

スマートフォンによる食事画像からの 自動カロリー量推定システム

電気通信大学

修士2年 柳井研究室 岡元 晃一

- 食事記録は非常に**重要**

個人での健康管理

Foodlog



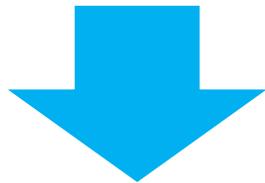
栄養学などでの調査



- しかし食事記録をつけるのは**難しい**

個人での健康管理

写真撮影
食事名の入力
カロリー量計算 etc...



めんどうくさい!!

栄養学などでの調査

食事記録法
24時間思い出し法
食物摂取頻度調査法 etc...



対象者への負担が多い
誤差が大きい

- アプリなど提案されているが課題もある

Foodlog



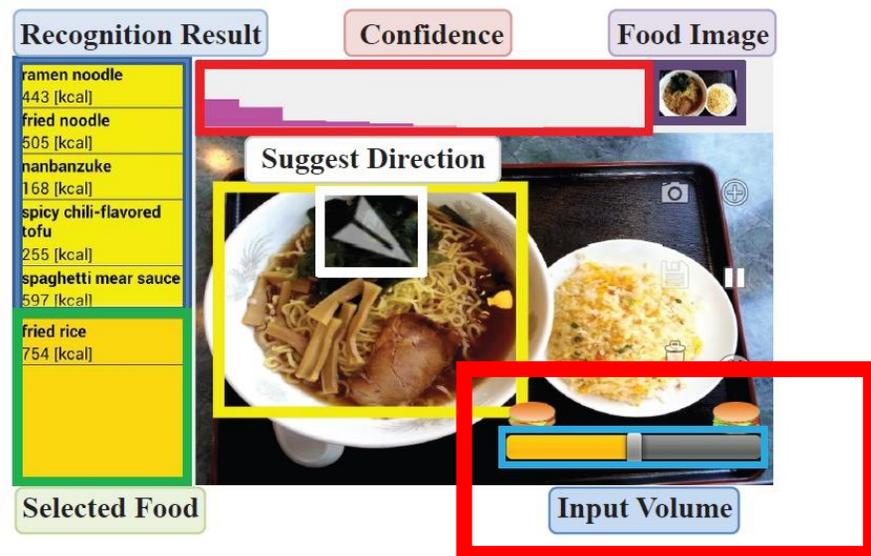
基準カロリー量のみ

カロナビ



お金と時間がかかる

- 今までも食事記録支援システムは研究されているが正確なカロリー量計算はできなかった
 - 簡易的な方法を用いるものがメイン



FoodCam(2014 河野ら)



GrillCam(2015 岡元ら)

- 今でも様々な方法が提案されている
 - サーバーとのやり取りが必要
 - ディープラーニングに全て任せる
 - 基準と一緒に撮影

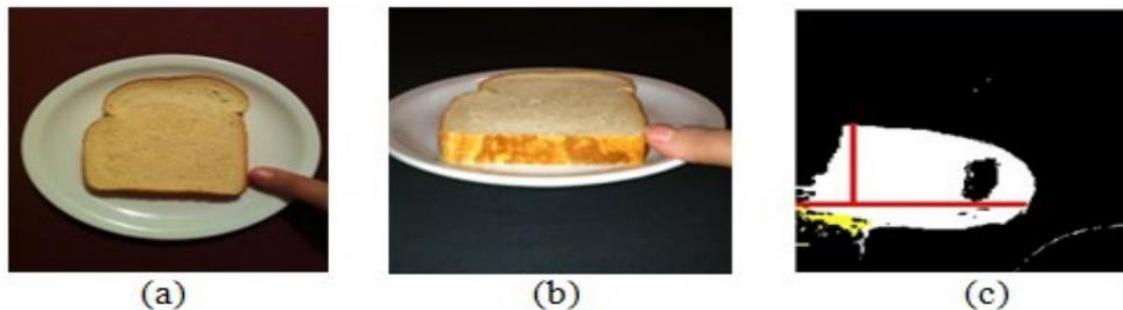
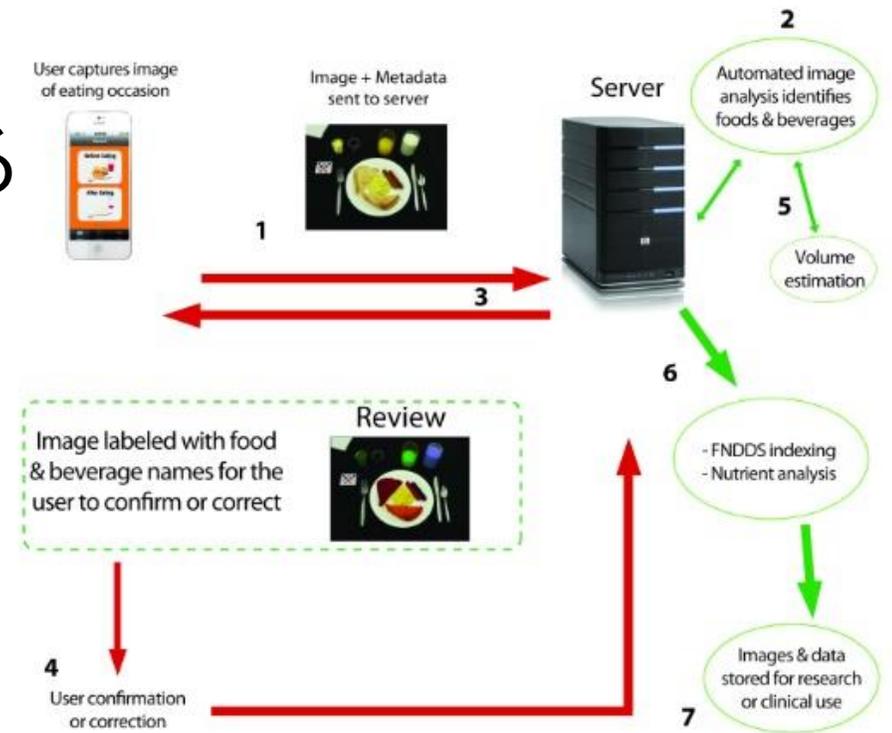


Figure 2 (a, b) Test images with food and thumb (c) Calculation of the thumb dimensions



*1 P. Pouladzadeh, S. Shirmohammadi, and R. Almaghrabi. Measuring calorie and nutrition from food image. IEEE Transactions. on Instrumentation and Measurement, Vol. 63, No. 8, pp. 1947–1956, 2014.

