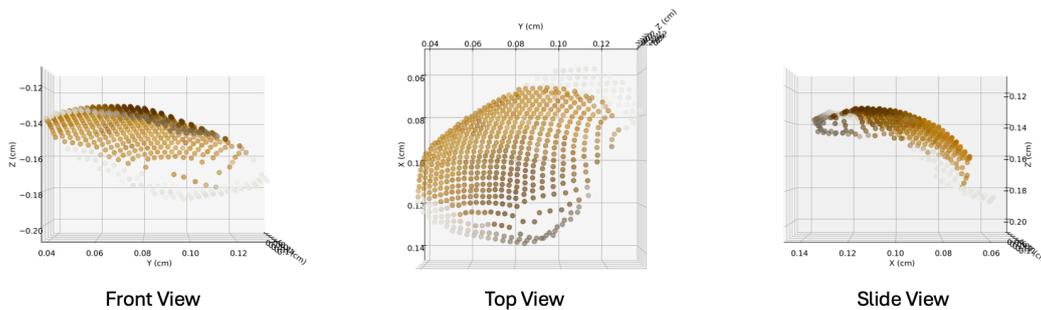


図 6 コロケの点群データの例

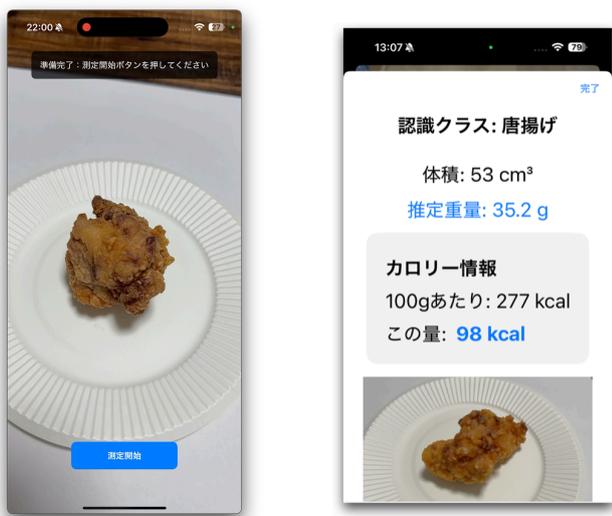


図 7 撮影画面

図 8 カロリー推定結果



図 9 唐揚げの体積測定例

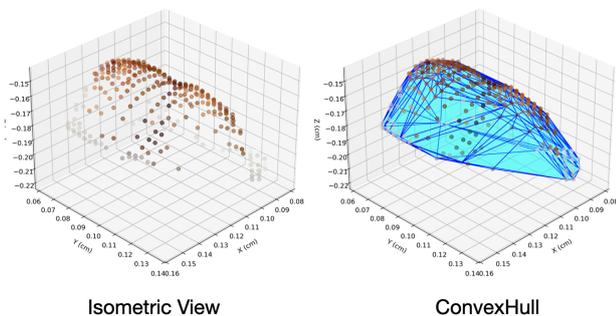


図 10 唐揚げの体積推定結果例

4.2 重量推定の回帰分析

実験方法

推定体積 (V) と実重量 (W) の関係を単回帰分析によっ

表 2 食品ごとの推定体積 (V) から重量 (W) を求める回帰式

食品	回帰式 ($W = aV + b$)	R^2
唐揚げ	$W = 0.45V + 11.4$	0.95
コロケ	$W = 0.33V + 38.0$	0.87
ホットドッグ	$W = 0.17V + 23.81$	0.65
焼き鳥	$W = 0.36V + 26.0$	0.92
トースト	$W = 0.15V + 22.6$	0.97

