社団法人　電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
TECHNICAL REPORT OF IEICE.

全方位カメラを用いた複数人食事動作同時認識

寺内　健人†　　柳井　啓司†

† 電気通信大学 大学院情報理工学研究科 情報学専攻
E-mail: †terauchi-k@mm.inf.uec.ac.jp, ††yanai@cs.uec.ac.jp

あらまし　近年，世間では健康志向が高まるとともに，食事管理の重要性が高まっている．食事管理アプリでは，取った食事や摂取カロリー量を記録することで，食生活の改善に役立てることができる．しかし，既存手法では，複数人同時に食事を記録することはできない．したがって，全方位カメラを用いて食卓全体の食事を一度に全て取得することで手軽に食卓を囲む全員の食事を記録するシステムを構築する．

キーワード　食事認識, モバイルアプリケーション, 全方位カメラ

1. はじめに

近年，世間では健康志向が高まるとともに，食事管理の重要性が高まっている．食事管理アプリでは，取った食事や摂取カロリー量を記録することで自らの食習慣をよりよく把握し，食生活の改善に役立てることができる．既存手法では，1 人分の食事のみを記録する，もしくは摂食動作のみを記録するものが存在する．しかし，複数人同時に食事推定することはできない．家族で食卓を囲む場合などでは，今までのものでは全てを記録するには全員がアプリを使うなど，手間がかかってしまう．全方位カメラでは，2 つの広角カメラの映像を合成することでデバイス一つでテーブル、周りの人物など食卓全体を捉えることができる．そこで，全方位カメラを用いて食卓全体の食事を一度に全て取得することで手軽に食卓を囲む全員の食事を記録することで，簡素化を目指す．全方位カメラは近年 VR の普及や，低価格化などによってより身近なものになっている．しかし，全方位カメラでリアルタイムに深層学習を適用する研究はまだ少ないため工夫が必要である．また，全方位カメラは持ち歩くことを想定するためより手軽に持ち歩けるスマートフォンを用いる．

本研究では，全方位カメラとスマートフォンのみで食卓全体の食事を図 1 のように記録するシステム，CalorieCam360 を構築する．食卓全体の食事について，食事カロリー量を推定し，食卓全体の人物それぞれが摂取した食事とそのカロリー量についても記録する．食事カロリー量は，食事カテゴリと食事の実寸面積により推定できる．実寸面積はリファレンスの矩形物体を検出し，物体の面積をユーザーが入力することで求める．その後，物体認識で検出された食事カテゴリ，領域分割と組み合わせることでカロリー量推定を行う．全方位カメラから取得できる画像は多くの場合正距円筒図法で保存されており，特有の歪みがあるので，平面投影で補正する．また，食卓全体の人物について，それぞれの摂取カロリー量について記録するため，さらに人物を検出し，検出された料理と対応付ける．対応付けられた料理は前フレームとの分割領域の差分を各人物の摂取量として摂取カロリー量を計算する．

図 1: 食卓全体の食事を記録するシステムの概要

2. 関連研究

2.1 食事認識アプリ

食事管理アプリには，料理を撮影し，カロリー量を推定するもの [1]〜[7]，摂食動作を認識し，摂食カロリー量を推定するもの [8], [9] が存在する．しかし，既存のアプリケーションでは，一人分の食事しか認識せず，複数人で食卓を囲んだ場合を想定しておらず，複数人が一つの机についているにもかかわらずそれぞれアプリケーションを使う必要がある．本研究では，全方位カメラを用いることで食卓全体の食事を一度に認識することを試みる．

2.1.1 料理カロリー量推定アプリ

料理全体のカロリーを推定する手法は大きく分けて 3 つに分かれ，直接推定，面積ベース，体積ベースの手法が存在する．直接カロリー量を推定する手法 [1] では，マルチタスク CNN を用い，カロリー量とカテゴリ，具材，調理手順の情報を同時推定することで，カロリー量推定の精度を向上させている．直接推定するので，見た目だけで判断してしまい，誤差が大きくなりやすい．AR DeepCalorieCam [2] として iOS アプリとしても実装されている．