*2 The TADAProject www.tadaproject.org

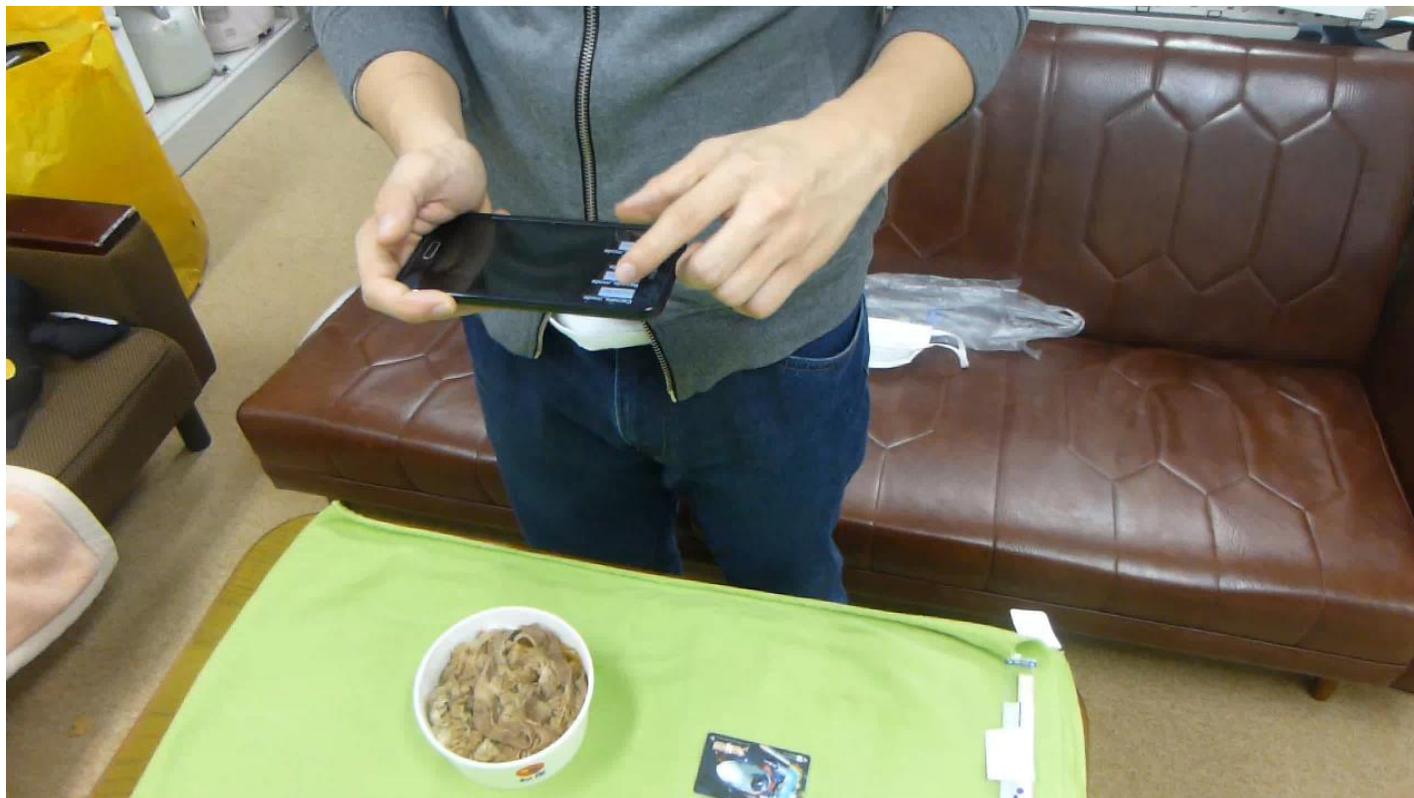
• 画像認識を用いた手軽なカロリー量推定

Androidアプリの作成

• 栄養の知識に影響されないシステム



食品名	カロリー量
とんかつ	528kcal
ご飯	172kcal
味噌汁	43kcal
サラダ	27kcal



実行時間は1秒未満！



実行時間は1秒未満！

Camera_mode

Camera_Start

Recode_mode

Recode_Check