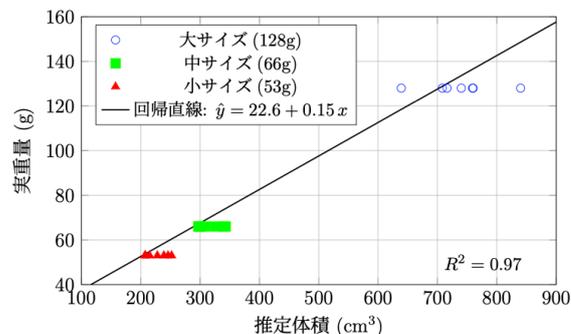


図 11 トーストにおける推定体積と実重量の関係

てモデル化する。食品カテゴリごとに複数サイズの食品を撮影し、回帰式 $W = aV + b$ を求める。

実験結果

表 2 に食品サンプルを除く 5 カテゴリの回帰式と決定係数 R^2 を示す。

図 11 に、トーストに対する推定体積と実重量の関係を示す。大きさの異なるトーストを複数回撮影したデータを用い、回帰直線を重ねてプロットしている。

4.3 カロリー推定の精度評価

体積から重量を推定し、さらに食品成分表や既存のカロリー情報に基づいてカロリー量を計算する。表 3 と表 4 に、食品ごとの推定カロリーについて MAE と MAPE を示す。カロリー量は文部科学省が公開している日本食品標準成分表 [18] と、食品栄養情報が掲載された Web サイトであるカロリー Slism^{*1} を参考に行っている。既存研究 DepthCalorieCam と比較するために、唐揚げとコロケの結果を合わせて示す。

*1 <https://calorie.slism.jp/>

表 3 カロリー量の推定結果の誤差 MAE (kcal)

食品カテゴリ	DepthCalorieCam	LiDARCalorieCam
唐揚げ	101.29	15.88
コロッケ	43.09	12.83
焼き鳥	—	25.68
ホットドッグ	—	12.03
トースト	—	23.95

表 4 カロリー量の推定結果の誤差 MAPE (%)

食品カテゴリ	DepthCalorieCam	LiDARCalorieCam
唐揚げ	52.84	7.87
コロッケ	20.11	6.10
焼き鳥	—	11.50
ホットドッグ	—	5.59
トースト	—	8.64

4.4 まとめ

体積推定・重量推定・カロリー推定の一連の処理を総合的に検証した。結果として、既存手法と比較して推定誤差を低減でき、高精度なカロリー推定が可能になることを示した。

5. 考察

本章では、開発したシステムの性能を評価するために実施した実験結果をもとに、体積推定アルゴリズム、カロリー推定手法に関する考察を行う。さらに、今後の研究課題と展望についても述べる。

5.1 体積推定の精度に関する考察

体積推定の実験では、ピザトーストのように高さや形状が複雑なサンプルで誤差がやや大きいものの、多くの食品で実体積に比較的近い推定値が得られた。推定体積が実体積を上回る傾向が見られた点については、メッシュ生成や Convex Hull 法における凸包近似が一因と考えられる。とくに表面に凹凸が多い食品では、凹部が潰されてしまい、過大評価が生じる場合がある。また、カメラ死角となる裏面や隙間付近は点群が薄くなりがちで、計測精度が低下することが確認された。

複数の食品をまとめて計測する際に、過大評価が生じるケースを確認した。例として、図 12 では、2つのコロッケを並べた状態で計測した結果を示しているが、実体積と推定体積の差が大きくなっている。これは、全体的に点群の密度が低く、食品間の重なり部分の空間について点群の取得が不十分であったり、重なり部分の凸包近似が正確でなかったりすることが原因であると考えられる。

撮影条件の検討（カメラの距離・角度）や複数視点からの撮影を導入することで、この誤差をある程度抑制できる見込みがある。実験でも、同一食品を複数回計測した結果が誤差の範囲内で安定していることが示されたため、再撮影やマルチアングル撮影を行えば推定結果の信頼性を高