面積ベース手法は，CalorieCam [3] で提案されている．CalorieCam では，セグメンテーションによってピクセル数を数え，基準物体との比較により面積を特定する．その後，カテゴリごとに面積による回帰曲線を作り，面積からカロリー量を推定する．AR DeepCalorieCam v2 [4] では，AR 機能による実寸推定

により，基準物体なしに面積からのカロリー量推定を可能にしている．また，會下ら [5] は米飯の大きさを推定することによる実寸推定を行うことで基準物体の代わりとした．

近年のスマートフォンには，深度カメラが搭載されるようになったことで，体積ベースのカロリー量推定も可能になった．DepthCalorieCam [6] では，深度カメラで料理の表面と皿の底の基準面を取得することで，体積を推定し，カロリー量を推定できる．また，皿の基準面の代わりに，三次元推定を用いることで，変わった皿の形にも対応することを目指した HungryNetworks [7] がある．本研究では，カラー画像のみの全方位カメラを用いるため，体積を求めることが難しい．そのため面積ベースのカロリー量推定手法を用いてカロリーを推定する．

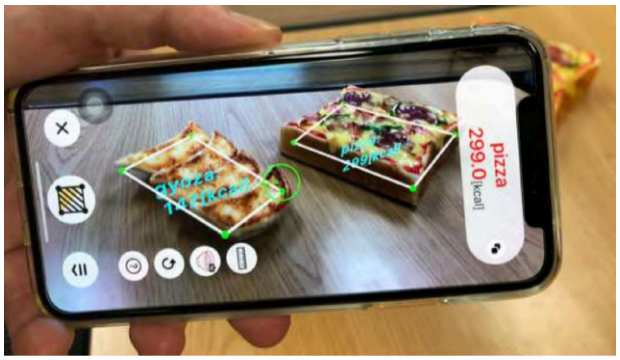

図 2: AR DeepCalorieCam v2 の AR による実寸推定 ([4] より引用)

2.1.2 摂食動作認識アプリ

食事画像のカロリー量推定ではなく，接触動作認識による摂取カロリー量推定の手法もいくつか存在する．摂食動作認識アプリでは，スマートフォンを机にセットし，顔を写して摂食する様子を撮影する．GrillCam [8] では，顔と口，箸を検出し，口と箸の先が重なったときに摂食動作として認識する．摂食動作が起こったとき，箸の先を GrubCut [10] でセグメンテーションし，取得したピクセル数から摂取カロリーを推定する．

DepthGrillCam [6] では，GrillCam に様々な改良が施されている．摂食動作の検出では，Deeplab v3+ [11] を用いて箸で持った料理を検出し，iOS の Vision フレームワーク [12] を用いて口のランドマークを検出している．箸で持った料理と口のランドマークが重なったときに摂食動作として検出する．摂食動作が検出されたとき，検出した食事をセグメンテーションし，深度情報と併せて体積を推定し．体積からカロリー量を推定する．本研究では，食事が口に入る摂食動作を検出する代わりに，テーブル上の料理が減る瞬間を検出し，それを人物と結びつけることで人物ごとの摂取カロリー量を推定する．

2.2 全方位物体認識

全方位画像の物体認識では，既存の研究では，一般的な画像に用いるものと同一の物体認識手法を用いるが，物体検出手法の適用にはいくつか方法が存在する．正距円筒図法の画像に直接アノテーションしたもので学習する手法 [13]，学習画像に正距円筒図法の歪みを加えて学習する手法 [14]，学習は一般的な画像を用いて行い，推論時に正距円筒図法の歪みを補正する手法 [15] 等がある．提案手法は正距円筒図法の歪みを投影を用いて補正する手法を用い，食事画像に適した平面投影を用いて検出する．

3. CalorieCam360

3.1 提案手法概要

全方位カメラ，スマートフォン用いて食卓全体の人物ごとの食事のカロリー量推定を目指す．全ての処理は，外部のサーバを用いず，全方位カメラ，スマートフォンのみで完結するようにシステムを実装する．深層学習モデルは，全て iOS の Vision フレームワーク [12] で完結するよう，CoreML モデルに変換して実装する．全方位カメラは Insta 360 ONE X2 を用いる．Insta 360 ONE X2 では，幅 1024，高さ 512 の正距円筒図法の画像をリアルタイムに取得することができる．