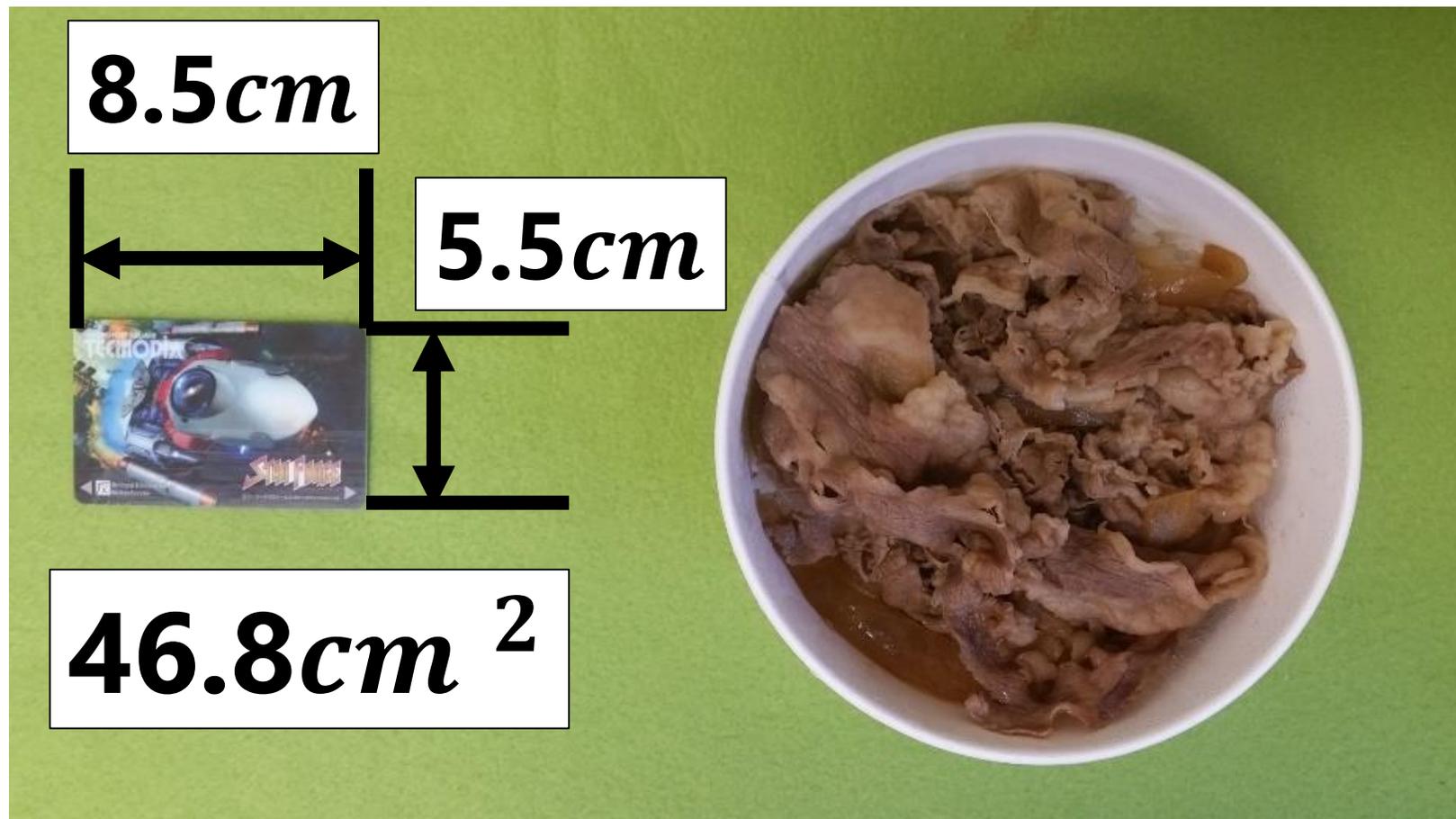
Test_mode

Test_Start

Touroku_mode

Touroku_Start

- 背景は無地
- 基準物体は左側
- 撮影は真上から



2つの面積を比較

基準物体の
領域抽出

食品の
領域抽出

食品面積情報

基準物体と食品を
撮影

食品
カテゴリ
認識

食品カテゴリ情報

回帰曲線に基づき
カロリー量推定

カロリー量を出力



画像を入力

食器矩形を探索

食品矩形を探索

食品領域を抽出



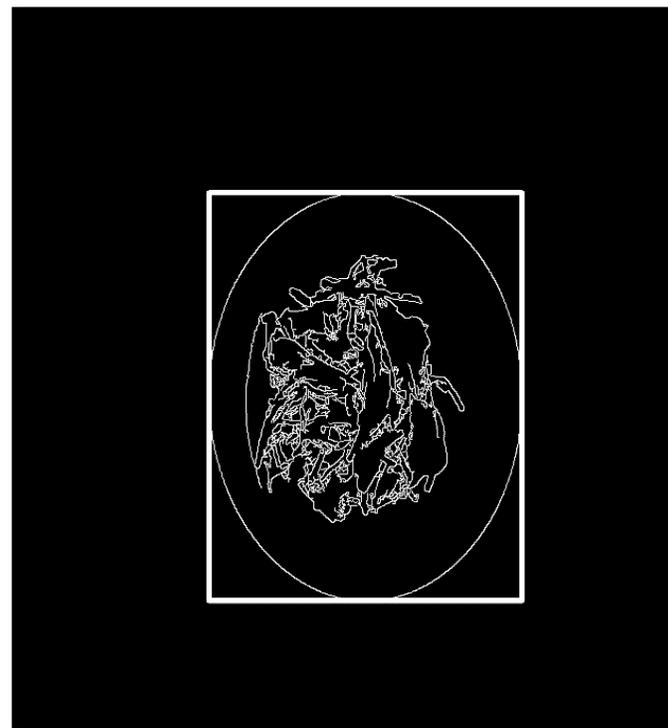
