

2018年度 修士論文

Webカメラを用いた人と物体の認識に関する研究

Study on recognition of a person and object using a Web
camera

指導教員 田中 二郎 教授

早稲田大学大学院情報生産システム研究科

情報生産システム工学専攻

インタラクティブプログラミング研究

44161032-1 松田 健

目次

第1章	はじめに	1
第2章	目的とアプローチ	2
2.1.	目的	2
2.2.	アプローチ	3
第3章	システムデザイン	4
3.1.	システム全体の構成	4
3.2.	検索ビューワー	5
3.2.1.	この人の記録を見る	6
3.2.2.	この飲み物の記録を見る	8
3.3.	分析ビューワー	10
3.3.1.	人と本数の集計結果	11
3.3.2.	消費本数の集計結果	13
3.4.	認識部	15
3.5.	Recorder	21
第4章	システムの実装	29
4.1.	システムの開発環境	29
4.2.	フレームワークの構成	30
4.3.	データベースの実装	31
4.4.	検索ビューワーの実装	32
4.5.	分析ビューワーの実装	33
4.6.	認識部の実装	34
4.7.	Recorder の実装	47
第5章	関連研究	52
第6章	まとめ	55
参考文献	57
概要	59

目次

図： 1	本システムの構成とその流れ.....	4
図： 2	検索ビューワーの TOP ページの画面.....	5
図： 3	「この人の記録を見る」の画面.....	6
図： 4	「飲んだ飲料の種類」へ飲料を何本飲んだかと日時を表示.....	7
図： 5	「この飲料の記録を見る」の画面.....	8
図： 6	「顔を選択」へ誰が何本のんだかと日時を表示.....	9
図： 7	分析ビューワーの TOP ページの画面.....	10
図： 8	分析ビューワーの「人と消費本数の集計結果」.....	11
図： 9	分析ビューワーの「人と飲料の種類を集計結果」.....	12
図： 10	分析ビューワーの「全体の消費本数の記録」.....	13
図： 11	分析ビューワーの「人気の飲料」.....	14
図： 12	認識部の流れ.....	15
図： 13	顔の認識方法.....	16
図： 14	飲料の認識.....	17
図： 15	MICROSOFT FACE API FACEDETECTION による 20 枚の画像から FACEID を 1 個発行....	18
図： 16	MICROSOFT FACE API FINDSIMILAR による 1 個の FACEID から顔を認識.....	19
図： 17	MICROSOFT CUSTOM VISION API による顔画像から飲料画像を取得.....	20
図： 18	iSPY トップページ 破線内は歯車のマーク.....	21
図： 19	画面(図：16)の歯車マークをクリックすると遷移される画面.....	22
図： 20	画面(図：17)内の RECORDING タブ.....	22
図： 21	画面(図：18)から RECORDING をクリックすると遷移される画面.....	23
図： 22	iSPY の撮影前の待機画面(図左) 録画(図中央) 録画終了(図右).....	24
図： 23	FREE VIDEO TO JPG CONVERTER の画面.....	25
図： 24	フリッカーのトップページ.....	26
図： 25	アップロード画面.....	27
図： 26	画像群をアップロードする画面.....	28
図： 27	アップロード完了後に遷移される画面.....	28
図： 28	RAILS フレームワーク(MVC のデータの受け渡し).....	30
図： 29	IMAGE モデル.....	31
図： 30	マイグレーションファイルのデザイン.....	31
図： 31	フレームワーク RAILS の VIEW に書かれている顔と飲料の表示の処理.....	32
図： 32	フレームワーク RAILS の VIEW に書かれている顔と飲料の表示と集計結果の処理..	33
図： 33	フレームワーク RAILS の CONTROLLER に書かれている更新メソッドの処理.....	34
図： 34	CUSTOM VISION API の編集画面.....	41
図： 35	人と物体が写っている画像と物体が大きく写った画像のアップロード画面.....	42
図： 36	アップロードした画像に飲料の名前を記入.....	43
図： 37	設定画面 (PROJECT-ID と ITERATIONID の取得).....	44
図： 38	RECORDER の WEB カメラから N 個の画像群を生成する流れ.....	47
図： 39	iSPY トップページ 破線内は歯車のマーク.....	48
図： 40	画面(図：37)の歯車マークをクリックすると遷移される画面.....	49
図： 41	画面(図：38)内の RECORDING タブ.....	49
図： 42	画面(図：39)から RECORDING をクリックすると遷移される画面.....	50

Contents

図 : 43	FREE VIDEO TO JPG CONVERTER の画面	51
--------	---------------------------------------	----

表目次

表 1	IMAGE テーブルのデザイン	31
表 2	FLICKR API の REQUEST BODY	35
表 3	MICROSOFT FACE API の REQUEST PARAMETERS	36
表 4	MICROSOFT FACE API の RESPONSE BODY	36
表 5	CREATE FACE LIST	37
表 6	REQUEST URL	38
表 7	RESPONSE BODY	38
表 8	REQUEST URL	39
表 9	REQUEST BODY	39
表 10	REQUEST PARAMETERS	40
表 11	RESPONSE BODY	40
表 12	REQUEST PARAMETERS	45
表 13	RESPONSE BODY	46
表 14	RESPONSE BODY 飲料リスト	46

第1章 はじめに

ライフログは人の日々の生活全て、またそれに伴う一部をセンサーやカメラを用いて、数値型情報・文字列型情報・画像情報として記録し続けることである。ライフログの研究には次の(1)-(3)の目的がある。

- (1) 過去の出来事の記憶補助[1][2][3][4]
- (2) 記録の分析[5]
- (3) ライフスタイル変化の通知[6]

ライフログの伝統的な方法として、センサーを内蔵したライフログカメラ(SenceCam等)を用いて、一人称視点の画像を取得する方法がある。一人称視点の画像とは、撮影者の向いた方向の画像が見える。一人称視点の特徴は、同じ人が見る画像を記録し続ける。不利な点は、撮影者自身を記録できないことである。一人称の画像を認識する研究は多くある[4][7][8]。一人称視点の動画を認識する研究について、Wolf[7]らは、人の顔を認識する研究を行った。1個の動画に複数の人を含む場合、全員を認識することが難しいと報告されている。画像を1枚ずつ認識する方法について、Sun[4]らは食事の皿を認識する研究を行った。正確度は9割を超えていると報告されている。

もう一つの方法は、固定化されたカメラ(WEBカメラやFOSCAM等)を用いて、三人称視点の画像を取得することである。三人称視点の画像とは、少し離れた位置から見下ろす視点で画像が見える。三人称視点の特徴は、画像に人と物体を撮影する点である。不利な点は、カメラが固定されているため同じ場所しか撮影できないことである。利用目的は、監視カメラのような防犯が多い。近年では、スーパーにて来店から購入までの間に、客がどんな商品に興味を持っているのかを撮影することで在庫数へ活用する例がある[URL: <https://netshop.impress.co.jp/node/5492>]。三人称視点の画像を認識する研究は多くある[9][10][11][12][13][14]。一例として、動画から画像へ変換し、画像に含む人を認識する比較的新しい試みがある。西崎[9]らは、定点カメラから取得した動画を300-600枚の画像へ分割し、顔の表情を抽出するシステムを提案している。

一人称視点の画像の不利な点は、撮影者自身を記録できないことである。三人称視点の画像の不利な点は、同じ場所しか撮影できないことである。本研究では三人称視点の画像を認識するため、固定カメラからさまざまな知見を得ることに注目する。

第2章 目的とアプローチ

2.1. 目的

本研究の目的は、固定カメラからわかる「さまざまな知見」を「自動的に」得ることである。

そのために、固定カメラに人と物体を含めた画像を自動的に撮影させる。次に、画像に含む人は誰か、物体は何かを認識する。

最後に、人と特定の物体を含む画像を、ビューワーへ自動的に表示する。ビューワーは2つからなる。1つは、「人と飲料に関する記録」を表示させるビューワーであり、人に関する記録と飲料に関する記録からなる。2つ目は「人と飲料に関する集計結果」を表示させるビューワーであり、人に関する「集計結果」と飲料に関する「集計結果」を表示する。

2.2. アプローチ

目的達成のための具体的事例として、研究室の冷蔵庫と人を撮る位置に Web カメラを設置する。冷蔵庫から取り出されるペットボトルの飲料と人の顔を撮影することで、誰が何の飲料を取ったかを判定する。

冷蔵庫を選択した理由は2つある。1つは冷蔵庫の前は研究室の中のため、撮影する人数が限定的な点である。2つ目は冷蔵庫にある飲料の本数が限られている点である。

そのため、本研究における最初のステップとして冷蔵庫の前の飲料を取り出す人を撮影する。

次に、我々は本システムを3部(Recorder部, 認識部, Viewer部)で構成する。

Recorder部とは、Web カメラから iSpy によって、冷蔵庫の前に立つ人の飲料を取り出す場面が撮影される。iSpy とは人の動作を検知して動画を撮影するソフトウェアである。Web カメラから動画を取得し、各動画を画像 20 枚の画像群へ変換し、認識部へ渡す。

認識部とは、Recorder部から画像群を取得する。各画像群から顔と飲料を含む画像 1 枚を抽出し、画像を Viewer部へ渡す。

Viewer部とは、人に関する記録と飲料に関する記録を閲覧する検索ビューワーと、人と飲料に関する集計結果を閲覧する分析ビューワーからなる。

第3章 システムデザイン

3.1. システム全体の構成

我々は図:1 に示すように本システムを構築する.