CalorieCam360 は図 3 の 4 つの段階に分けられる．(1) リファレンスサイズの決定，(2) 料理物体の検出，(3) 料理物体のカロリー量推定，(4) 人物ごとのカロリー量推定となる．

(1) リファレンスサイズの決定では，ユーザーが面積が既知の矩形物体を選び，面積を入力することで 1 ピクセルあたりの面積を推定し，料理の実寸推定を可能とする．(2) 料理物体の検出では，YOLO v7 [16] を UEC-FOOD100 [17] を用いて学習し，料理物体の位置とカテゴリを検出する．(3) 料理物体のカロリー量推定では，DeepLab v3+ [11] を用いて料理領域分割を行い，料理物体の実寸面積を計算し，面積からカロリー量推定を行う．(4) 人物ごとの食事カロリー量推定では，継続して料理領域分割を行い，面積の追跡をする．同時に人物の追跡も行い，人物と料理の対応付けを行うことで人物ごとに食事ごとの食べた割合を計算できる．食べた割合をもとに，人物ごとの食事カロリー量の計算が可能になる．それぞれの段階では，テーブルに置いた食事の情報を扱うため，平面投影を用いて歪みを補正したテーブルの画像を用いる．

図 3: 提案手法概要

3.2 テーブルを対象とした平面投影

全方位カメラから送られてくる映像は，正距円筒図法の画像として毎フレーム取得することができる．料理はカメラ下方のテーブルに置かれると想定するため，正距円筒図法では歪みが強くなってしまい，検出が難しい．歪みを補正し，検出に向いた画像にするため，テーブルを対象として平面投影する．平面投影では，食事がカメラ下方の水平面にあると仮定し，球として表せる正距円筒図法の画像を水平面に投影する．投影式は式 1 のように表す．この時 θ は緯度，ϕ は経度である．

$$x = \frac{\sin\phi}{tan\theta}, \quad y = \frac{\cos\phi}{tan\theta} \tag{1}$$

図 4: テーブル面に対する平面投影

3.3 リファレンスサイズの決定

料理のカロリー量を推定するために本研究では物体の面積を用いる．しかし，面積を求めるにはカメラに写った物体の実寸が必要になる．そこで，ユーザーが面積が既知な矩形物体を検出することで実寸を計算する．料理検出は平面投影後の画像で行うため，リファレンスサイズの計算も平面投影後の画像で行う．この際，平面投影後の画像にも歪みが残っているが，本研究では考慮しないものとする．アプリでは矩形検出をした後，ユーザーが既知の面積の矩形物体を選択し，面積を入力する．矩形検出では，Apple の Vision フレームワークの矩形検出機能を用いる．検出した矩形は左上，左下，右下，右上の 4 点で表され，その 4 点で囲まれたピクセル数を計算する．入力された面積を検出矩形のピクセル数で割ることで 1 ピクセルあたりの面積を求めることができる．

3.4 料理物体検出

物体検出では，全方位カメラで取得した画像を投影した後，UEC-FOOD100 [17] で学習した YOLO v7 [16] を用いて食事検出を行う．テーブル全体の画像を扱うため，料理が小さくなってしまう．小さいサイズの物体に対応するために，学習画像のデータ拡張において画像スケールを 0.04 倍から 0.3 倍にリサイズする．また，投影後の全方位画像は，料理が全方位から写るため，-180° から 180° に画像を回転する．他の設定は標準の YOLO v7 の設定に従う．画像の解像度は学習時 640，推論時 1280 に固定する．投影後画像の中央は全方位カメラの直下であり，死角になり黒くなる．そのため料理と認識されることがあるため，真ん中のバウンディングボックスは取り除く．

