Real-Time Mobile Recipe Recommendation System Using Food Ingredient Recognition

ACM Multimedia 2012 Workshop
Interactive Multimedia on Mobile and Portable Devices (IMMPD'12)

Takuma Maruyama, Yoshiyuki Kawano, Keiji Yanai
(The Univ. of Electro-Communications, Tokyo)
[**Demo**] Recipe search by object recognition at a grocery store

- Point a camera to carrots
- Object recognition
- Recipe lists
- Select one from the list
- Get to know how to cook

**mobile × recipe search × object recognition**
Background

OpenCV × Smartphone

Easy to implement an object recognition system on smartphones
Proposed System

**mobile × Recipe search × Object Recognition**

**Easy recipe search while shopping**

**Recognition performed on a smartphone**

**The first system for recipe search with object recognition on smartphones**

Impossible to use both hands for operation of mobile devices while shopping at a grocery stores
Related work — mobile image recognition —

Google Goggles

- Sudoku
- Specific object recognition
- Similar image search
- OCR

Object recognition for generic object is impossible

Implement useful app with generic object recognition
The flow of the proposed system

1. Point a camera to food ingredients

2. Recognize food ingredients

3. Search an online recipe database

4. Display a menu list

The system recognizes photo stream repeatedly and continuously
Guacamole Grilled Cheese Sandwich

Ingredients  categorized  original

- This recipe is vegetarian

Produce
- Avocados, ripe (2)
- Cilantro, leaves (2 tablespoons)

Bread
- Whole grain bread (2 slices)
- Grilled (optional)

Cheese
- American cheese (2 slices)
- Cheddar cheese (2 slices)

Salsa
- 1/4 cup of your favorite salsa (optional)

Directions:
1. Preheat a grill or grill pan.
2. Toast the bread on one side.
3. Place the cheese slices on the toast, cover with a slice of avocado, and top with salsa if desired.
4. Place the top slice of bread on the sandwich, grill for 1-2 minutes, or until the cheese is melted.

Nutritional Information:
- Calories: 450
- Protein: 20g
- Total fat: 18g
- Saturated fat: 6g
- Carbohydrates: 45g
- Dietary fiber: 10g
- Sodium: 450mg

Serves: 2

Source: www.twopeasandtheirpod.com/guacamole-grilled-cheese-sandwich
Obtain recipe list from WebAPI

Currently, the order of the recipe on the screen is the same as the results from WebAPI.

Need to be improved
Method for Object Recognition

Image Features

- Bag-of-Local Color Histogram, Bag-of-SURF (with KD-tree, 1000 dim codebook)

Classification method

- Linear SVM for saving memory and speed-up

Can recognize a food ingredient in 0.17 second
### Time for recognition

<table>
<thead>
<tr>
<th>Phone</th>
<th>Avg. (msec.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>GAXALY S2</td>
<td>167.8</td>
</tr>
<tr>
<td>HTC Desire HD</td>
<td>394.0</td>
</tr>
</tbody>
</table>

- **Galaxy S2**
  - June 3, 2011
  - 1GHz Dual Core
  - 1GB RAM

- **HTC Desire HD**
  - Nov. 12, 2010
  - 1GHz
  - 756MB RAM
Feature(1): Local Color Histogram

Divide color space into 64 bins (4x4x4)

12 x 12 grids → 144 local color histogram
**Feature (2): SURF**

4dim * 4x4 grids = 64 dims. 

\[ \sum dx, \sum dy, \sum |dx|, \sum |dy| \]

Tested in two ways of keypoints

- Default way (Hessian-based)
- 12x12, 24x24, 48x48, 96x96 multi-scale grid
**Bag-of-Features**

**Input image** → **Local features** → **BoF vectors**

Vote each local feature to corresponding visual word

Codebook size: \(N=1000\) (for both local color hist. and SURF)

**Use KD-tree** to search for visual words: \(O(\log N)\)
Multi-frame

Aggregate local features extracted from multi frames and build a BoF

- Single Frame

- Multi Frame

in experiments $n=1, 2, 3, 4, 5$
Linear SVM for speed-up and saving memory

**Linear SVM classifier**

\[ y(x) = \sum_{i=1}^{N} w_i x_i^T x + b \]

\[ = x \sum_{i=1}^{N} w_i x_i + b \]

\[ = x^T w + b \]

Inner product

\[ O(DN) \]

- \( y \): output
- \( x \): input
- \( x_i \): support vector
- \( w_i \): support vec. weight
- \( N \): # support vectors
- \( D \): dimension of Bof (1000)