画像を入力

食器矩形を探索

食品矩形を探索

食品領域を抽出

- Canny法でエッジを検出
- 外周を食器矩形として選択



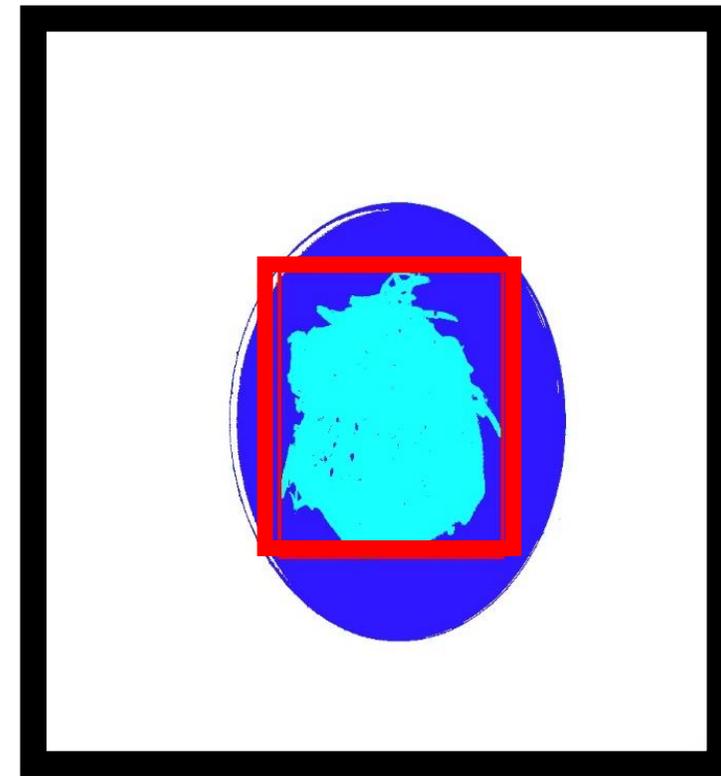
画像を入力

食器矩形を探索

食品矩形を探索

食品領域を抽出

- k-Meansで色情報をクラスタリング
- k=3とし中心から食品, 食器, 背景



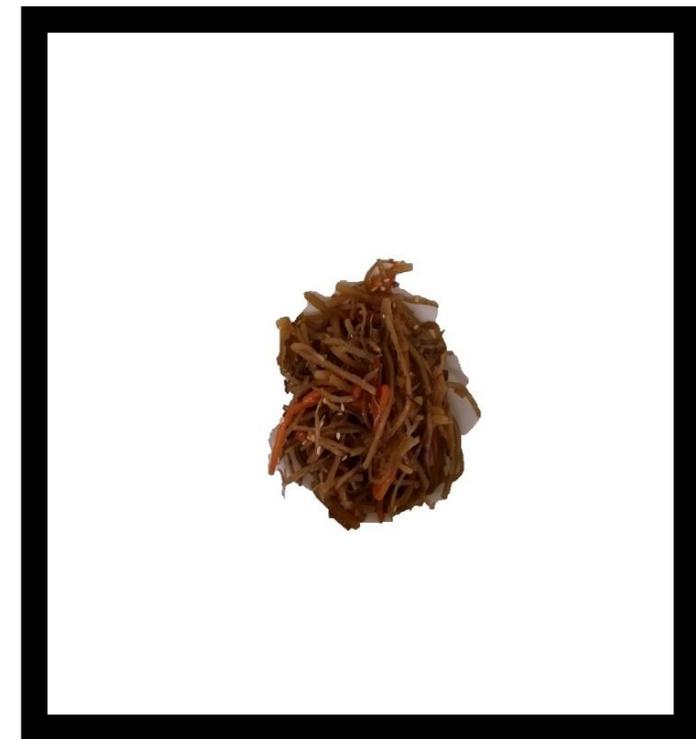
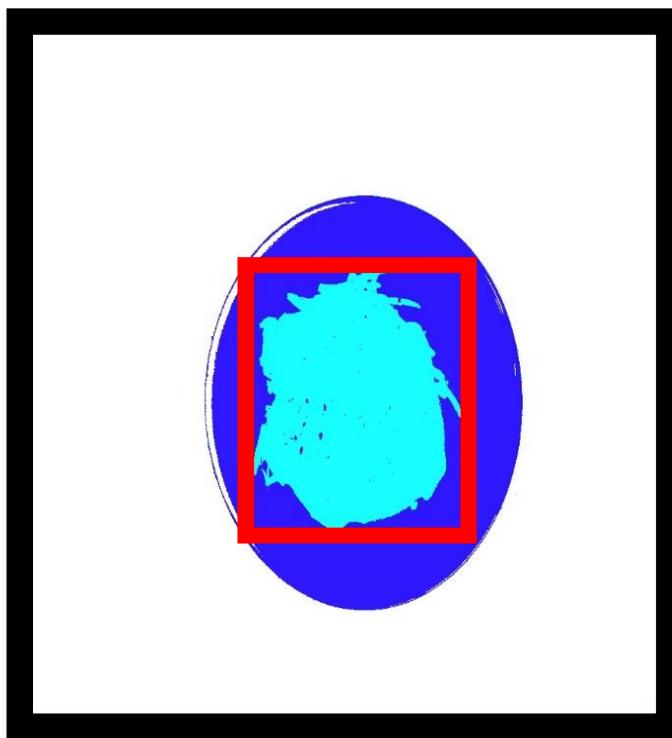
画像を入力