3.5 料理領域分割

料理物体を検出後，検出バウンディングボックスを領域分割することで料理のピクセル数を求め，ピクセル数をもとに面積を推定する．は，料理物体検出同様に平面投影後の画像に対して行う．料理領域の分割では，セマンティックセグメンテーションモデルの Deeplab v3+ [16] を用いる．Deeplab v3+ は領域分割ツールセットである MMSegmentation [18] を用いて実装し，バックボーンが ResNet50 [19] の標準設定を用いて学習する．データセットは UEC-FoodPIX Complete [20] を用いて学習する．UEC-FoodPIX Complete は 100 種類 1 万枚の食事画像データセットであり，それぞれの画像にはカテゴリごとに手動でピクセル単位のアノテーションがされている．データ拡張はクロップ，スケーリングの他に，回転，カットアウトを加え，食べかけの料理画像に対しても頑強になるようにする．

3.6 食事カロリー量推定

料理領域分割で求めた面積をもとに食事カロリー量の推定を行う．食事カロリー量の推定では岡元 [3] らの手法に従い，面積，カテゴリからカロリー量を計算する．會下ら [5] が作成したカロリー量の回帰曲線を用い，図 5 のように料理領域分割で求めた料理の面積，物体検出で検出したカテゴリでのカロリー量を求めることで料理のカロリー量とする．UEC-FOOD100 では，100 種類のカテゴリがあるが，全てのカテゴリの面積，カロリー量データがあるわけでは無いため，データがないカテゴリでは，カロリー量推定は行わない．

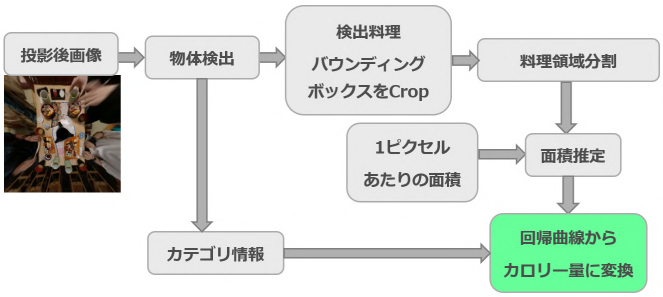

図 5: 食事カロリー量推定の流れ

3.7 人物ごとの摂取カロリー量推定

テーブルを使っている人物一人ひとりについての摂取カロリー量推定について考える．平面投影により，テーブル全体の料理を観測し，カロリー量を推定することはできるが，料理情報のみを扱うだけでは，テーブルを使っている人全員分の摂取量しか観測できない．したがって，テーブルを使用している人々の追跡も必要となる．

人物ごとの摂取カロリー量を推定する手法は図 6 のようになり，人物を検出し，人物と食事の対応付けをし，食事領域の減少量から人物ごとの食事の摂取量を求め，摂取カロリー量に変換する．人物の検出には，Vision フレームワークの骨格検出機能を用いる．骨格検出機能では，検出した人物ごとに首，頭，手，足などのランドマークの位置の情報を得ることができる．検出された人物は，前フレームで検出された最も近い人物と同一人物として扱うことで人物の追跡を可能にすることができる．この時，離れすぎている，もしくは前フレームより検出人物数が多いなどの理由で対応が見つからなかった人物は新しく検出した人物として扱う．

料理ごとに，現在食べている人物を対応付けることで，人物ごとの食べた料理の量を推定することができる．料理ごとに，最も近い手首，肘の人物を対応づける．この際，用いる画像は正距円筒図法で行うため，緯度軸は内回り，外回り両方からの距離を考慮し，近い方を計算に用いる距離とする．食事の摂取量は食事領域の減少量によって求める．最初全ての人物の全ての料理に対する摂取量を 0 とし，各フレームにおいて，それぞれの料理ごとに，対応する人物の料理の摂取量にそのフレームでの食事領域の変化量を足す．このようにすることで，人物ごとに料理ごとの摂取量が記録できる．摂取カロリー量は人物