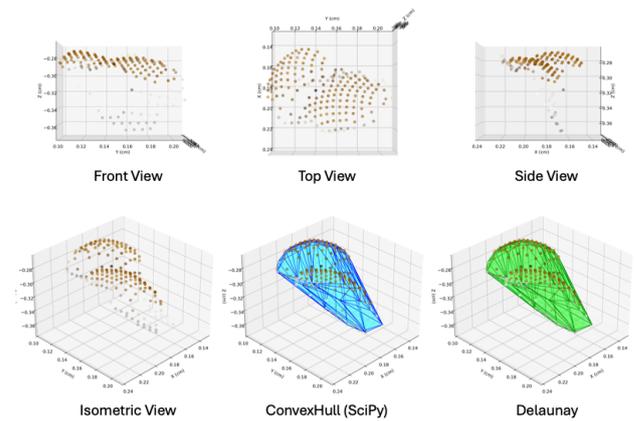


図 12 推定誤差の大きい例

められる。また、点群の外れ値除去アルゴリズムを高度化することや、より細かいメッシュを取り入れた凹部の形状再現を改善することで、さらに精度向上を図れると考えられる。

さらに、モバイル内で一連の処理を完結できるよう、モバイル上での体積推定処理の実現が求められる。具体的には現在、モバイル内単体で点群から高精度な体積推定の実装に課題が残されている。例えば、点群をボクセル化を行うことでデータ量を削減しつつより単純な計算で体積を求めることができるのではないかと考える。

5.2 カロリー推定の精度に関する考察

カロリー推定においては、推定体積とカテゴリ別の回帰式を用いて重量を推定し、食品成分表に基づいてカロリー量を算出する仕組みを実装した。唐揚げやコロッケといった比較的形状や密度が安定している食品では DepthCalorieCam よりも絶対平均誤差が大幅に低下した。一方で、焼き鳥やホットドッグのように形状が細長く、撮影時の角度の影響を受けやすい食品では、誤差のばらつきがやや大きい傾向が見られた。

さらに、同一カテゴリであっても調理法や具材が変われば実際のカロリー量に大きな差が生じる点が、本システムにおける回帰式の限界として挙げられる。あくまでカテゴリごとの平均値に依存した推定であるため、ソースや追加トッピングが施された食品を正しく補正できない場合がある。未知の食品カテゴリに対応できない点も含め、今後はより汎用的なアプローチが必要となる。また、本研究では1つの食品カテゴリに対してのみ体積推定を行っているが、食事全体のカロリー推定を行うためには、複数の食品の同時計測や、食事全体の体積推定が可能な機能の追加が必要である。

5.3 今後の研究課題と展望

本研究で開発したシステムは、スマートフォン上での LiDAR 計測と深層学習モデルを用いることで、高精度な

カロリー推定を可能にする基盤を提供している。しかし、実験結果から明らかになったように、撮影条件の変動や食品の多様性への対応など、まだ解決すべき課題が残されている。

まず、セグメンテーションでは微細な領域分割や暗所での認識精度を向上させる必要がある。今後は、マルチスケールでの学習や大規模データ拡張を行い、実際の食事環境を想定した耐性を付与することが望ましい。さらに、複数視点からの撮影や点群合成技術を導入することで、凹凸の激しい食品の三次元再構成精度を高め、体積推定の正確性を向上させることができると考えられる。

カロリー推定に関しては、カテゴリ別回帰式の枠を超えた多変量推定モデルや、深層学習による直接的なカロリー回帰モデルへの拡張が有望である。特に、「Im2Calories」[11]や「Nutrition5k」[17]を拡張してモバイルアプリケーションに組み込むことで、より高精度なカロリー推定を実現できる可能性がある。

6. まとめ

本稿では、iPhoneのLiDARセンサと深層学習モデルを活用し、食事の3次元形状を正確に測定することで、基準物体を用いずにリアルタイムなカロリー推定を実現するシステム「LiDARCaloriCam」を提案した。実験の結果、従来手法と比較して体積推定およびカロリー推定の精度向上が確認され、日常的な食事管理における応用可能性が示された。

しかしながら、実験結果から明らかになったように、認識できる食事カテゴリの種類が限られている点や、撮影条件の変動や食品の多様性に起因する課題が残されている。例えば、セグメンテーションにおいては微細な領域分割や複数における認識精度の向上が求められるほか、カロリー推定に関しては、カテゴリ別回帰式に依存しない多変量推定モデル[10]や深層学習による直接的なカロリー回帰[17]への拡張が有望である。