(1) Recorder

Web カメラから iSpy によって, 冷蔵庫の前に立つ人の飲料を取り出す場面が n 回撮影される. iSpy とは人の動作を検知して動画を撮影するソフトウェアである. Recorder は Web カメラから n 個の動画を取得し, 各動画を画像 20 枚の画像群へ変換し, 認識部へ渡す.

(2) 認識部

Recorder から n 個の画像群を取得する. 各画像群から顔と飲料を含む画像 1 枚を抽出し, 画像 n 枚を検索ビューワーと分析ビューワーへ渡す.

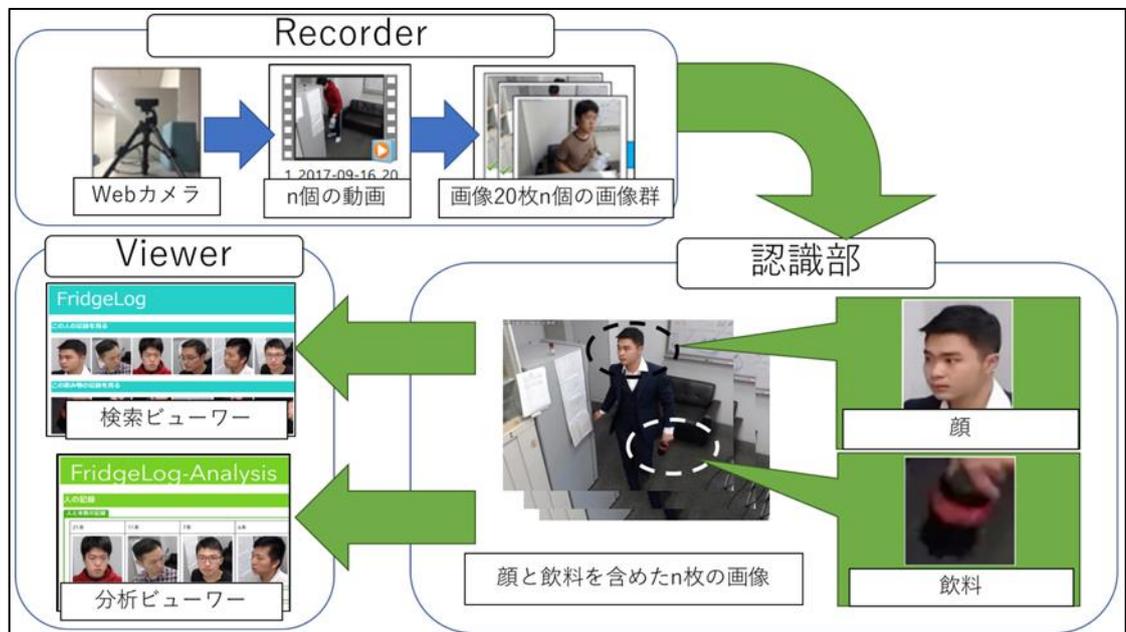
(3) Viewer

I. 検索ビューワー

認識部から顔と飲料を含む画像 n 枚を取得する.
人に関する記録, 飲料に関する記録を出力する.

II. 分析ビューワー

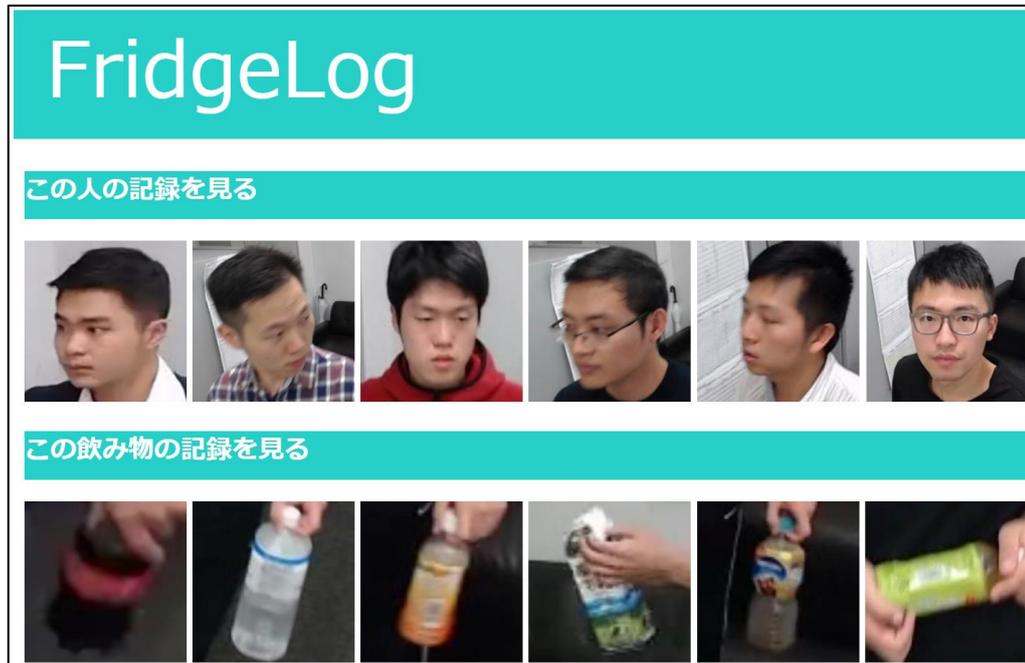
認識部から顔と飲料を含めた画像 n 枚を取得する.
人と飲料に関する集計結果を出力する.



図：1 本システムの構成とその流れ

3.2. 検索ビューワー

検索ビューワーは図:2 に示すように「この人の記録を見る」と、「この飲み物の記録を見る」からなる。「この人の記録を見る」には顔が表示され、「この飲み物の記録を見る」には飲料が表示される。



図：2 検索ビューワーの TOP ページの画面

3.2.1. この人の記録を見る

検索ビューワーの TOP ページ画面(図:2)の「この人の記録を見る」から顔を1つ選んでクリックすると、画面(図:3)へ遷移する。

画面に、クリックされた顔が表示され(他の顔は無くなる),「飲んだ飲料の種類」の欄には、その人が飲んだ飲料が表示される。



図：3 「この人の記録を見る」の画面

飲んだ飲料の表示

「この人の記録を見る」の画面(図:3)の「飲んだ飲料の種類」の欄から、飲料をクリックする。するとクリックされた飲料の下一列へ、その飲料を取り出した場面といつ飲まれたか日時が表示される(図:4 破線内).



図：4 「飲んだ飲料の種類」へ飲料を何本飲んだかと日時を表示

3.2.2. この飲み物の記録を見る

検索ビューワーの TOP ページ画面(図:2)の「この飲み物の記録を見る」から飲料を1つ選んでクリックすると、画面(図:5)へ遷移する。

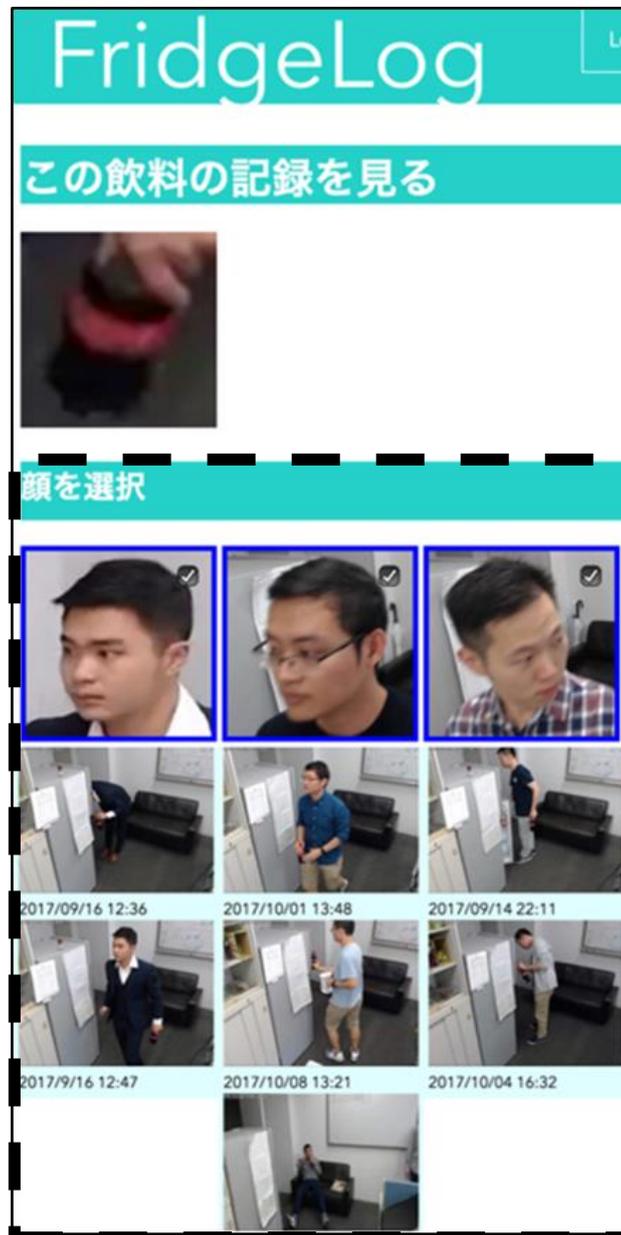
画面に、クリックされた飲料が表示され(他の飲料は無くなる)、「顔を選択」の欄には、その飲料を飲んだ人が表示される。



図：5 「この飲料の記録を見る」の画面

誰が何本飲んだか表示

「この飲料の記録を見る」の画面(図:5)の「顔を選択」にて、顔をクリックする。するとクリックされた顔の人の下一列へ、その顔の人が飲料を取り出した過去の場面といつ飲んだ記録が表示される(図:6 破線内).