ごとの摂取量から 3.6 同様に求める．最終的に，食事結果閲覧ボタンを押すことで人物ごとの食事の追跡結果を見ることができる．

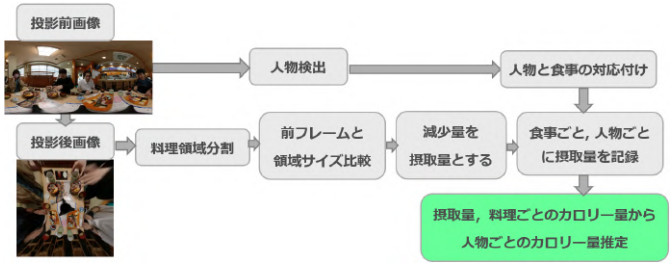

図 6: 人物ごとの摂取カロリー量推定の流れ

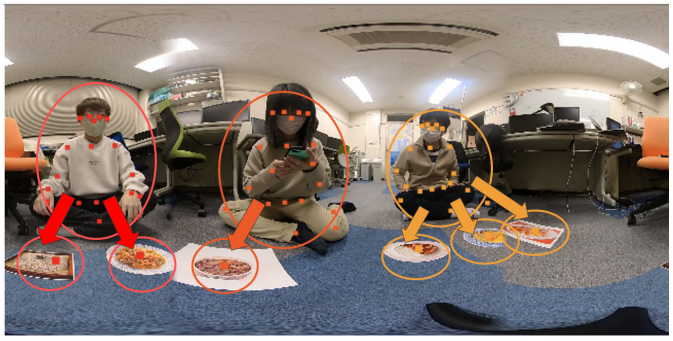

図 7: 人物と料理の対応付け

4. 実　　験

本章では，提案手法の機能の一部である物体検出，領域分割，カロリー量推定に関する有効性を確かめるために様々な実験を行う．最後に，実際の使用例を示し，CalorieCam360 を用いて食卓全体の食事記録ができることを示す．

4.1 物 体 検 出

取得画像に対して相対的に小さい物体に対する物体検出を行う必要があるため，解像度による影響が大きいと考える．そのため，料理物体検出に用いる画像の解像度による影響を調べる．学習時は 640 ピクセルで固定し，推論時に解像度を変えて推論を行う．小規模料理全方位画像データセットを作成し，物体検出による比較を行う．データセットは UEC-FOOD100 に含まれるカテゴリのうち 19 種類のバウンディングボックスアノテーションが含まれる全方位カメラで撮影した食事写真を平面投影した画像 37 枚となる．実環境で撮影された画像なため，ノイズが多く，挑戦的なデータセットとなっている．推論を行う解像度は 640，960，1280 ピクセルの 3 つを比較する．評価指標は Faster RCNN [21] と同様，mAP@0.5，mAP@0.5:0.95 の 2 指標を用いる．mAP@0.5 は IoU が 0.5 の時のクラス毎の適合率の平均であり，mAP@0.5:0.95 は IoU0.5 から 0.95 まで 0.05 毎の IoU のクラス毎の適合率の平均となる．

結果は表 1 のようになり，mAP@0.5 は推論の解像度 640 ピクセルが最も高くなっている．学習時と同様の解像度なためにこのような結果になったと考える．また，小規模データセットなこともあり，バイアスがかかり，取得画像に対し相対的に小さい物体の検出が高解像度画像でも難しかったことなども要因として考えられる．実際の検出の結果は図 8 のようになり，牛丼と麻婆豆腐など，色味の似た料理を混同する，ごはんが検出されないなど検出が失敗するときもある．

表 1: 物体検出結果の比較

推論時の解像度 (px)	mAP@0.5	mAP@0.5:0.95
640	0.304	0.126
960	0.269	0.199
1280	0.217	0.122

図 8: 料理物体検出結果

4.2 食事途中の料理画像に対する領域分割

料理の盛り付け直後の画像は図のように料理に沿って領域分割できていることが分かる．しかし，食事の過程を捉えることで人物ごとの食事量を記録する必要があるため，食事途中の料理画像に対しても正しく領域分割できる必要がある．食事途中の料理画像は自作の小規模食事途中全方位画像を用いる．