今後は、より多様な食品カテゴリに対応するための大規模な学習データの拡充、容器や皿など外部オブジェクトの影響除去、および液体や複雑形状の食品に対する推定精度の向上に取り組むことで、実用性の高い食事管理システムの実現を目指す。

参考文献

- [1] Akpa, E., Suwa, H. and Yasumoto, K.: Smartphone-Based Food Weight and Calorie Estimation Method for Effective Food Journaling, *SICE Journal of Control, Measurement, and System Integration* (2017).
- [2] Ando, Y., Ege, T., Cho, J. and Yanai, K.: DepthCalorieCam: A Mobile Application for Volume-Based Food Calorie Estimation using Depth Cameras, *Proc. of ACM MM Workshop on Multimedia Assisted Dietary Management (MADIMA)* (2019).
- [3] Chen, L.-C., Papandreou, G. et al.: Rethinking Atrous Convolution for Semantic Image Segmentation, *arXiv preprint arXiv:1706.05587* (2017).
- [4] Chi-en Amy Tai, Keller, M., Nair, S., Chen, Y., Wu, Y., Markham, O., Parmar, K., Xi, P., Keller, H., Kirkpatrick, S. and Wong, A.: NutritionVerse: Empirical Study of Various Dietary Intake Estimation Approaches, *arXiv preprint arXiv:2309.07704* (2023).
- [5] Chollet, F.: Xception: Deep Learning with Depthwise Separable Convolutions, *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1800–1807 (2017).
- [6] Ester, M., Kriegel, H.-P., Sander, J. and Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise, *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, p. 226–231 (1996).
- [7] Fang, S., Liu, C. et al.: Single-view food portion estimation based on geometric models, *IEEE International Symposium on Multimedia (ISM)*, pp. 385–390 (2015).
- [8] Inc., A.: Create ML, <https://developer.apple.com/jp/machine-learning/create-ml/>.
- [9] Matsuda, Y., Hoashi, H. and Yanai, K.: Recognition of Multiple-Food Images by Detecting Candidate Regions, *Proc. of IEEE International Conference on Multimedia and Expo (ICME)* (2012).
- [10] Meyers, A., Johnston, N. et al.: Im2Calories: towards an automated mobile vision food diary, *Proc. of IEEE International Conference on Computer Vision*, pp. 1233–1241 (2015).
- [11] Myers, A., Johnston, N., Rathod, V., Korattikara, A., Gorban, A., Silberman, N., Guadarrama, S., Papandreou, G., Huang, J. and Murphy, K.: Im2Calories: towards an automated mobile vision food diary, *ICCV* (2015).
- [12] Okamoto, K. and Yanai, K.: UEC-FoodPIX Complete: A Large-scale Food Image Segmentation Dataset, *Proc. of ICPR Workshop on Multimedia Assisted Dietary Management (MADiMa)*, p. 647–659 (2021).
- [13] Okamoto, K. and Yanai, K.: An Automatic Calorie Estimation System of Food Images on a Smartphone., *In Proc. of ACM MM Workshop on Multimedia Assisted Dietary Management (MADiMa)* (2016).
- [14] OpenCV: Camera Calibration and 3D Reconstruction, https://docs.opencv.org/4.11.0/d9/d0c/group_calib3d.html.
- [15] Ronneberger, O., Fischer, P. and Brox, T.: U-Net: Convolutional Networks for Biomedical Image Segmentation, *In Proceedings of Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pp. 234–241 (2015).
- [16] Tanno, R., Ege, T. and Yanai, K.: AR DeepCalorieCam V2: Food calorie estimation with cnn and ar-based actual size estimation, *In Proc. of the 24th ACM Symposium on Virtual Reality Software and Technology* (2018).
- [17] Thames, Q., Karpur, A., Norris, W., Xia, F., Panait, L., Weyand, T. and Sim, J.: Nutrition5k: Towards Automatic Nutritional Understanding of Generic Food, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8903–8911 (2021).
- [18] 文部科学省：日本食品標準成分表（八訂）増補 2023年, https://www.mext.go.jp/a/_menu/syokuhinseibun/\mext_00001.html.