図：6 「顔を選択」へ誰が何本のんだかと日時を表示

3.3. 分析ビューワー

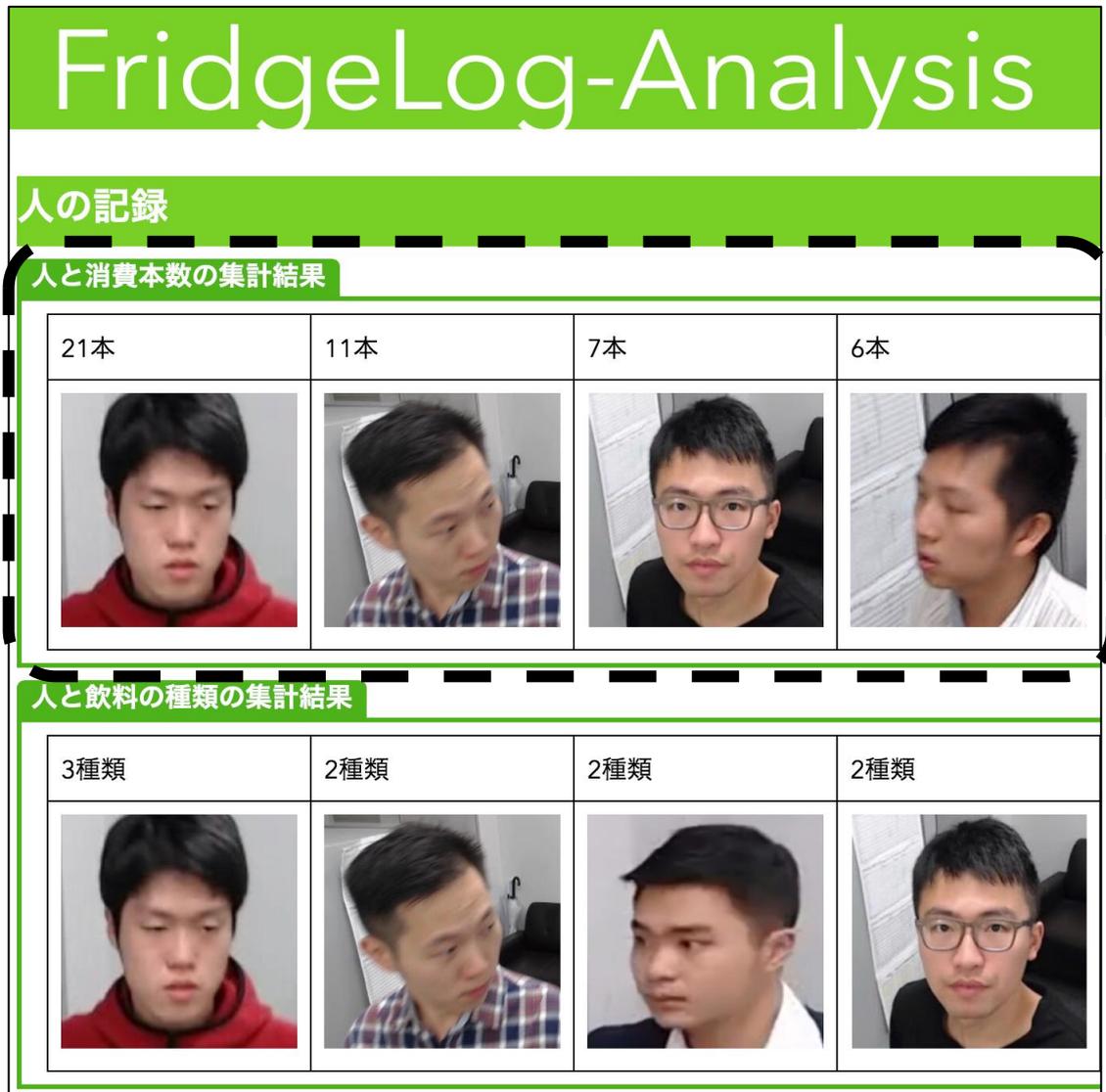
分析ビューワーは図:7に示すように「人の記録(図:7 上破線内)」と、「飲料の記録(図:7 下破線内)」からなる。



図: 7 分析ビューワーの TOP ページの画面

3.3.1. 人と本数の集計結果

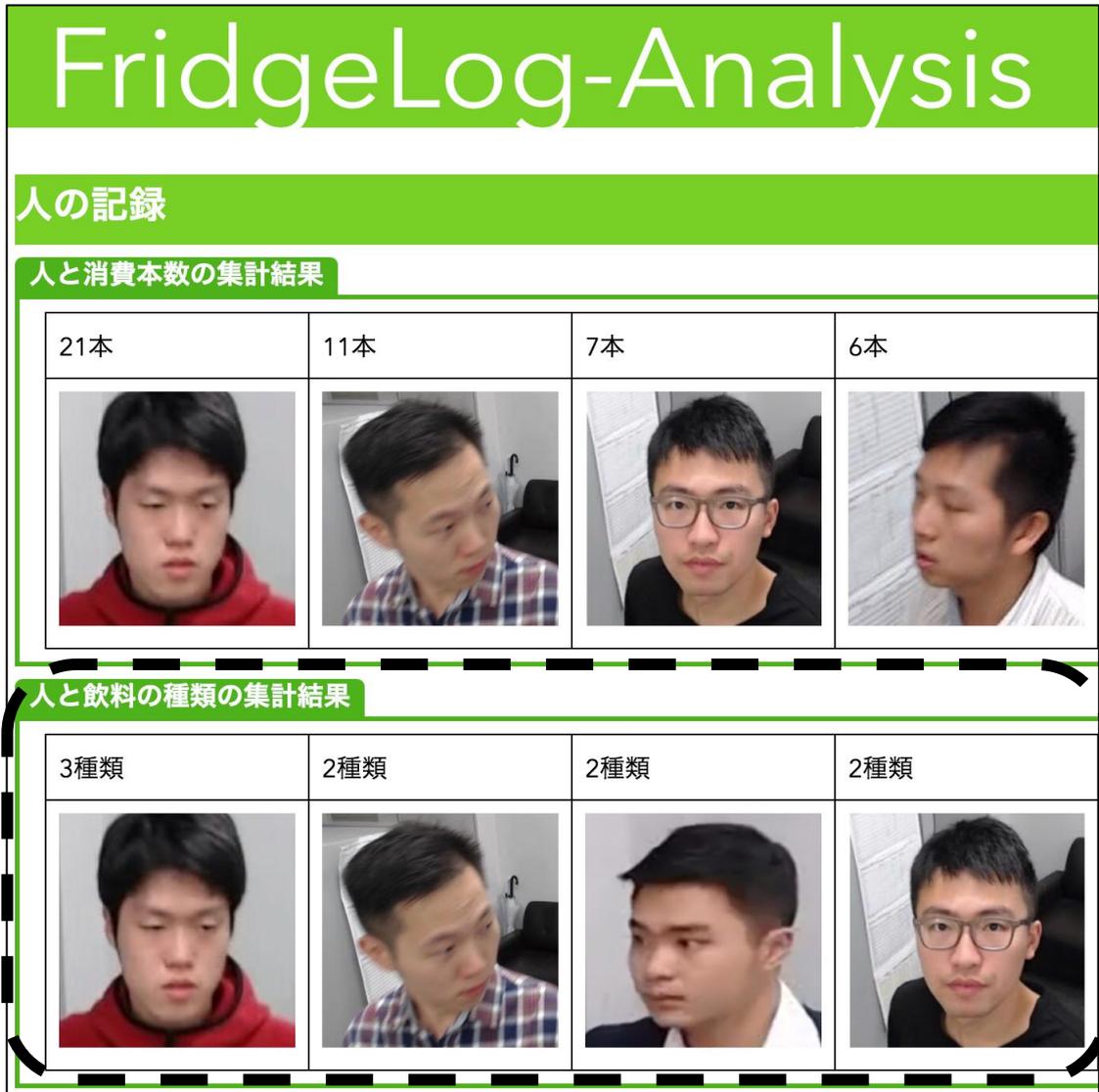
「人と消費本数の集計結果(図:8 破線内)」の欄には、その人が飲んだ飲料の本数の集計結果が表示される。