平面投影後の画像から料理がある部分を切り抜き，Deeplab v3+ を用いて領域分割した結果を図 9，10 に示す．それぞれのカテゴリの画像は 1280 × 1280 の画像から 250 × 250，200 × 200 の範囲を切り抜き，256 × 256 にリサイズした後領域分割した．領域分割は食事途中の料理画像に対してもうまく機能するが，手が写ると手が料理領域と認識してしまったり，図 10 のようにカテゴリによっては少し欠けるだけで正しく領域分割できなくなることがある．

図 9: ざるそばの食事途中の画像に対する領域分割 (250 × 250 の範囲を切り抜き)

図 10: 麻婆豆腐の食事途中の画像に対する領域分割 (200 × 200 の範囲を切り抜き)

4.3 カロリー量推定

推定の精度を示すため，既存の食事カロリー量推定アプリと比較する．カメラが異なるため，条件を合わせるのは被写体の食事のみとする．また，カロリー量推定のみに焦点をあてるため，提案手法では検出バウンディングボックスを手動で作成した上で領域分割からカロリー量を推定する．検出で評価する食事は，DepthCalorieCam [6]，提案手法両方でカロリー量を推定できる食事カテゴリである鶏のから揚げ，コロッケ，酢豚とする．それぞれの料理のカロリー量は，グラム当たりカロリー量，個数あたりカロリー量，または梱包あたりカロリー量として記載されているため，重量，個数を測ることで正解カロリー量として評価する．梱包あたりカロリー量の場合は重量を測り，グラム当たりカロリー量を算出する．実験は，テーブルの中央にカメラを置き，その横に料理を置いて測定する．測定では，料理ごとにカロリー量を固定し測定する，カテゴリごとのカロリー量は表 2 に示す．測定ごとに料理の盛り付けを変え，それぞれのカテゴリについて 6 回測定し，測定結果の誤差を平均する．正解カロリー量と推定カロリー量の差が小さいほど良い結果と言える．

結果は表 3 となる．酢豚，コロッケでは DepthCalorieCam の方が良い結果になっているが，鶏の唐揚げでは提案手法の方が正確なカロリー量を求めることができた．この結果は，提案手法では料理の場所とカテゴリを手動で指定しているため，食事物体をより良い精度にすることで，既存研究と同等に近い推定ができることがわかる．DepthCalorieCam では深度情報を用いており，料理を立体的に捉えている一方，提案手法は平面情報のみの不利な条件下で上回るカテゴリもあることは，提案手法がカロリー推定を正確にできることを示している．また，ほとんどの測定において測定カロリー量が上振れしていたため，回帰式を正確に作成しなおすことでより良い推定をすることが期待できる．

表 2: 料理ごとの測定するカロリー量

料理名	カロリー量 [kcal]
鶏の唐揚げ	510.3
コロッケ	255.0
酢豚	443.0

表 3: カロリー量推定誤差 [kcal]

料理名	CalorieCam	提案手法
鶏の唐揚げ	121.2	328.3
コロッケ	29.8	69.1
酢豚	492.9	952.9

4.4 CalorieCam360 の使用例

実際に使った結果は図 11，12，13，14 のようになる．準備として，図 11 のように (a) テーブルにカメラを設置し，(b) リファレンスサイズ決定画面で A4 用紙を映して A4 用紙を囲う矩形を検出する．(c) 検出できたことが確認出来たら A4 用紙の矩形を選択し，A4 用紙の面積である $624cm^2$ を入力し，リファレンスサイズ決定を終了する．

次に料理物体検出画面が表示されるので，図 12 のように (a) テーブルに料理を置き，置いた料理を検出する．(b) 検出出来たらボタンを押し，実際に検出が出来ているかを確認する．料理が検出出来たことが確認出来たら，人物ごとの摂取カロリー量追跡へ進む．

実際に料理を食べ進めると，図 13 のようにパーセント表記の食事の残りが減ることが分かる．

料理を食べ終えたら図 14 のように (a) それぞれの料理の残り量が少なくなっていることが確認できる．(b) 食事結果閲覧ボタンを押すと食事結果閲覧画面に進み，人物ごとの料理ごとの摂取量が分かる．カロリー量が検出できる料理カテゴリであった場合は摂取カロリー量も表示される．

