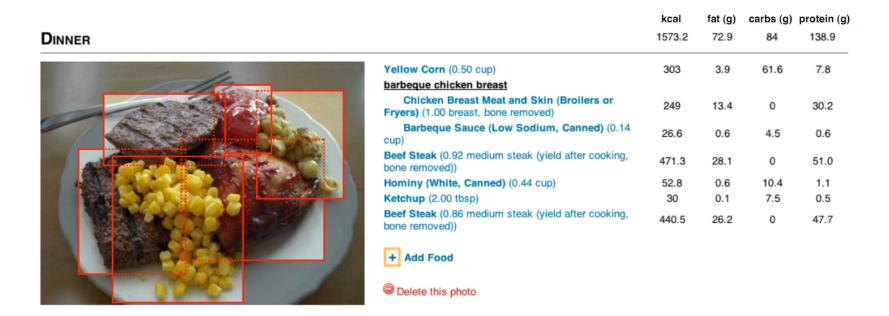


# PLATEMATE

# CROWDSOURCING NUTRITION ANALYSIS FROM FOOD PHOTOGRAPHS

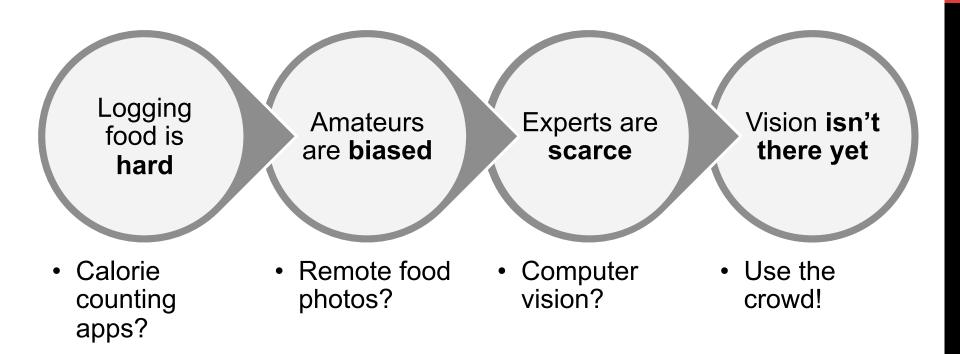
JON NORONHA
ERIC HYSEN
HAOQI ZHANG
KRZYSZTOF Z. GAJOS

### **OVERVIEW**



- Combines many crowdsourcing techniques
- Accuracy comparable to trained dietitians