図：8 分析ビューワーの「人と消費本数の集計結果」

人と種類の集計結果

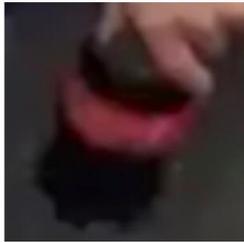
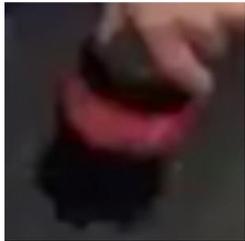
「人と飲料の種類を集計結果(図: 9 破線内)」の欄には、その人が何種類の飲料を飲んだか集計結果が表示される。



図：9 分析ビューワの「人と飲料の種類を集計結果」

3.3.2. 消費本数の集計結果

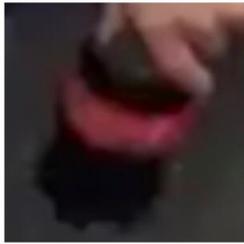
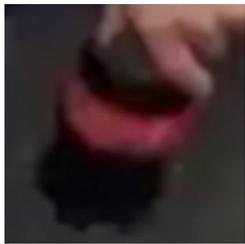
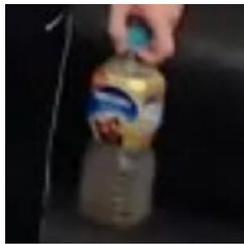
「全体の消費本数の記録(図:10 破線内)」の欄には、飲料が消費された本数の集計結果が表示される。

飲料の記録			
全体の消費本数の記録			
12本	10本	6本	6本
			
人気の飲料			
5人	2人	2人	2人
			

図：10 分析ビューワーの「全体の消費本数の記録」

人気の飲料の集計結果

「人気の記録(図:11 破線内)」の欄には、飲料を取り出した人数が表示される。

飲料の記録			
全体の消費本数の記録			
12本	10本	6本	6本
			
人気の飲料			
5人	2人	2人	2人
			

図： 11 分析ビューワーの「人気の飲料」

3.4. 認識部

我々は図:12 に示すように認識部を構築する.

Recorder の n 個の画像群から画像 20 枚ずつ取得する. 20 枚の画像から FaceId がある画像 1 枚を取得する. FaceId とは顔の ID である. FaceId を含む画像 1 枚から顔が既知の顔か未知の顔かを認識する.次に顔のある画像 1 枚から飲料を認識する. 最後に顔と飲料を含む画像 1 枚ずつを取得し, Viewer へ顔と飲料を含む n 枚の画像を渡す.

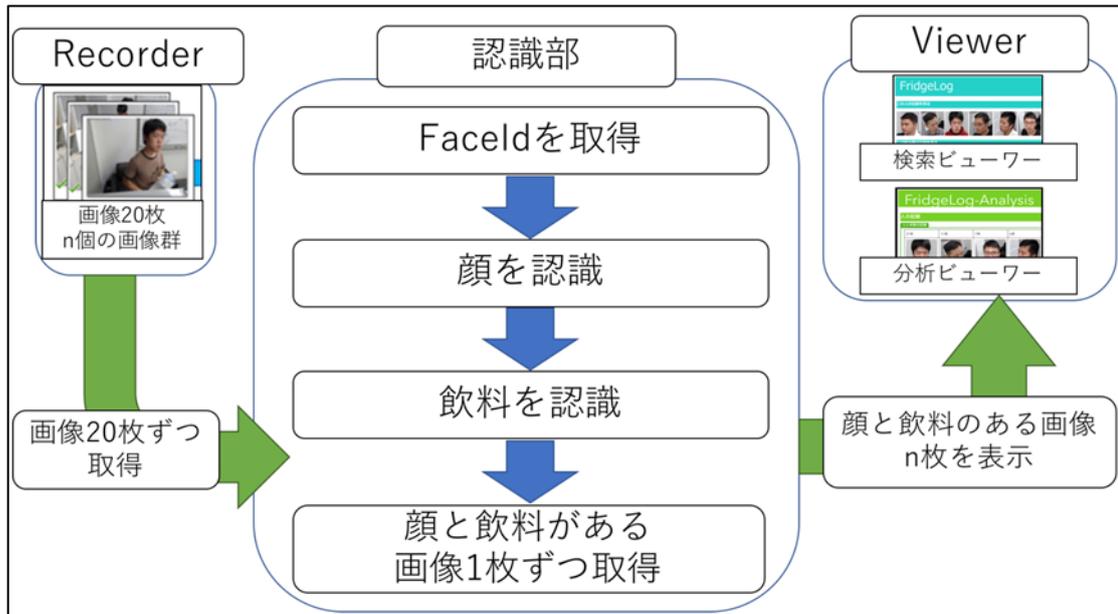


図 : 12 認識部の流れ

顔の認識方法

未知の顔の入力に対して、事前に編集した顔リストの一意的な顔とどのくらい似ているのか信頼度を取得する。信頼度が低い場合、未知の顔は新規の顔と認識する。信頼度が高い場合、既存の顔と認識する。

顔リストの構築

使用する API

顔リストの構築には Microsoft FaceList API を使用する。FaceListAPI とは、顔認識に使用する顔リストを構築する。

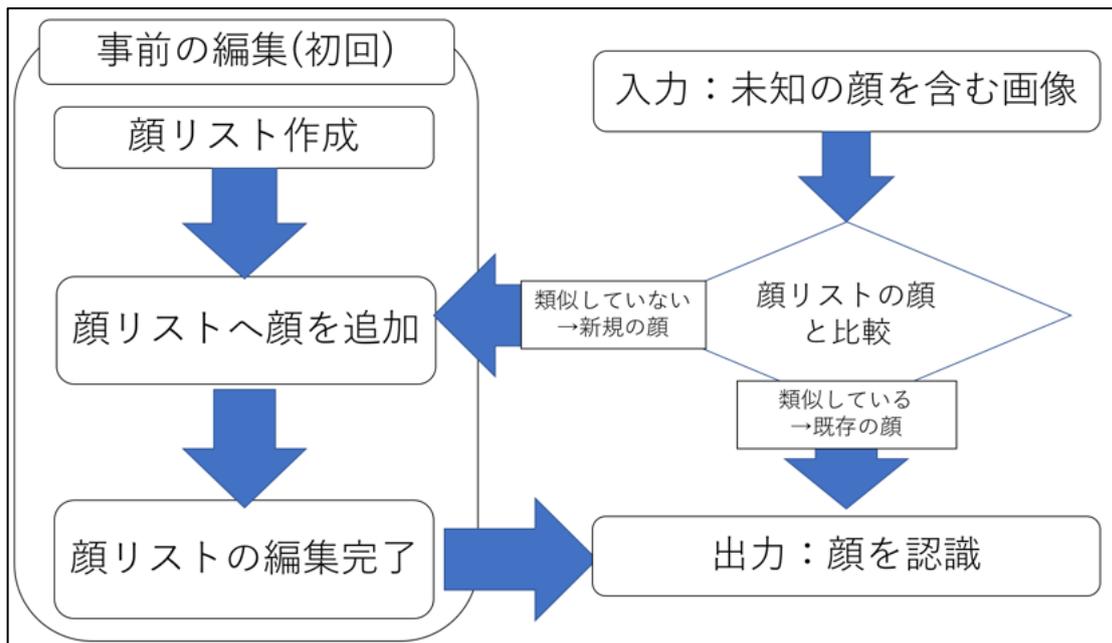
事前の編集(初回)

FaceListAPI を用いて空の顔リストを作る。顔リストへ顔を 1 個追加する。顔リストの編集を完了する。

顔リストの利用

顔の認識と、顔リスト利用の流れを図：13 に示す。

顔リストへ未知の顔を入力する。リストの顔とどのくらい類似しているか信頼度を取得する。信頼度が低い顔は新規の顔として認識される。新規の顔は、FaceListAPI を用いて顔リストへ追加される。1 個の画像群が n 回入力され、顔リストの信頼度を n 回取得する。すなわち、 n 人の顔が新規の顔か既存の顔か特定する。



図：13 顔の認識方法

飲料の認識方法

飲料の認識は顔の認識の後に行う。

顔を含む画像の1枚の入力に対して、事前に編集した飲料リストの一意的な飲料とどのくらい似ているのか信頼度を取得する。信頼度が高い場合、飲料を認識する。信頼度が低い場合、飲料はないと認識する。

飲料リストの構築

使用する API

飲料リストの構築には Custom Vision API を使用する。Custom Vision API とは、事前に API を編集することで特定のオブジェクトを認識する。