結果として，牛丼は焼きそばと検出されてしまい，検出，追跡がうまく働かなかった．スパゲッティは検出は出来たが，追跡はある時点から領域分割が不安定になっている．餃子については検出，追跡がうまく働いた．食事結果閲覧画面と人物の対応は図 15 のようになり，正距円筒図法の画像に写った右側の人物が “人物 1”，中央の人物が “人物 2”，左側の人物が “人物 3” として表示され，最も近い料理を摂取したとして対応付けられていることが分かる．

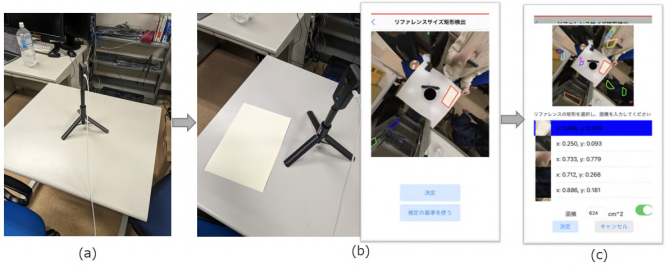

図 11: 実際の使用例：準備

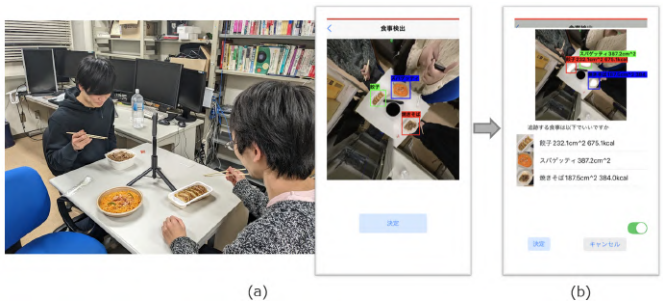

図 12: 実際の使用例：食前

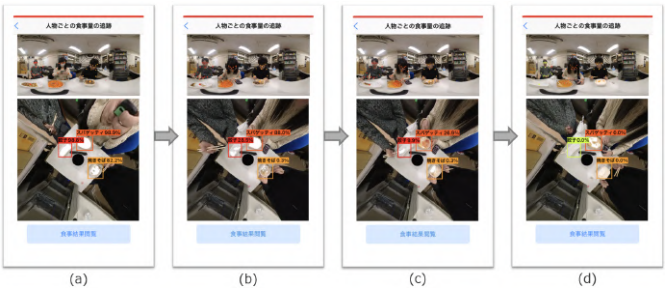

図 13: 実際の使用例：食事中

図 14: 実際の使用例：食後

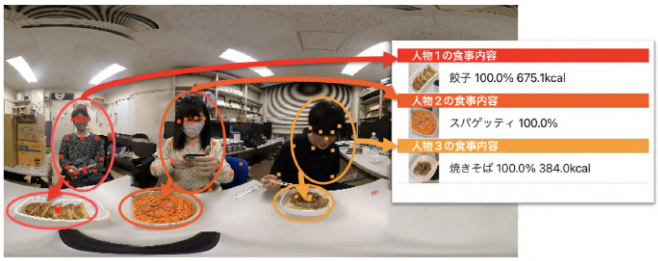

図 15: 実際の使用例：人物の対応

5. おわりに

全方位カメラを用いて食卓全体の食事を一度に全て取得することで手軽に食卓を囲む全員の食事を記録するため，全方位カメラとスマートフォンを用いた食卓全体の食事認識アプリケーションを提案した．アプリではリファレンスサイズの決定，料理物体検出，料理カロリー量推定，人物ごとのカロリー量推定の機能により，食事が始まってから終わるまでを記録し，各人物の各料理に対する食事量を求める．アプリの各機能は矩形検出，物体検出，領域分割，骨格検出を組み合わせることで実現した．物体検出，領域分割は料理画像データセットを用いて学習し，CoreML モデルに変換してスマートフォンに実装した．アプリケーションは全てのモデルを Vision フレームワークを用いてスマートフォン上に実装し，全ての動作をサーバーなどを用いず全方位カメラとスマートフォンで完結するようにした．実験により，各段階での手法の有効性を示し，同時に改善点についても特定した．

今後は，各段階のモデルの改良，顔検出による人物追跡，物体追跡による料理物体の追跡などの実装による使いやすさの向上に取り組んでいきたい．

文　献

[1] Takumi Ege and Keiji Yanai. Image-based food calorie estimation using knowledge on food categories, ingredients and cooking directions. In *Proc. of the on Thematic Workshops of ACM Multimedia*, 2017.