### **MOTIVATION**



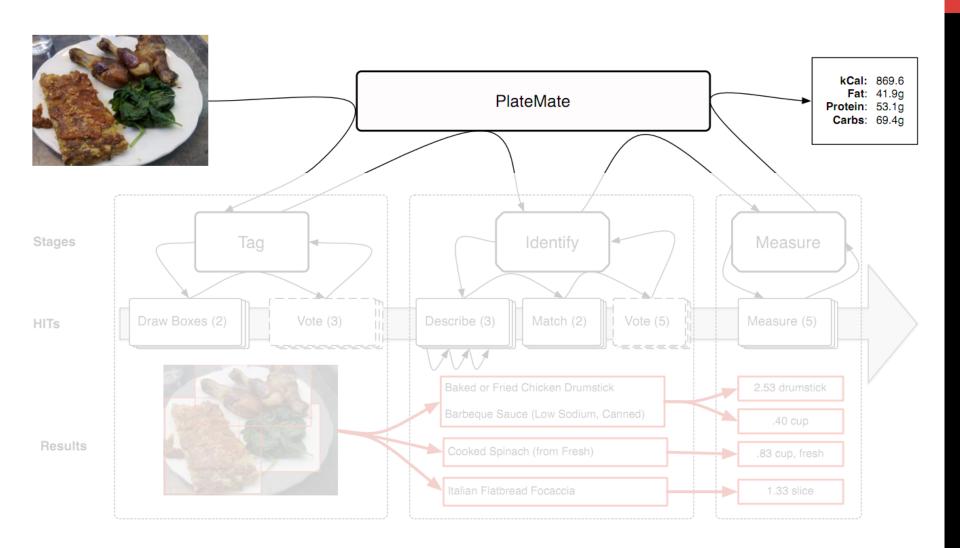
# CHALLENGE: SOLVE A COMPLEX REAL-WORLD PROBLEM WITH AN UNTRAINED CROWD

## **APPROACH**

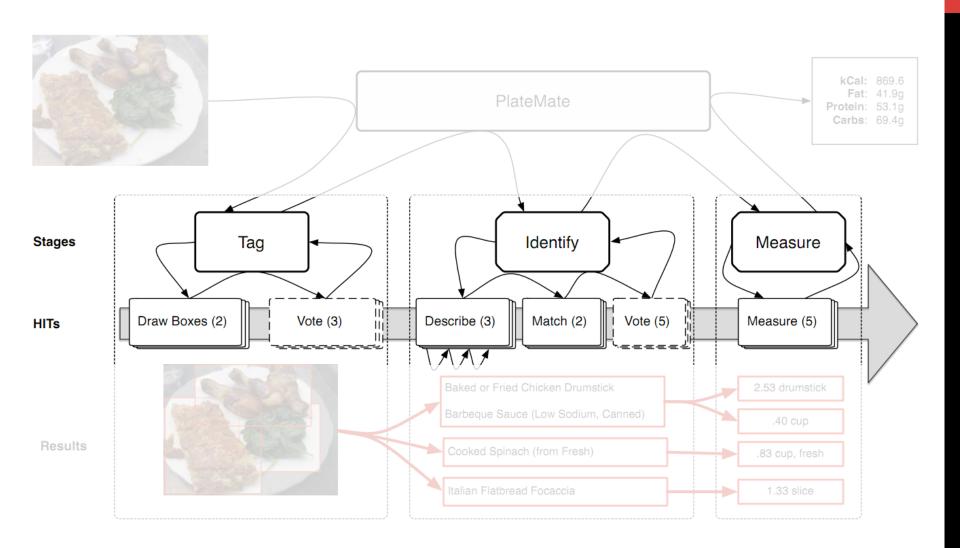
- Observed professional nutritionist
- Noticed three steps
  - \* Partition plate into distinct items
  - 1. Identify foods
  - 2. Estimate portions
  - 3. Calculate calories

