

# DepthCalorieCam: 深度カメラと深層学習による 自動食事カロリー量推定システム

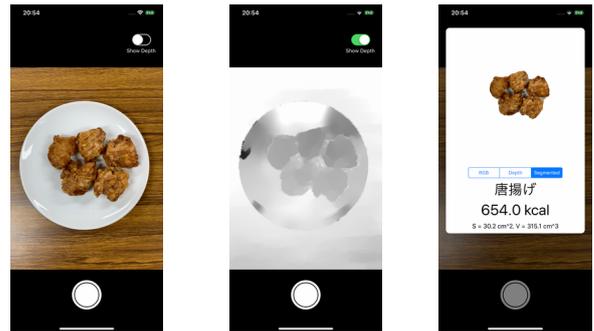
安蒜 祥和<sup>1,a)</sup> 會下 拓実<sup>2,b)</sup> 岡本 開夢<sup>1,c)</sup> 泉 裕貴<sup>2,d)</sup> Jaehyeong Cho<sup>2,e)</sup> 柳井 啓司<sup>2,f)</sup>

## 概要

近年のスマートフォンなどの携帯端末の普及や健康意識の向上による端末での食事内容やカロリーの記録の需要と、iOS における深度カメラのような端末の実世界を認識する能力の向上を踏まえて、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、酢豚・唐揚げ・コロッケについての評価実験において、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。

## 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。しかし、原則として食品のカロリー量はその体積に比例するが、これを考慮していないものや、体積を考慮していても既知の面積を持つ基準物体と食事を一緒に画像に収めなければならず利便性に難があり、また体積についても先述の方法で計算した食品の表面積をその代替としていて精度が低くなってしまっているものもある。一方で、端末の実世界の情報を認識する能力も年々向上しており、iOS においては特定のモデルにデュアルカメラの原理で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に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: DepthCalorieCam 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量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で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。

<sup>1</sup> 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

<sup>2</sup> 電気通信大学 大学院情報理工学専攻 情報学専攻

a) ando-y@mm.inf.uec.ac.jp

b) ege-t@mm.inf.uec.ac.jp

c) okamoto-ka@mm.inf.uec.ac.jp

d) izumi-y@mm.inf.uec.ac.jp

e) cho@mm.inf.uec.ac.jp

f) yanai@mm.inf.uec.ac.jp

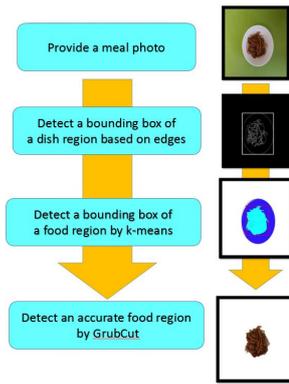


図 2: CalorieCam の食品領域の抽出手順 ( [6] から引用)

### 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。

食品の表面積  $S_{food}$  は、撮影時にあらかじめ表面積がわかっている基準となる物体を用意し、撮影した画像から図 2 のように GrabCut [8] などを用いて食品および基準物体領域を抽出し、そのピクセル数から式 1 で計算している。

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

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

このことからわかるように、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ったり、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、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に、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 AR を用いることで CalorieCam のこれを不要とした AR DeepCalorieCam V2 を開発している [9]。AR DeepCalorieCam V2 では図 3 のようにユーザーが端末を動かしながら AR 空間上で食品にアンカーを複数設置し、その位置関係から食品の表面積を推定している。

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まではアンカーの設置ができず時間がかかったり、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するという行為自体の難易度が高く、アンカー設置の精度がユーザーに大きく依存してしまう。また、AR DeepCalorieCam V2 についても、食品の体積ではなく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



図 3: AR DeepCalorieCam V2 の食品領域指定の様子 ( [9] から引用)

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。

## 3. 提案手法

本システムにおけるカロリー量推定は以下のような流れで行われる。

- (1) 撮影した RGB 画像について食品領域を分割
- (2) 分割された領域について RGB 画像と深度画像の各ピクセルごとにその体積を推定
- (3) RGB 画像を元に写っている食品カテゴリーを推定
- (4) 推定した体積と食品カテゴリーからカロリーの回帰式でカロリー量を推定

各詳細な手法について以下で説明する。

### 3.1 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 [7] を使い、データセットには UECFOOD-100 [4] 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 4 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 4: U-Net による領域分割例

### 3.2 深度付き画像からの実寸の推定

本システムでは、ピクセルごとの物体が設置されている基準面からの食品表面の高さおよびそこに写っているの面積をもつ長方体の体積を求め、すべてのピクセルの長方体の体積を合計することで食品の体積とする。推定の前段階として、カメラから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 5 のように物体が基準面を覆っている場合、目的のカメラ  $C$  と画像内の基準面の中心  $R$  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 5 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 2 で計算した距離の平均を基準面までの距離として扱う。

$$CR = CR_X \cos \angle RCR_X \quad (2)$$

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 5 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 3~5 で表される。

$$R_NR_S = 2CR \tan \frac{\angle R_NCR_S}{2} \quad (3)$$

$$R_ER_W = 2CR \tan \frac{\angle R_ECR_W}{2} \quad (4)$$

$$S_{pixel} = \frac{R_NR_S \times R_ER_W}{N_{pixel}} \quad (5)$$

そして最後に 3.1 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 6 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 6 のように定める。

$$r_p = \left( \frac{z_{ref}}{z_p} \right)^2 \quad (p \in P) \quad (6)$$

これを用いて最終的な体積  $V$  を以下の式 7 で求める。

$$V = \sum_{p \in P} \frac{S_{pixel}}{r_p} (z_{ref} - z_p) \quad (7)$$

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ため、上から見ることはできない食品下部の何もない空間の体積も含まれてしまう。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する事でカロリー量推定の精度を確保する。