食器矩形を探索

食品矩形を探索

食品領域を抽出

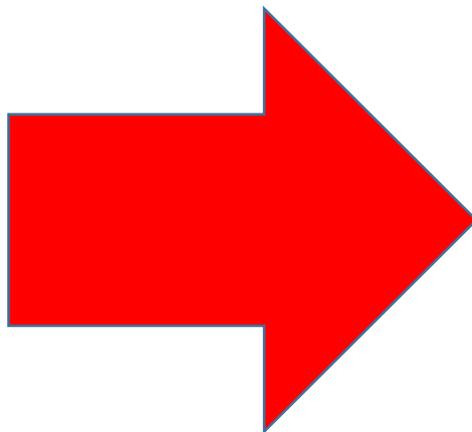
- グラフベースの分割手法GrabCutを用いる
- 矩形を与えることで分割



- 食品矩形を与える事で正確な**食品領域**を抽出



GrabCutのみ



食品矩形でのGrabCut

- 食品領域抽出と同様に**GrabCut**を用いて抽出
- 面積さえ分かっていたら形状などには関係ない

46.8cm^2

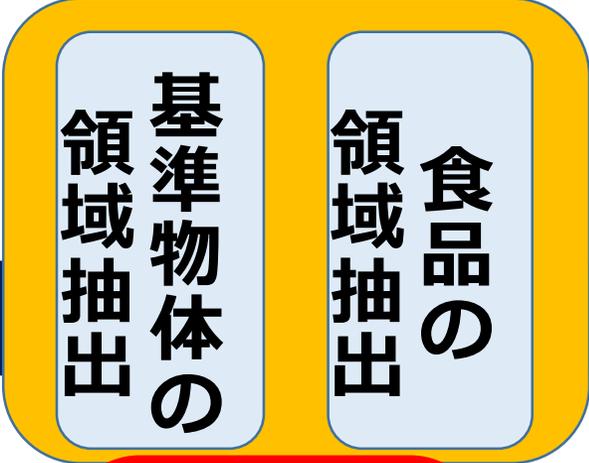


180.5cm^2





2つの面積を比較



食品面積情報

基準物体と食品を撮影

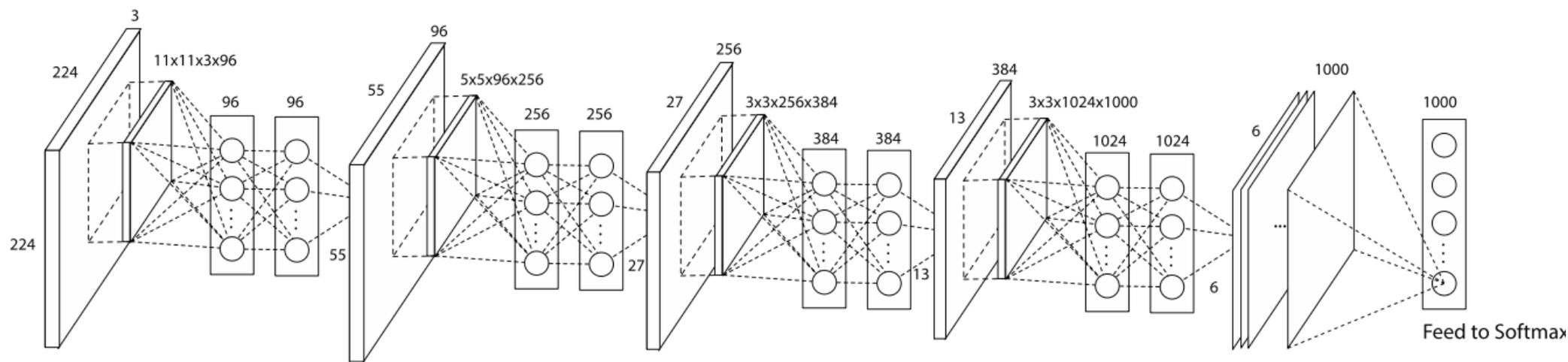


食品カテゴリー情報

回帰曲線に基づき
カロリー量推定

カロリー量を出力

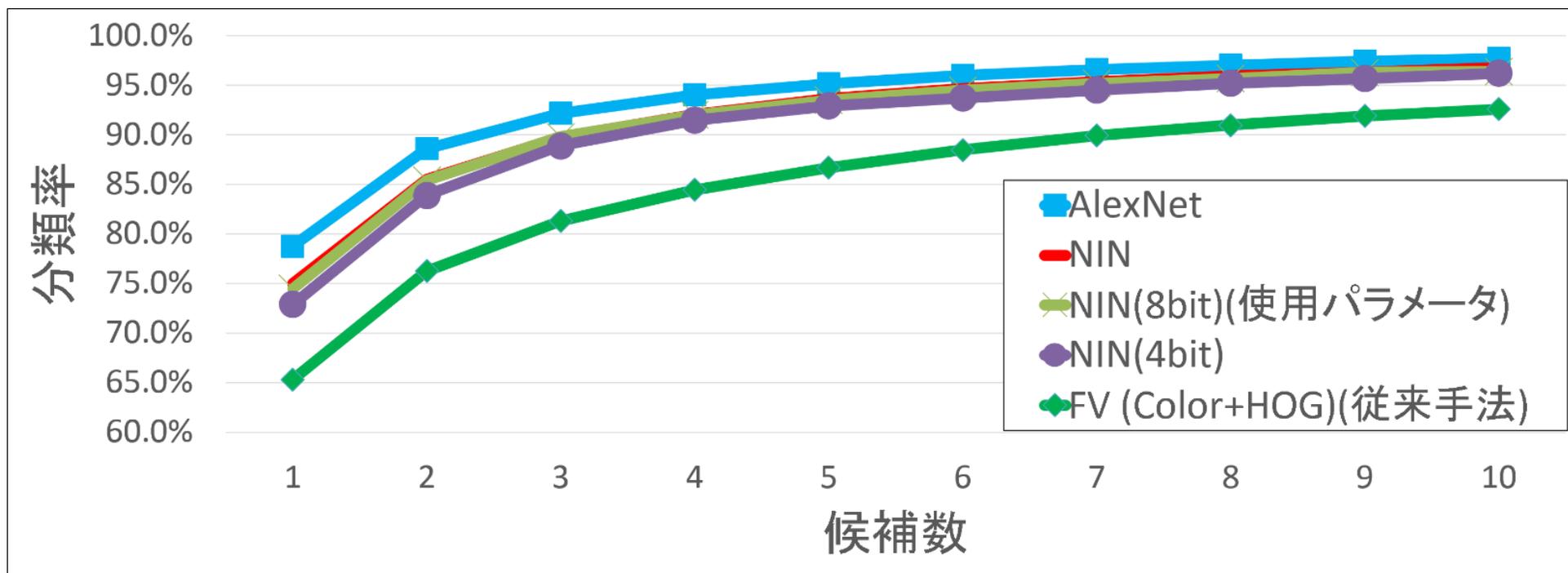
- Network in Network(NIN)
 - 全結合層を無くしパラメータ数を抑えたネットワーク
 - AlexNet : 6000 万パラメータ NIN : 750 万パラメータ
- **1/8**の圧縮が可能！