# **WORKFLOW**

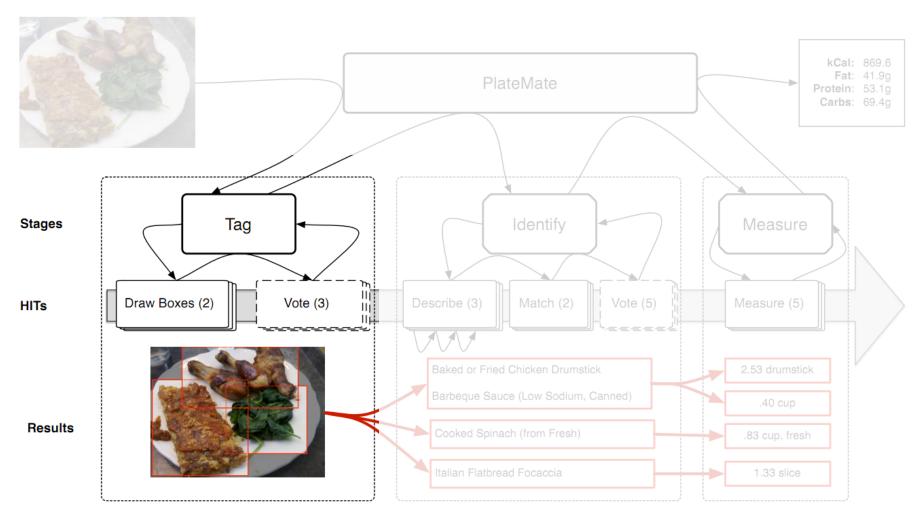
# **OVERVIEW**



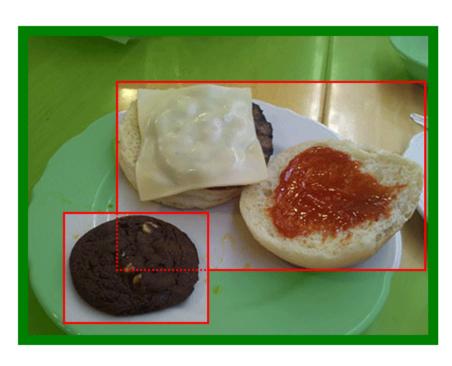
# **OVERVIEW**

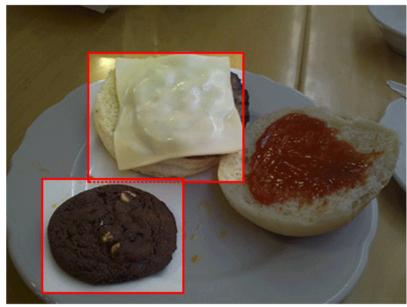


# TAG DISTINCT FOODS ON THE PLATE

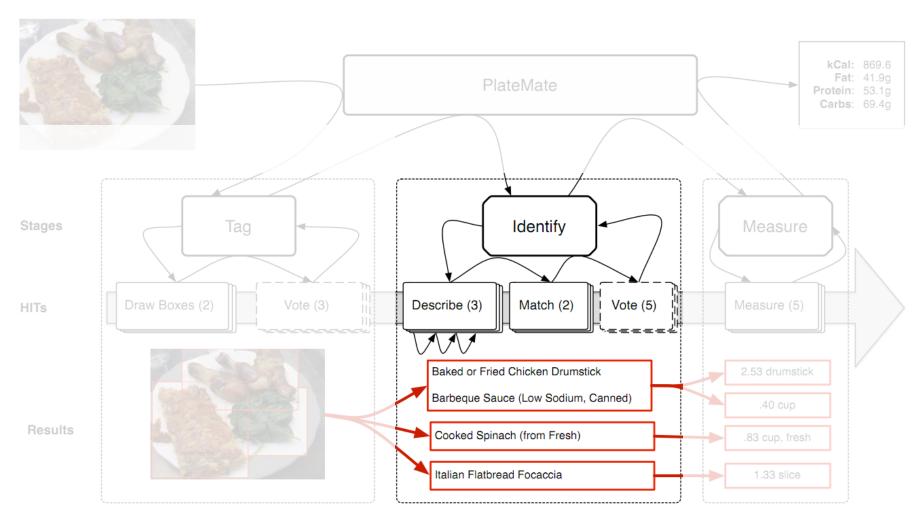


# TAG DISTINCT FOODS ON THE PLATE





# **IDENTIFY EACH ITEM**



# DESCRIBE FOOD AND COMPONENTS ITERATIVELY



**Step 1: Look at the picture.** Decide what food is in the solid red box. Ignore foods that belong to other boxes, which are marked with dashed lines.

**Step 2: What is this food?** What do people call it? Name the food as a whole, even if it contains many parts. Example: "turkey sandwich", "lettuce and tomato"

salad with chicken

#### Step 3: What is it made of? Is it as a combination of other foods?

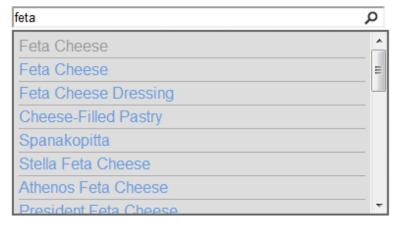
arugula shredded chicken olives tomato onion feta cheese **Step 1: Look at the picture.** Decide what food is in the solid red box. Ignore foods that belong to other boxes, which are marked with dashed lines.

#### Step 2: Read the food description below.

This was written by earlier Turkers looking at the same picture. It could be wrong or incomplete, but it might help you identify the food.

# What this is salad with chicken What it's made of arugula shredded chicken olives tomato onion feta cheese

#### Step 3: Find matching foods in the database.



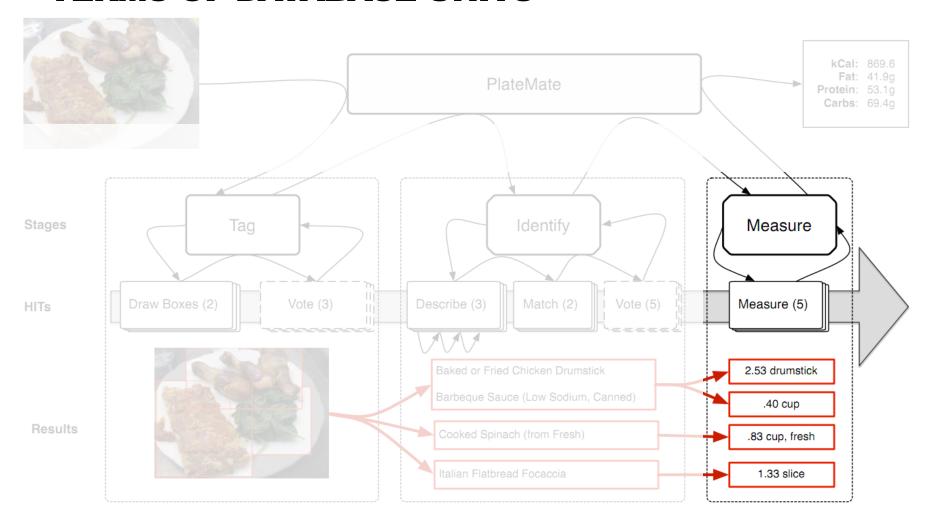
# MATCH DESCRIPTIONS TO NUTRITION DATABASE