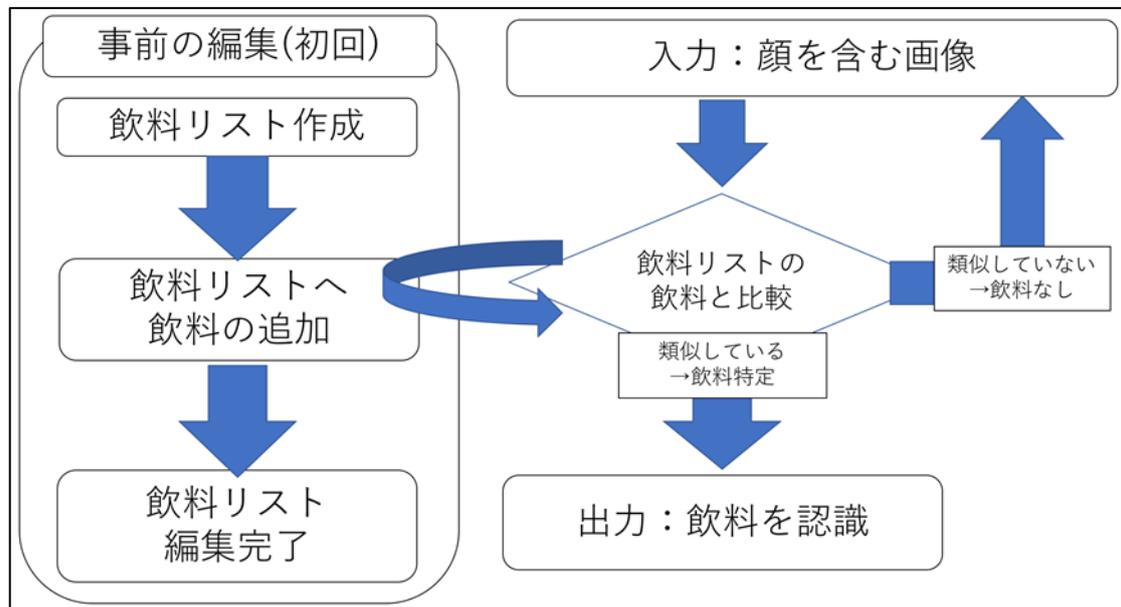
事前の編集(初回)

Microsoft Custom Vision API を用いて、空の飲料リストを生成する。認識したい飲料の画像と飲料名を飲料リストへ追加する。追加する飲料の画像は、飲料全体の映った画像と、飲料を含む人の画像の2種類である。飲料リストの編集を完了する。

飲料リストの利用

飲料の認識と、飲料リスト利用の流れを図：14 に示す。

飲料リストへ顔を含む画像を入力する。リストの飲料とどのくらい類似しているか信頼度を取得する。信頼度が低い顔は飲料が含まれていないと認識される。1個の画像群が n 回入力され、飲料リストの信頼度を n 回取得する。すなわち、n 本の飲料を認識する。



図：14 飲料の認識

FaceId を取得

使用する API

(I) Flickr API を使用する. Flickr API とは, 画像をフリッカーから取得する.

(II) Microsoft Face API の FaceDetection を使用する. Microsoft Face API とは, 画像から顔を認識する. FaceDetection とは, 顔を含む画像に FaceId を発行する.

取得方法

Flickr API を用いて, Microsoft Face API の FaceDetection へ画像 20 枚ずつ入力する. 20 枚の画像から顔がある画像 1 枚へ FaceId を 1 個出力する(図:15).

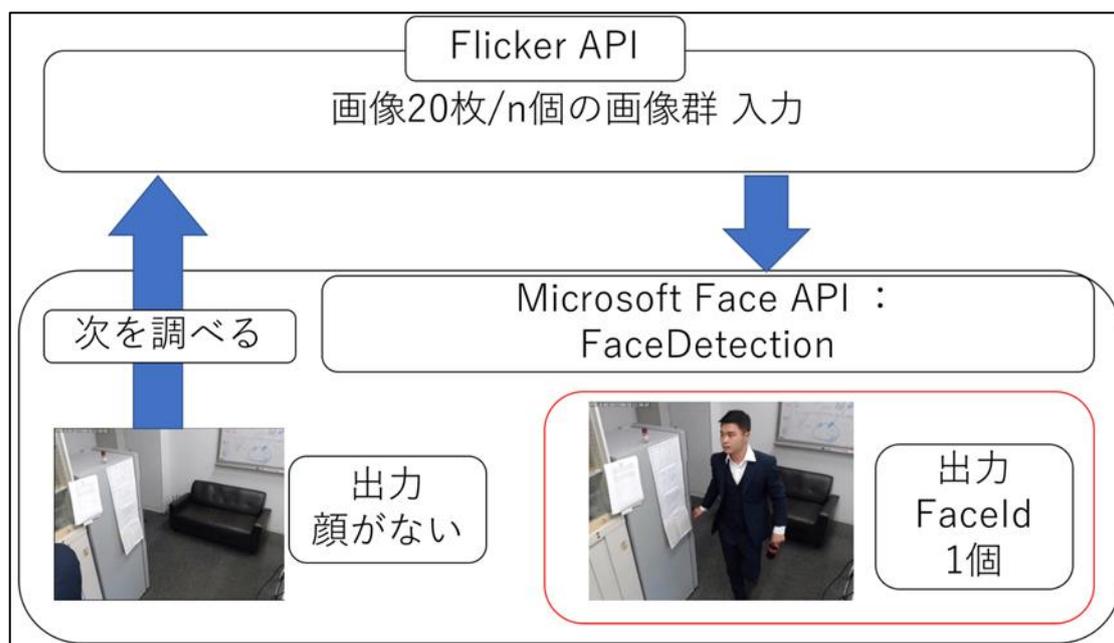


図 : 15 Microsoft Face API FaceDetection による 20 枚の画像から FaceId を 1 個発行

顔を認識

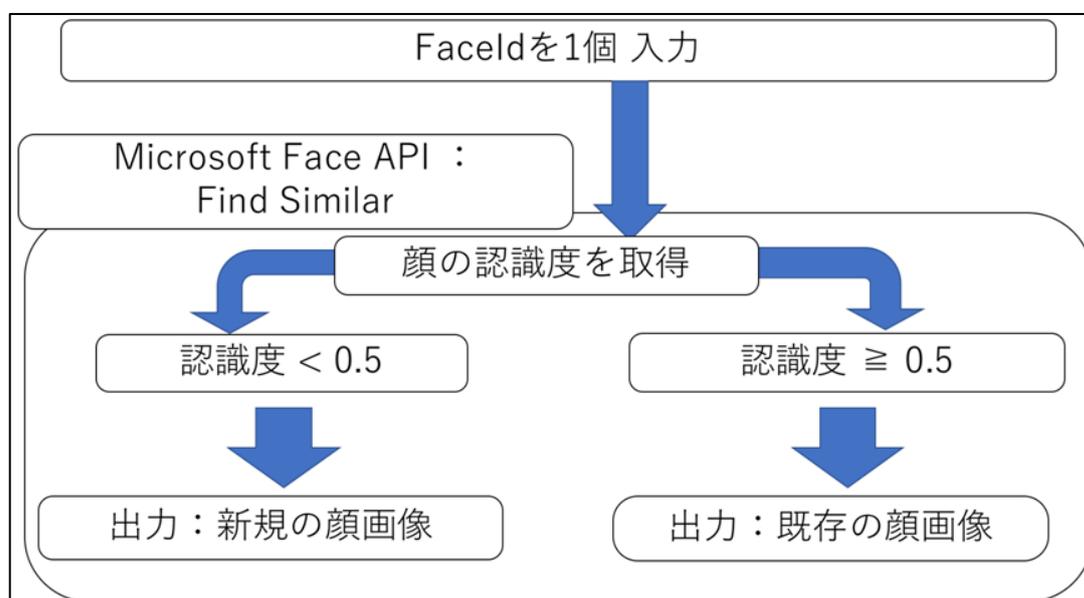
使用する API

Microsoft Face API の Find Similar を使用する。Find Similar とは、未知の顔と既知の顔を比較し、顔がどのくらい似ているのか認識度(0<1. 0)を取得する。

認識方法

Microsoft Face API の Find Similar へ FaceId を 1 個入力する。

FaceId の顔から顔の認識度を取得する。認識度が 50%以上の場合、FaceId の顔を既存の顔と認識する。50%未満(認識度なしも含む)の場合、新規の顔と認識する。1 枚の顔画像(既存の顔か、新規の顔)を出力する(図:16)。



図： 16 Microsoft Face API FindSimilar による 1 個の FaceId から顔を認識

飲料を認識

使用する API

Microsoft Custom Vision API を使用する. Microsoft Custom Vision API とは, 画像に含む未知の物体と既知の物体を比較し, 物体がどのくらい似ているのか信頼度を取得する.

認識方法

Microsoft Custom Vision API へ顔画像を 1 枚入力する.飲料の信頼度を取得する.信頼度が 40%以上の場合, 飲料を認識する.40%未満の場合, 画像に飲料を含まないと認識する. 1 枚の飲料画像を出力する.

飲料の認識を終了すると, Viewer の検索ビューワーと分析ビューワーへ顔と飲料を含む画像を 1 枚ずつ出力する(図:17).