*1 L. Min, C. Qiang, and Y. Shuicheng. Network in network.
In Proc. of International Conference on Learning Representation Conference Track, 2013.

- 学習にはDCNNは4つの畳み込み層
- ILSVRC1000種類と食事に関連した1000種類のImageNet画像2000クラス210万枚で
プレトレーニング
- UECFOOD-100の100クラスと非食事写真1万枚の
101クラスでファインチューニング

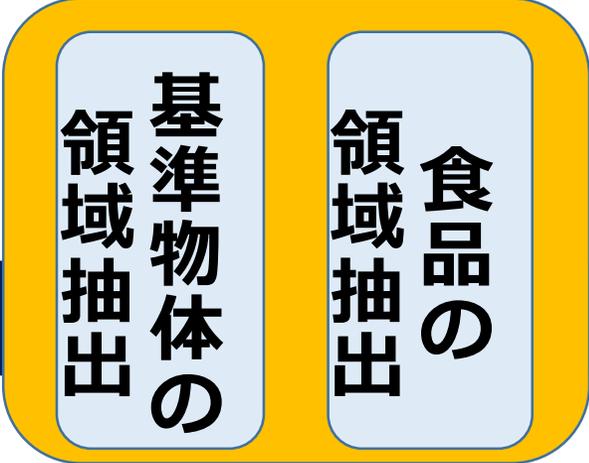
- 100種類での認識率：1位**75.0%** 5位以内**93.7%**





2つの面積を比較

食品面積情報



基準物体と食品を撮影



食品カテゴリー情報



カロリー量を出力

- 食品領域と基準物体領域のピクセル数から比率を求め、比率を基準物体の面積に掛け合わせる



基準物体
ピクセル数 : 43149
面積 : 46.8cm^2 (既知)