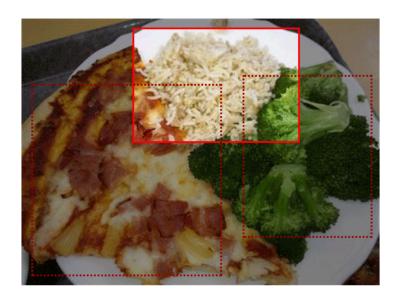
Click on an item to remove it.

Arugula Lettuce	٨
Chicken (Skin Not Eaten)	
Olives	
Feta Cheese	
	-

# MEASURE PORTIONS IN TERMS OF DATABASE UNITS



# MEASURE PORTIONS IN TERMS OF DATABASE UNITS



#### PORTION GUIDE

1 teaspoon (tsp)	single dice, finger tip
1 tablespoon (Tbsp)	thumb tip
2 tablespoons (Tbsp)	ping pong ball or shot glass
1/4 cup	large egg or golf ball
1/2 cup	tennis ball, computer mouse
1 cup	baseball or apple

#### Step 1

Look at the **White Rice** in the red box

#### Step 2

Choose the best measurement option

1 cup cooked

#### Step 3

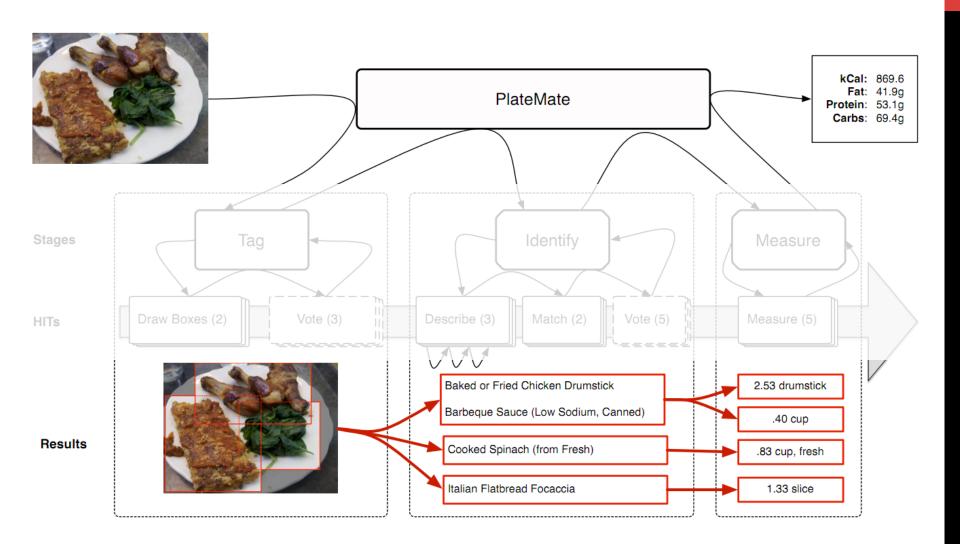
Estimate the portion

.75 x 1 cup cooked = 153 calories.