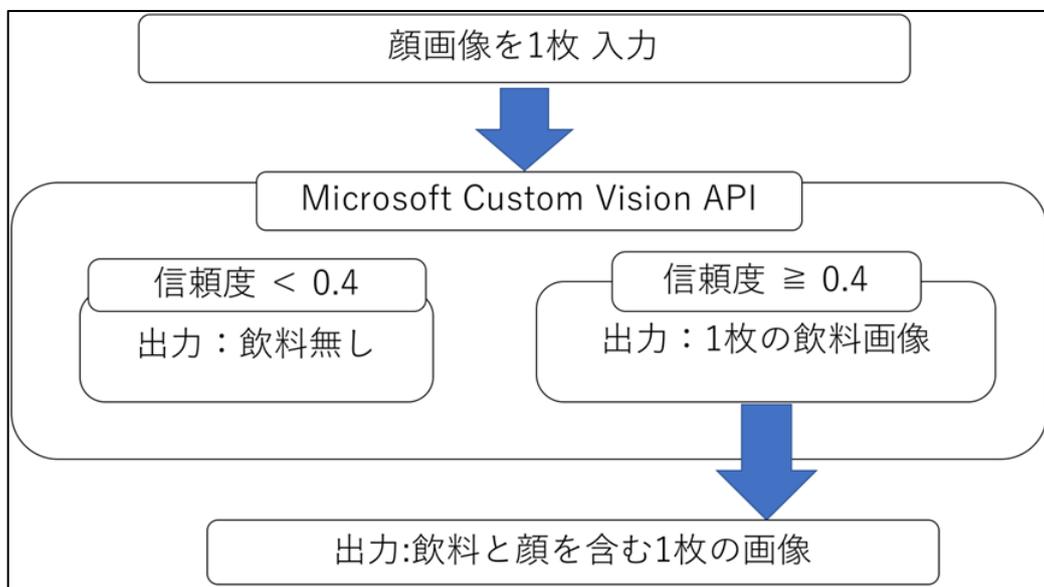


図 : 17 Microsoft Custom Vision API による顔画像から飲料画像を取得

3.5. Recorder

人と飲料を含む動画を撮影

使用するソフトウェア

iSpy を使用する. iSpy とは, 事前に設定を行うことで人の動作を自動的に検知する. 検知をするとすぐ撮影が開始される.

動作設定の手順(1)

Web カメラとパソコンを接続し, iSpy を起動する. パソコン上で iSpy のトップページが起動され, 画面(図: 18)の歯車のアイコンをクリックする(図:18 破線内).

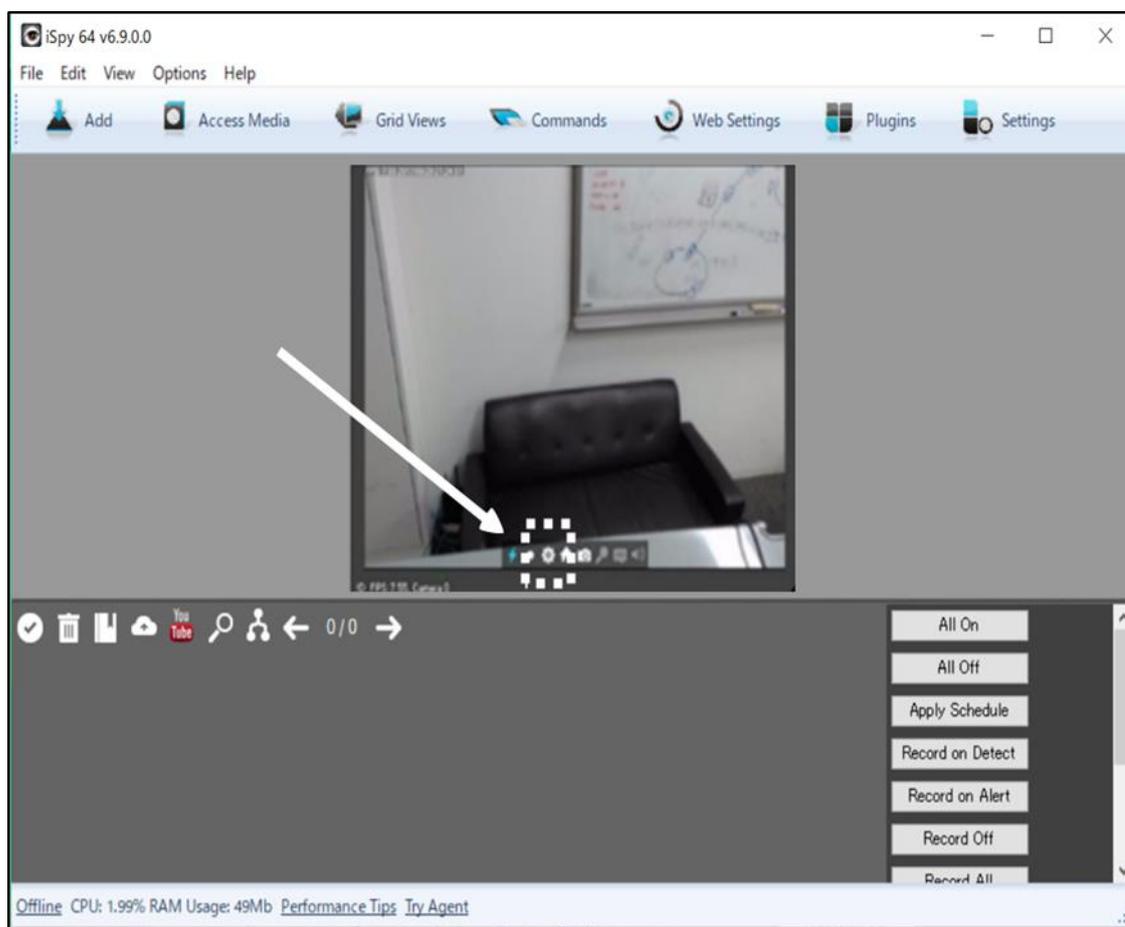
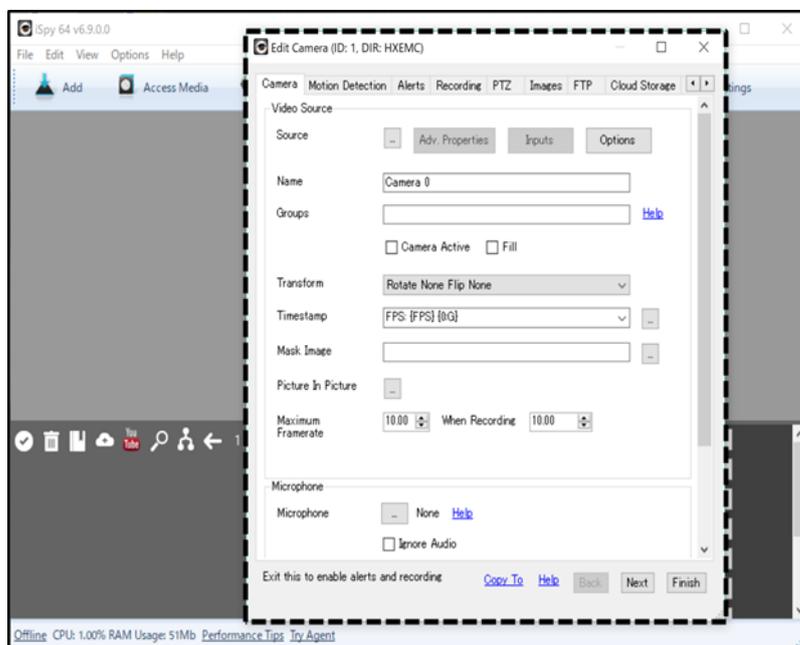


図: 18 iSpy トップページ 破線内は歯車のマーク

動作設定の手順(2)

iSpy トップページの歯車のアイコンをクリックされると画面(図:19 破線内)が表示される。同画面内(図:20 破線内)の Recording タブをクリックする。



図： 19 画面(図：16)の歯車マークをクリックすると遷移される画面



図： 20 画面(図：17)内の Recording タブ

動作設定の手順 (3)

画面(図: 20)から Recording をクリックすると画面(図:21)へ遷移される。

Recording Mode(録画の方法)の欄から, Record on Movement Detection(動作検知)の横にあるラジオボタンにチェックを入れる。

Recording Settings(録画の設定)の欄から, Min Record Time(最小録画時間)の横にある空欄に 5 秒と記入する。また, Max Record Time(最大録画時間)の横にある空欄に 10 秒と設定する。

Section	Setting	Value	Unit
Recording Mode	No Recording	<input type="radio"/>	
	Record on Movement Detection	<input checked="" type="radio"/>	
	Record on Alert	<input type="radio"/>	
Recording Settings	Max. Calibration Delay	15	seconds
	Buffer	0	seconds
	Inactivity Record	0	seconds
	Min Record Time	5	seconds
	Max. Record Time	10	seconds