### 3.3 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。モデルは既存手法の食事カテゴリー認識に用いられたものより認識精度が優れているとされている Xception [2] 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 [1] 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 [4] から抽出して Fine-tuning したものを利用している。

### 3.4 食品カロリー量の回帰

最終的なカロリー量はここまでで求めた食品の体積をカテゴリーごとのカロリー量の回帰式に当てはめることで計算する。なおこの回帰式については今回利用した酢豚・唐揚げ・コロッケについてそれぞれ惣菜店でカロリー量が既知のものを S・M・L の 3 種類の大きさ用意し、各サイズごとに 20~70cm の範囲の距離から撮影し、推定された体積と既知のカロリー量から作成した。

## 4. 実験

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 [9] 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。実験には先述のカロリー量の回帰式の作成に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 7 と表 1 に示す。

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

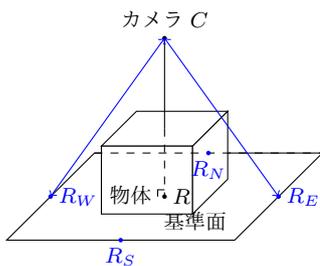


図 5: 基準面までの距離の推定

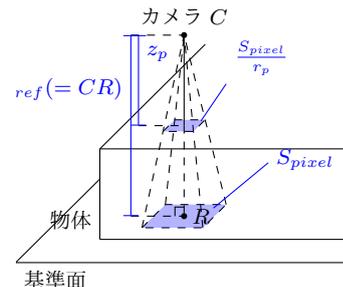


図 6: 体積の推定 ( $R$  はピクセルの中心)



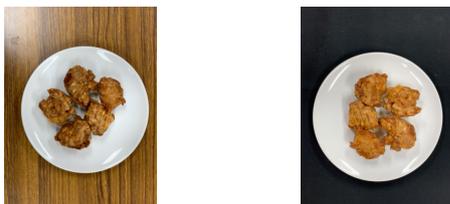
(a) 酢豚 (b) 唐揚げ (c) コロッケ

図 7: システムの評価に用いた食品

表 1: 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

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 8 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 8: 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 2 に示す。

表 2: 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

## 5. まとめと今後の課題

実験の結果より、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 [9] 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。

一方で、現在のシステムでは対応している食品カテ

グリーが 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、カロリー量が既知の食品を複数サイズ用意し、それを撮影してカロリー量の回帰式とカテゴリー認識器を作成する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。なお、本システム“DepthCalorieCam”は現在 App Store にてデモアプリを配信中\*1である。

## 参考文献

- [1] Bossard, L., Guillaumin, M. and Van Gool, L.: Food-101 – Mining Discriminative Components with Random Forests, *Proc. of European Conference on Computer Vision (ECCV)*, pp. 446–461 (2014).
- [2] Chollet, F.: Xception: Deep Learning with Depthwise Separable Convolutions, *Proc. of IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1800–1807 (2017).
- [3] Ege, T. and Yanai, K.: Image-Based Food Calorie Estimation Using Recipe Information, *IEICE Transactions on Information and Systems*, Vol. E101-D, No. 5, pp. 1333–1341 (2018).
- [4] 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).
- [5] Myers, A., Johnston, N., Rathod, V., Korattikara, A., Gorban, A., Silberman, N., Guadarrama, S., Papandreou, G., Huang, J. and Murphy, K. P.: Im2Calories: towards an automated mobile vision food diary, *Proc. of IEEE International Conference on Computer Vision (ICCV)*, pp. 1233–1241 (2015).
- [6] Okamoto, K. and Yanai, K.: An Automatic Calorie Estimation System of Food Images on a Smartphone, *Proc. of ACM MM Workshop on Multimedia Assisted Dietary Management (MADiMa)* (2016).
- [7] Ronneberger, O., Fischer, P. and Brox, T.: U-Net: Convolutional Networks for Biomedical Image Segmentation, *Proc. of Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pp. 234–241 (2015).
- [8] Rother, C., Kolmogorov, V. and Blake, A.: Grabcut: Interactive foreground extraction using iterated graph cuts, *ACM transactions on graphics (TOG)*, Vol. 23, No. 3, pp. 309–314 (2004).
- [9] Tanno, R., Ege, T. and Yanai, K.: AR DeepCalorieCam V2: Food Calorie Estimation with CNN and AR-based Actual Size Estimation, *Proc. of ACM Symposium on Virtual Reality Software and Technology (VRST)* (2018).

\*1 <https://apple.co/2Uaodog>