#### Step 4

Double check your answer

# **EXAMPLE**



## **FRAMEWORK**

- Simulating expertise in crowd of amateurs
- Prior work on "programming crowds"
  - Our paradigm: organizational structure
- "Managing crowds"
  - Division of labor
  - Managers assign, evaluate, and route work
  - Employees are Turkers or other managers

# **EVALUATION**

# **GROUND TRUTH STUDY**

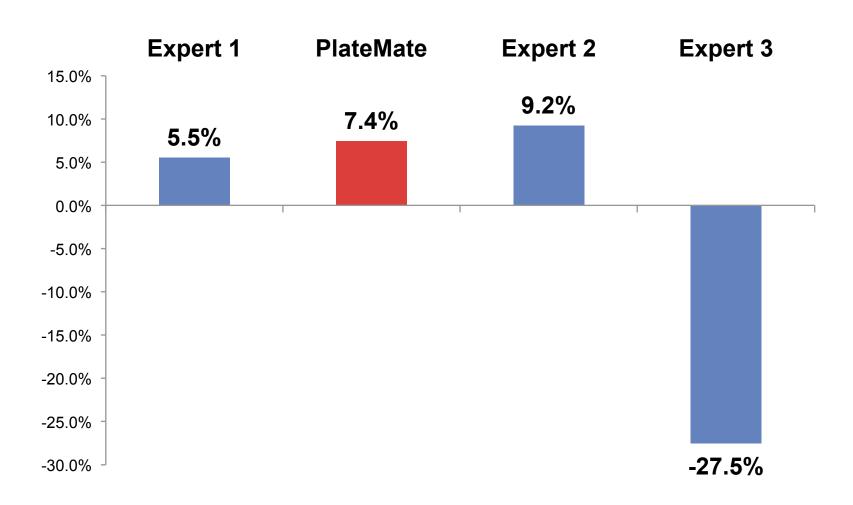
- 36 distinct foods
  - Plates, bowls, packages
  - Restaurant, cafeteria, grocery
  - Weighed as needed
- 3 trained experts



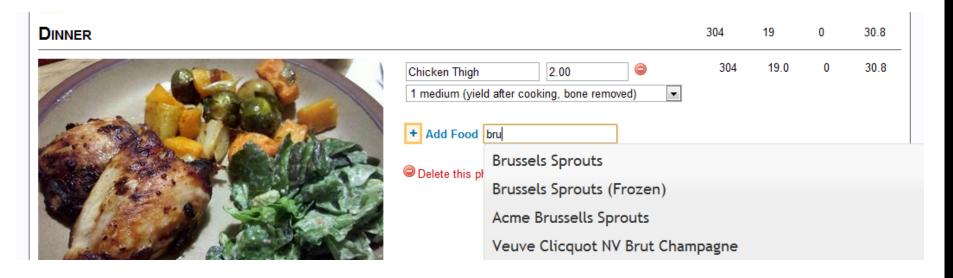




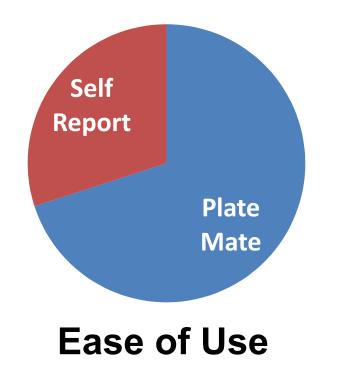
## **GROUND TRUTH STUDY**

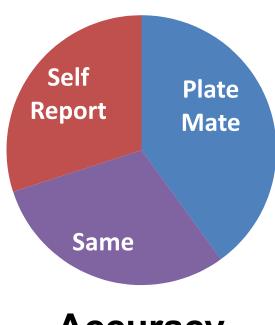


- 10 participants (6 female)
- 2 days receiving PlateMate estimates
- 2 days manually logging with photo
- Before/after interviews on habits, preferences



- 7/10 users preferred PlateMate overall
  - Emphasized convenience, accuracy, ease of correction
- 3/10 preferred manual logging
  - Easier to do it themselves, couldn't trust results