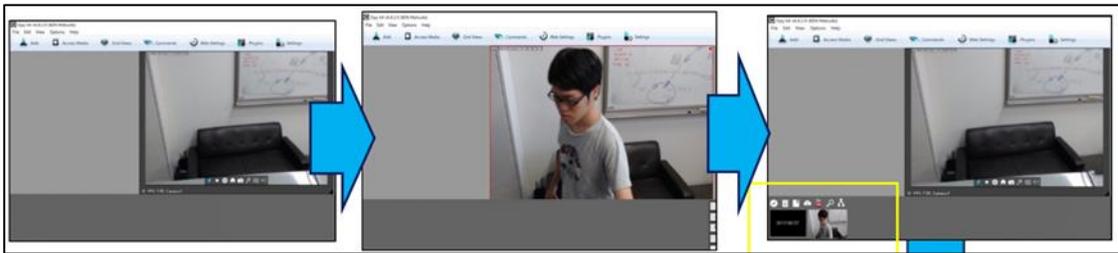
図: 21 画面(図: 18)から Recording をクリックすると遷移される画面

n 個の動画を生成

iSpy を起動すると、画面(図:22 左)にて、録画の待機画面が表示される。

冷蔵庫の前に人が立って飲料を取り出すと iSpy によって録画が自動的に開始される(図:22 中央)。飲料を持ち去ると、録画が終了する(図:22 右)。

録画が終了すると、画面(図:22 右の枠線内)に撮影された直前の動画のサムネイルが表示される。冷蔵庫の前の人の飲料の取り出しを行う動作が n 回であるとすると、n 回撮影され、iSpy から n 個の動画が生成される。



図： 22 iSpy の撮影前の待機画面(図左) 録画(図中央) 録画終了(図右)

画像 20 枚 n 個の画像群を生成

使用するソフトウェア

FreeVideoToJPGConverter を使用する。FreeVideoToJPGConverter とは、1 個の動画を画像へ分割する。

静止面枚数の設定

Free Video to JPG Converter を起動する。画面（図:23 破線内）にて、合計の横にあるラジオボタンにチェックを入れる。「動画内のフレーム」に 20 と記入する。

取得方法

画面(図:23)の「ファイル追加」をクリックする。動画を選択し、入力する。1 動画を 20 枚の画像へ分割する。入力する動画の数が n 個だとすると、画像 20 枚で構成された n 個の画像群を生成する。



図： 23 Free Video to JPG Converter の画面

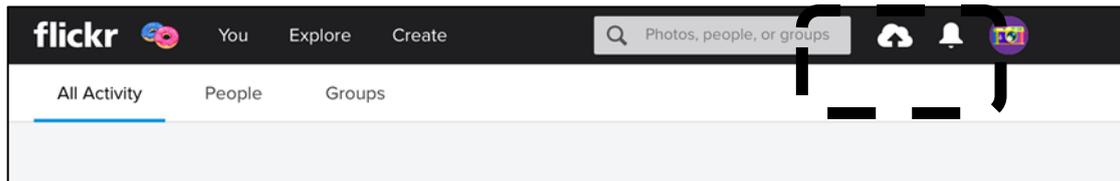
n 個の画像群をフリッカーへ保存

使用する Web サービス

フリッカーを使用する。フリッカーとは画像をサーバーで保存する。

保存手順 (1)

フリッカーのトップページ[Flickr online: <https://www.flickr.com/>]へアクセスする。画面(図:24)に遷移される。アップロードボタン(図:24 破線内)をクリックする。



図： 24 フリッカーのトップページ

保存手順 (2)

画面(図: 24)のアップロードボタンがクリックされると、画面(図:25)に遷移される.n
個の画像群をドラッグ&ドロップでフリッカーへアップロードする(図:25 破線内).

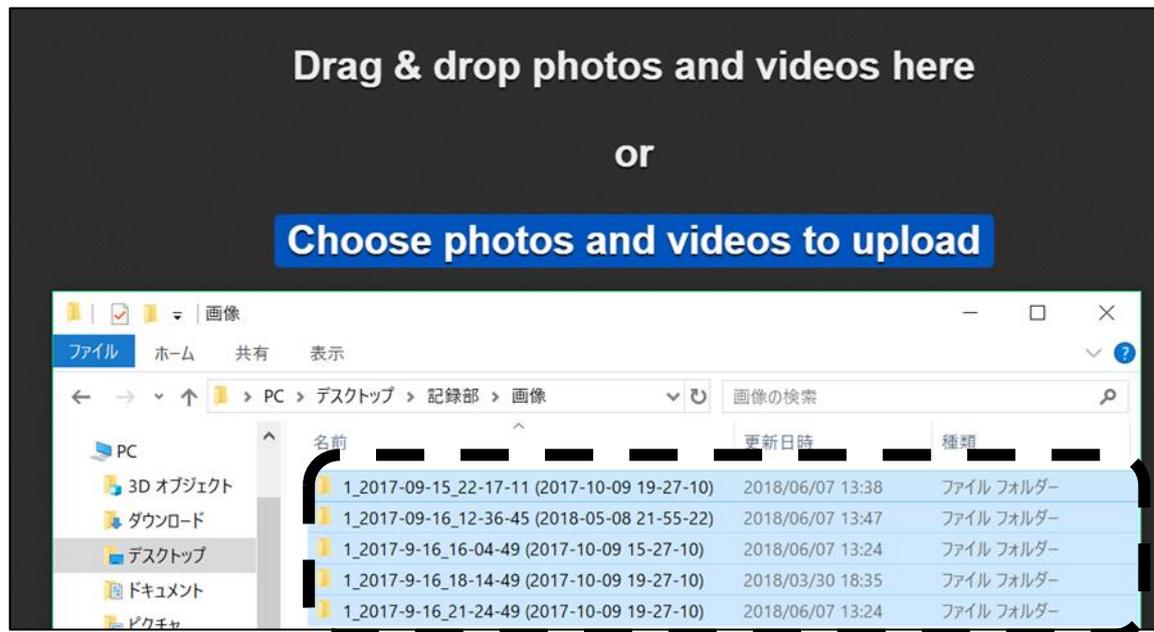
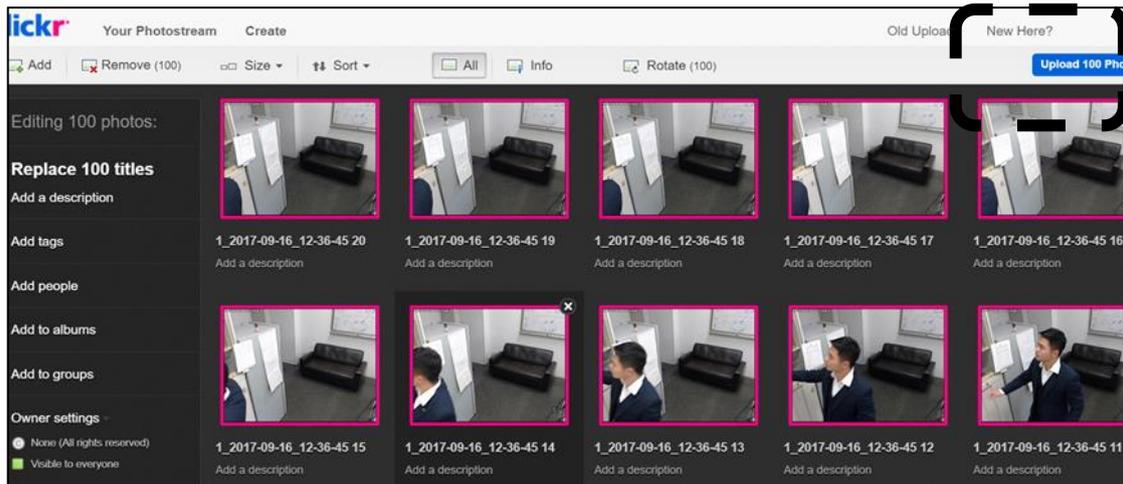


図: 25 アップロード画面

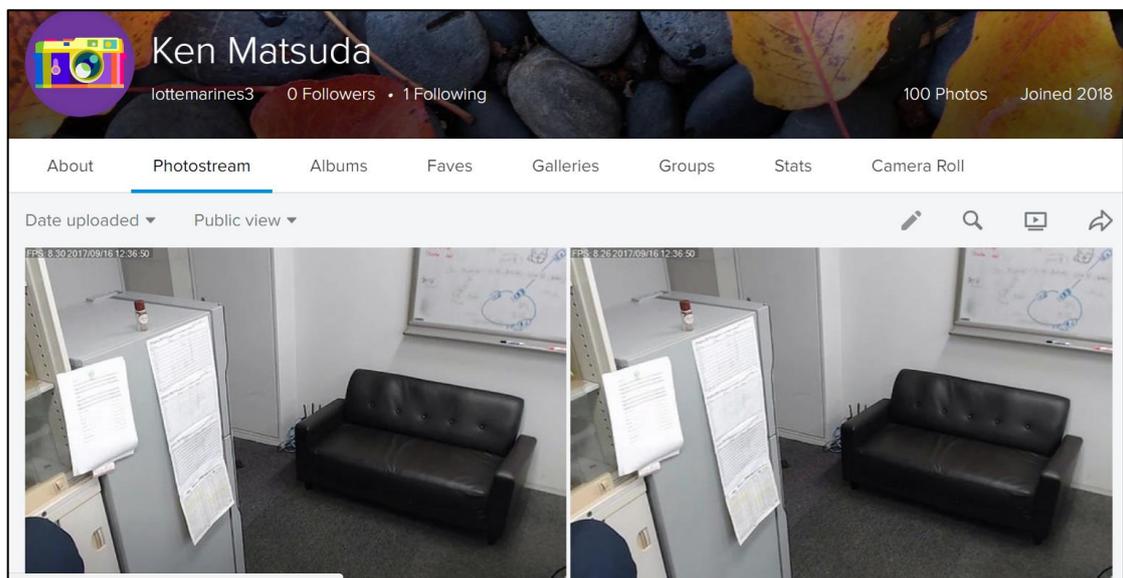
保存手順 (3)

n 個の画像群をアップロードすると、画面(図:26)に遷移する。各画像群から、画像が 20 枚ずつアップロードされている。

画面右上のボタン(図:26 破線内)をクリックする。すると画面(図:27)に遷移する。画像 20 枚で構成された画像群 n 個をアップロードしたとすると、n 個の画像群を保存することを完了する。