きんぴらごぼう
ピクセル数 : 68066
面積 : $??\text{cm}^2$



基準物体

ピクセル数 : 431

面積 : 46.8 cm^2

$$F_a = B_a * \frac{F_p}{B_p}$$

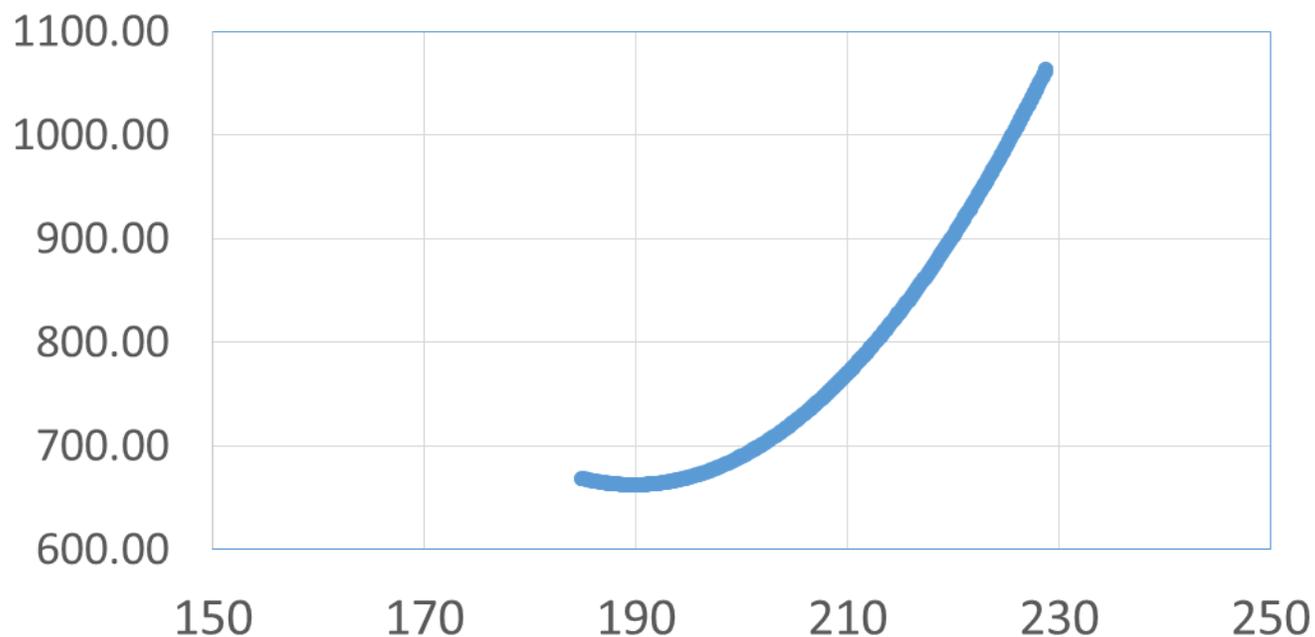
食品領域の面積 : F_a , ピクセル数

基準物体領域の面積 : B_a , ピクセル数

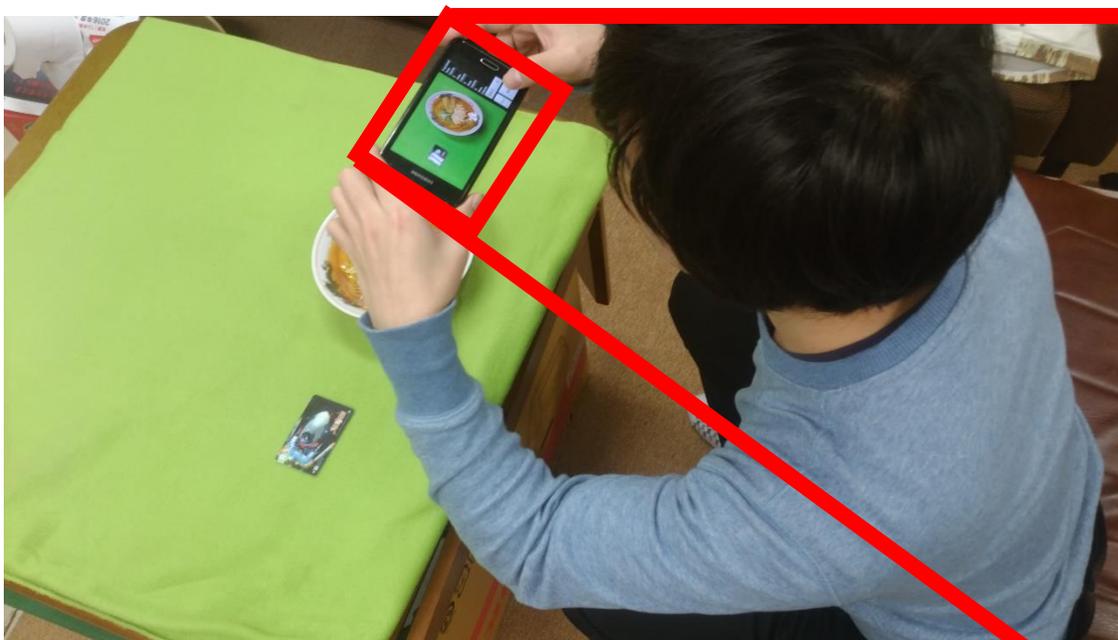


きんぴらごぼう

牛丼



比率 : **1.577** 面積 : **73.8** cm^2 カロリー量 : **74.4** kcal



Camera_mode

Camera_Start

Recode_mode

Recode_Check

Test_mode

Test_Start

Touroku_mode

Touroku_Start

Camera_mode

Camera_Start

Recode_mode

Recode_Check

Test_mode

Test_Start

Touroku_mode

Touroku_Start

- カロリー量推定実験
 - **PC**上での実施
- ユーザー評価実験
 - **スマートフォン**を用いて実施

- データセット
 - カロリー量のわかっており基準物体と共に映っている食品画像を事前に撮影し収集
 - 食品20種類 1食品3サイズごと1枚ずつの60枚



ご飯 340	ひじきの煮物 52	サラダ 10	ナポリタン 441	お浸し 54
牛丼 962	肉じゃが 170	酢豚 292	春巻き 428	コロッケ 552
唐揚げ 293	きんぴら 72	エビチリ 268	餃子 246	たこ焼き 241
焼きそば 283	トンカツ 345	焼き鳥 165	筑前煮 85	フライドポテト 454

カロリー量推定実験

20食品

60枚での実験

全体 誤差平均	標準偏差 平均	相対誤差 平均	相対 標準偏差
52.231 kcal	40.401 kcal	0.213	0.823

誤差：実際の値と推定カロリー量との差の絶対値

相対誤差：誤差を実際のカロリー量で割った割合

カロリー量推定実験(良かった例)

31

元画像	食品領域	真値	推定値	誤差	相対誤差
	トンカツ 	586 kcal	559 kcal	27 kcal	0.05
	牛丼 	1322 kcal	1417 kcal	95 kcal	0.07

カロリー量推定実験(悪かった例)

32

元画像	食品領域	真値	推定値	誤差	相対誤差
	肉じゃが 	170 kcal	122 kcal	48 kcal	0.28
	焼きそば 	425 kcal	519 kcal	94 kcal	0.22

- 被験者：栄養の知識のない**12**人
- 河野らの提案したFoodCamと今回の提案システムの2つでカロリー量を推定してもらおう
- 対象食品：牛丼，コロッケ，サラダ



比較対象システム FoodCam (2015 河野ら) 34

The screenshot displays the FoodCam application interface. On the left, a 'Recognition Result' list shows various food items with their corresponding calorie counts. The 'Selected Food' is 'fried rice' (754 [kcal]). The 'Confidence' bar shows a high level of certainty. The 'Food Image' shows a bowl of food. The 'Suggest Direction' feature is active, showing a camera view of the food. The 'Input Volume' slider is visible at the bottom, indicating the user's selection of portion size.

Recognition Result	Confidence	Food Image
ramen noodle 443 [kcal]	[Progress bar]	[Food image]
fried noodle 505 [kcal]		
nanbanzuke 168 [kcal]		
spicy chili-flavored tofu 255 [kcal]		
spaghetti mear sauce 597 [kcal]		
fried rice 754 [kcal]		

Suggest Direction

Input Volume

- 食事カテゴリーごと
決まった基準カロリー量
から0.5~1.5倍する

ここで大きさを
選択する

食品名	真値	FoodCam		提案システム	
		誤差平均	標準偏差	誤差平均	標準偏差
牛丼	962	-53.25	209.79	-242	55.10
コロッケ	552	-242	91.26	-47.08	52.52
サラダ	14	54.83	36.28	4.86	11.87