[2] Ryosuke Tanno, Takumi Ege, and Keiji Yanai. Ar deepcaloriecam: An ios app for food calorie estimation with augmented reality. In *Proc. of International Conference on Multimedia Modeling*, 2018.

[3] Koichi Okamoto and Keiji Yanai. An automatic calorie estimation system of food images on a smartphone. In *Proc. of International Workshop on Multimedia Assisted Dietary Management*, 2016.

[4] Ryosuke Tanno, Takumi Ege, and Keiji 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*, 2018.

[5] 松平 礼史 柳井 啓司會下 拓実. 米飯を基準とした CNN による食事画像からのカロリー量推定. 画像の認識・理解シンポジウム, 2019.

[6] Yoshikazu Ando, Takumi Ege, Jaehyeong Cho, and Keiji Yanai. Depthcaloriecam: A mobile application for volume-based food calorie estimation using depth cameras. In *Proc. of International Workshop on Multimedia Assisted Dietary Management*, 2019.

[7] Shu Naritomi and Keiji Yanai. Hungry networks: 3d mesh reconstruction of a dish and a plate from a single dish image for estimating food volume. In *Proc. of ACM International Conference on Multimedia in Asia*, 2021.

[8] Koichi Okamoto and Keiji Yanai. Realtime eating action recognition system on a smartphone. In *Proc. of IEEE International Conference on Multimedia and Expo Workshops*, 2014.

[9] Kento Adachi and Keiji Yanai. Depthgrillcam: A mobile application for real-time eating action recording using rgb-d images. In *Proc. of the 7th International Workshop on Multimedia Assisted Dietary Management*, 2022.

[10] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. ”grabcut”: Interactive foreground extraction using iterated graph cuts. *ACM Trans. Graph.*, 2004.

[11] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proc. of European Conference on Computer Vision*, 2018.

[12] Vision — apple developer documentation. https://developer.apple.com/documentation/vision.

[13] Shih-Han Chou, Cheng Sun, Wen-Yen Chang, Wan-Ting Hsu, Min Sun, and Jianlong Fu. 360-indoor: Towards learning real-world objects in 360° indoor equirectangular images. In *Proc. of IEEE Winter Conference on Applications of Computer Vision*, 2020.

[14] Yiming Zhang, Xiangyun Xiao, and Xubo Yang. Real-time object detection for 360-degree panoramic image using cnn. In *Proc. of International Conference on Virtual Reality and Visualization*, 2017.

[15] Wenyan Yang, Yanlin Qian, Joni-Kristian Kämäräinen, Francesco Cricri, and Lixin Fan. Object detection in equirectangular panorama. In *Proc. of International Conference on Pattern Recognition*, 2018.

[16] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. *arXiv preprint arXiv:2207.02696*, 2022.

[17] Yuji. Matsuda, Hajime. Hoashi, and Keji. Yanai. Recognition of multiple-food images by detecting candidate regions. In *Proc. of IEEE International Conference on Multimedia and Expo*, 2012.

[18] MMSegmentation Contributors. MMSegmentation: Openmmlab semantic segmentation toolbox and benchmark. https://github.com/open-mmlab/mmsegmentation, 2020.

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proc. of IEEE Computer Vision and Pattern Recognition*, 2016.

[20] Kaimu Okamoto and Keiji Yanai. UEC-FoodPIX Complete: A large-scale food image segmentation dataset. In *Proc. of ICPR Workshop on Multimedia Assisted Dietary Management*, 2021.

[21] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*, 2015.