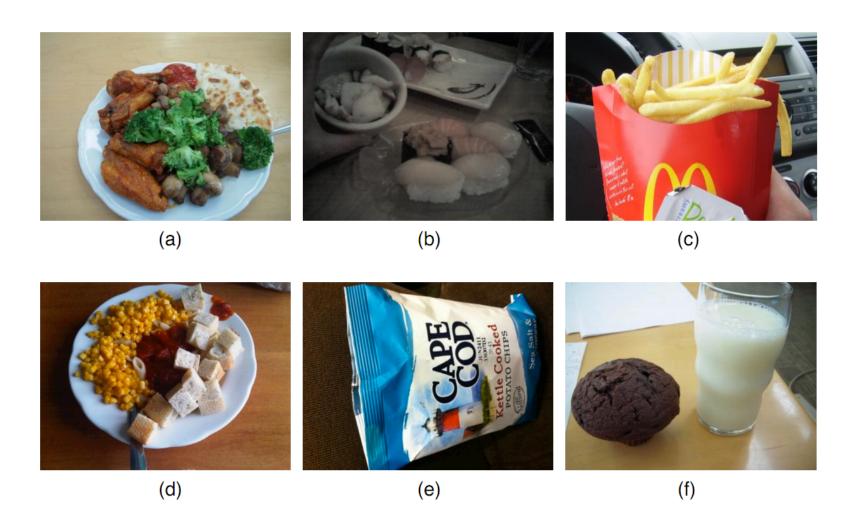
**Accuracy** 

"It was really helpful to have someone else do that for me rather than guess myself"

Prior attempts to record eating were "annoying," "tedious," and "inconvenient."

The evaluation
"confirmed my
suspicions that you
guys were more
accurate than I was. The
tendency is always to
say, 'oh, I didn't have
that much."

# **EASY AND HARD PHOTOS**



## **FUTURE WORK**

- Improve accuracy, lower costs
  - Geolocation
  - Personalization
  - Vision / learning
- Text/voice annotation
- Applying framework elsewhere

# **CONTRIBUTIONS**

- 1. PlateMate, an end-to-end system for crowdsourced nutrition analysis from food photographs
- 2. An evaluation demonstrating PlateMate's accuracy, usability, and robustness
- 3. The Management Framework for solving complex problems with untrained crowds

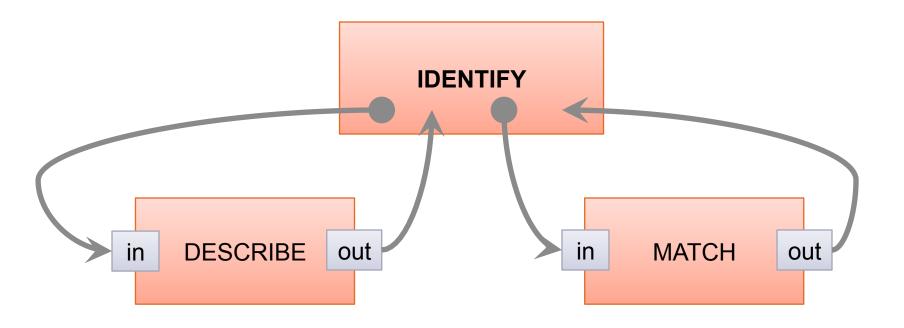
# **QUESTIONS?**

hysen @seas.harvard.edu hqz kgajos

# **BACKUP SLIDES**

# **FRAMEWORK**

- Hierarchy of virtual "managers"
  - Assign work to employees
  - Combine and verify results



- Experts saw no significant accuracy difference
- 49% of estimates within
   100 calories of each other
- PlateMate larger 63% of time
  - PM overestimates slightly
  - Amateurs heavily underestimate
- Daily intake +229 calories higher according to PlateMate



# MATCH DESCRIBED ITEMS TO NUTRITION DATABASE



**Step 1: Look at the picture.** Decide what food is in the solid red box. Ignore foods that belong to other boxes, which are marked with dashed lines.

Step 2: Pick the best option below. Other Turkers selected these foods to match the solid red box in the photo. Think about how well each food or list of foods matches the photo. Choose the most accurate option. If many choices are accurate, pick the simplest one that still fully describes the food(s) in the box. Never select a choice with duplicates, or multiple descriptions of the same food, like "brown rice, white rice"

	Arugula Lelluce
•	Olives
	Chicken (Skin Not Eaten)
	Feta Cheese

Arugula Lattuca

Oriental Chicken or Turkey Garden Salad (Chicken and/or Turkey, Lettuce, Fruit, Nuts)

# MANAGEMENT FRAMEWORK

```
def work(self):
    for input in self.assigned:
        self.employee('draw').assign(photo=input.photo)
    for output in self.employee('draw').finished:
        bg1,bg2 = output.box groups.all()
        similarity = BoxGroup.similarity(bg1,bg2)
        # If responses are similar enough, don't bother voting
        if similarity > MIN SIMILARITY:
            self.finish(photo = output.photo, box group = bg1)
        # Otherwise, we need to vote
        else:
            self.employee('vote').assign(photo = output.photo, bo
```