誤差を減らすことに成功
 標準偏差においても小さくすることができ知識のない使用者でもブレを少なくすることができた

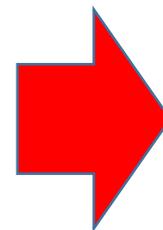
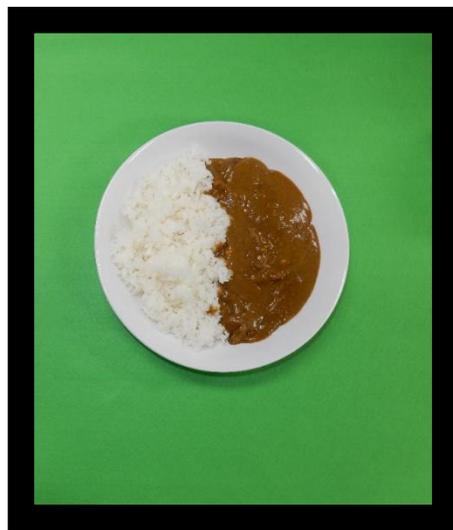
- カロリー量の記録の取りやすさの評価

	記録の取りやすさ
FoodCam	2.83 ± 0.80
提案システム	4.25 ± 0.72

- 操作が単純だったことが高評価につながったと考えられる

- 基準物体と共に撮影することでカロリー量を推定するシステムを作成
- エッジ検出や色特徴を用いて正確な食品領域の抽出
- 誤差は平均で**52**kcal, 相対誤差においても**20**%程度
- ユーザーからも既存システムよりも記録を取りやすいという評価を得た

- 領域抽出の改善
- 複数品目への対応方法の改善
 - どちらも最新のDCNNで可能？
 - 計算コストの問題



実際の推定値 良かった結果

39

食品名	正解カロリー量			推定カロリー量			誤差平均	標準偏差	相対誤差平均
	S	M	L	S	M	L			
トンカツ	120	345	586	110	318	559	21.2	7.46	0.07
たこ焼き	160	240	321	159	270	354	21.3	15.4	0.08
牛丼	737	962	1322	665	1032	1417	79.2	73.6	0.08
コロッケ	184	368	552	212	357	513	25.7	27.4	0.09
お浸し	30	54	81	34	47	79	4.23	4.62	0.10

実際の推定値 悪かった結果

40

食品名	正解カロリー量			推定カロリー量			誤差平均	標準偏差	相対誤差平均
	S	M	L	S	M	L			
肉じゃが	120	170	276	166	122	116	84.6	84.2	0.42
サラダ	6	10	14	12	9	13	2.58	3.34	0.40
酢豚	195	292	476	148	201	222	130.2	88.7	0.36
ごはん	221	340	459	147	126	438	102.6	81.3	0.34
焼きそば	188	283	425	241	395	519	86.2	25.0	0.30

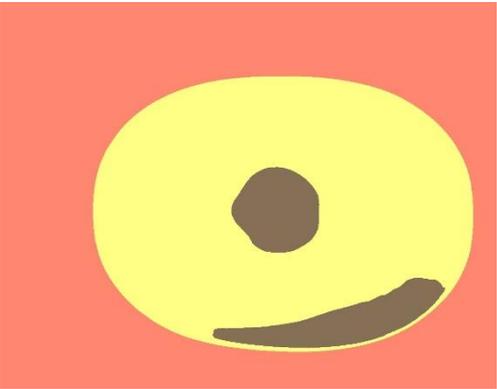
	牛丼	コロッケ	サラダ	記録の 取りやすさ
t値	3.012	-6.066	4.438	-4.214
p値	0.011	$8.118e^{-5}$	$9.981e^{-4}$	$1.450e^{-4}$

優位水準：5% t検定の両側の値：2.200

テストセット画像

42

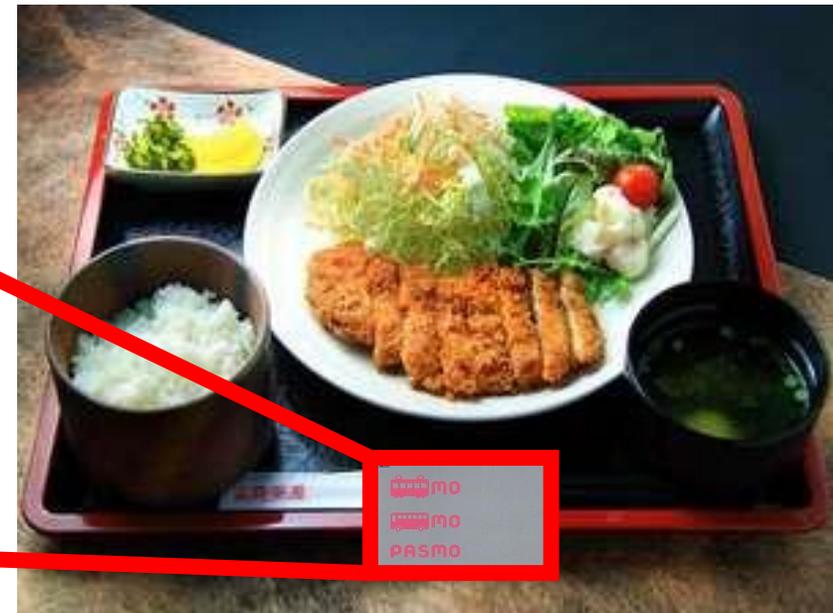


元画像	k-Means結果	GrabCut結果
		
		

- そこで写真撮影だけで自動的にカロリー量推定ができると大変**便利**である
- 栄養士が見てカロリー量推定をするサービスがあるが
お金や時間がかかり気軽に利用できない



- 大きさ推定には**基準物体**を登録しておき、それを一緒に映し、大きさを比較する
(以下の例ではPASMO)



画像を入力

食器矩形を探索

食品矩形を探索

食品領域を抽出