*Can be computed off-line*

**Training:** offline (on PC), **Classifying:** on-line (on smartphone)
Linear SVM for speed-up

Estimation of required memory

Codebook (1000-dim)

$1000 \times 64 \times 4\text{byte} \approx 256\, K\text{byte}$

Weight W (for 30 class)

$1000 \times 30 \times 4\text{byte} \approx 30\, K\text{byte}$

Fast & small memory

Limitation: Android app -> 16MByte
Experiments

Dataset

30 kinds of food ingredient: 10 5-sec videos for each

Evaluation

Recorded at 10 grocery stores in Tokyo

Classification rate (features, food, top k)

Evaluate by 10-cross validation

User study

食材3種類を対象に目的のレシピを検索、時間

システムの評価、ユーザのコメント
### 30 Kinds of Food Ingredient

<table>
<thead>
<tr>
<th>Meat</th>
<th>Chicken</th>
<th>Beef</th>
<th>Pork</th>
<th>Ham</th>
<th>Sausage</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Minced Meat</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fish</td>
<td>Tuna</td>
<td>Squid</td>
<td>Octopus</td>
<td>Shrimp</td>
<td>Salmon</td>
</tr>
<tr>
<td>Vegetable</td>
<td>Potato</td>
<td>Mushroom</td>
<td>Carrot</td>
<td>Eggplant</td>
<td>Lettuce</td>
</tr>
<tr>
<td></td>
<td>Radish</td>
<td>Tomato</td>
<td>Cucumber</td>
<td>Chinese Cabbage</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Shitake</td>
<td>Onion</td>
<td>Green Onion</td>
<td>Cabbage</td>
<td></td>
</tr>
<tr>
<td>Fruit</td>
<td>Apple</td>
<td>Banana</td>
<td>Pineapple</td>
<td>Orange</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Strawberry</td>
<td></td>
<td>Grapefruit</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Examples

Grapefruit
Recognized correctly
recipe

Salmon
Touch 3rd candidates
recipe

Recorded at 10 grocery stores
Difficulty in the Dataset

- Packed or wrapped
- Light reflected
- Looks similar
Classification rate for features

Classification rate

0.05 0.1 0.15 0.2 0.25 0.3 0.35 0.4 0.45 0.5

- SURF
- SURF(GRID)
- RGB
- HSV
- La*b*
- RGB(3frame)
- SURF-RGB(3frame)

42.5% 44.9%


For 30 kinds

- Orange 81.1%
- Shrimp 5.14%
[best] orange: Orange color is specific to oranges.

[worst] shrimp: color is too various. Freezed, boiled, raw
Confusing food ingredient

Apple: 2 types (green and red)

Tomato, grapefruit
Classification rate within top k

Within top 6: 84.1%
**User study**

**Method**

3 ingredient x 3recipes = 9 patterns

Measure times when using obj. rec. and manually

- Recognition accuracy (5 step)
- Usability (5 step)
- Which is better, obj. rec. or manually? (5 step)

**Comments**

**# Subjects**

5 students
User evaluation (1)

Usability

- 5 (very good)
- 4 (good)
- 3 (soso)
- 2 (bad)
- 1 (difficult)

Recognition Accuracy

- 5 (very good)
- 4 (good)
- 3 (soso)
- 2 (bad)
- 1 (very bad)
User evaluation (2)

Which is better to use, by object recognition or manually?

When items were recognized correctly, obj. rec.-based was better.

If accuracy is improved, it must be more practical.
It is convenient, because easy to search for recipe at a grocery.

If accuracy is improved, I’d like to use it regularly.

Accuracy seems to depend on kinds of ingredient greatly.

Taking account of the prices would be more helpful.
Conclusions

Propose system

✓ By only pointing food ingredient, we can search for the cooking recipe at a grocery store.
✓ It takes 0.17[sec] for one-time recognition of 30 kinds of food ingredient.

User study

The users pointed out improvement on recognition accuracy is needed.

Evaluation

Classification rate: 44.9% (1\textsuperscript{st})
84.1% (\sim 6\textsuperscript{th})
Future work

**System**

- Toward being more practical
  - Search recipe DB by combination of several ingredients (AND-search)
  - Taking account of budget and price for low-cost cooking
  - Taking account of ingredient left in the fridge

**Recognition**

- Improve rate and increase kinds
  - Find better features and make DB bigger
You can try it!

The Android app can be downloaded!
*(We do not provide iOS version.)*

**Download site**

http://mm.cs.uec.ac.jp/mobile_recipe/