図： 26 画像群をアップロードする画面



図： 27 アップロード完了後に遷移される画面

第 4 章 システムの実装

4.1. システムの開発環境

Viewer の開発環境は Ubuntu/Linux 環境で AWS Cloud9[URL1]の IDE ブラウザ開発を用いて、実装ブラウザは Chrome を用いた。プログラミング言語は Ruby を用いた。フレームワークは Rails を使用する。データベース開発には SQLite3 を使用する。

認識部の開発環境は、Ubuntu/Linux 環境で AWS Cloud9 の IDE ブラウザ開発を用いて、実装ブラウザは Chrome を用いた。プログラミング言語は Ruby を用いた。フレームワークは Rails を使用する。データベース開発には SQLite3 を使用する。画像群の取得に Flickr API[URL2]を使用する。顔認識に Microsoft Face API[URL3]を使用し、飲料認識に Microsoft Custom Vision API[URL4]を使用する。

Recorder の開発環境は Windows 環境で、Web カメラの使用には iSpy[URL5]のソフトウェアを使用する。動画を画像群へ変換することに Free Video To JPG Converter[URL6]のソフトウェアを使用する。

※URL 参照

[URL1: Amazon Web Service Cloud9 online: <https://aws.amazon.com/jp/cloud9/>]

[URL2: Flickr API online: <https://www.flickr.com/services/api/>]

[URL3: Microsoft Cognitive Services online: <https://azure.microsoft.com/ja-jp/services/cognitive-services/face/>]

[URL4: Microsoft Cognitive Services online: <https://azure.microsoft.com/ja-jp/services/cognitive-services/custom-vision-service/>]

[URL5: iSpy HP online: <http://www.ispyconnect.com/>]

[URL6: Free Video To JPG Converter DL リンクページ online: <https://free-video-to-jpg-converter.jp.uptodown.com/windows>]

4.2. フレームワークの構成

フレームワークは Rails を利用する。ブラウザから http の入力が行われると、Rails の遷移順 (ブラウザ→View→Root→Controller→Model→Controller→View→ブラウザ) でブラウザへレスポンスが返される(図:26)。誰が何の飲料を飲んだかの画像の表示と更新をする。View にはブラウザへ人と飲料を含む画像を表示する処理が書かれている。View の処理は Html, Scss, Javascript, Ruby で書いた。Controller には 20 枚ずつ画像の取得/人と飲料を含む画像 1 枚の認識/画像の保存の処理が書かれている。Controller の処理は Ruby で書いた。Model にはデータベースのデータ処理が書かれている。Model の処理は Ruby で書いた。Root には検索ビューワーと分析ビューワーのブラウザから http のアクセスに応じてコントローラー内のどのメソッドが応じるかのルート処理が書かれている。Root は Ruby で書いた。

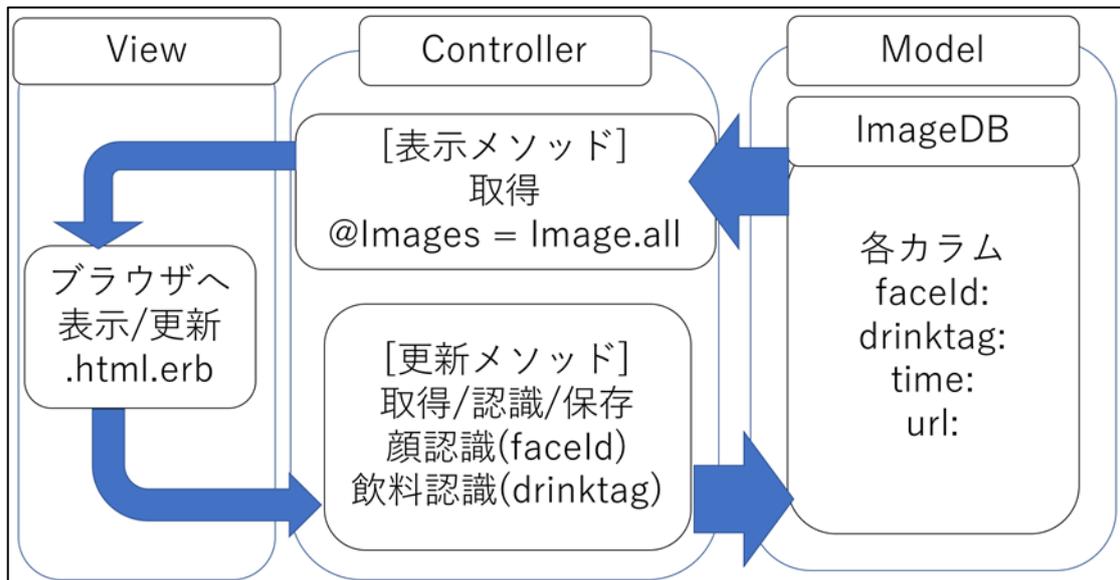


図 : 28 Rails フレームワーク(MVC のデータの受け渡し)

4.3. データベースの実装

データベースは Rails にある SQLite3 を用いて開発する。システムのデータベースは Rails のモデルからテーブルのデータ操作が行われる。テーブル名(表 1), モデル名は Image(図:27), マイグレーションファイルのカラムは画像情報, 時刻情報, 顔情報, 飲料名情報で構成される(図:28).

表 1 Image テーブルのデザイン

カラム名	値	辞書
url	:string	画像情報
time	:string	時刻情報
faceid	:string	顔情報
drinktag	:string	飲料名情報



図 : 29 Image モデル

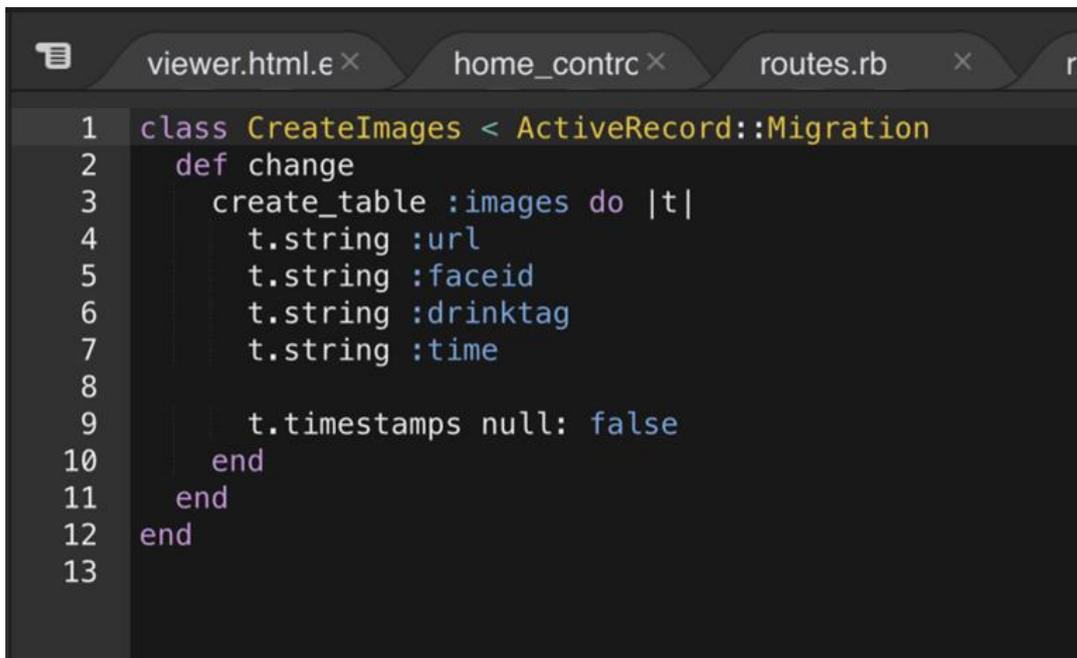


図 : 30 マイグレーションファイルのデザイン

4.4. 検索ビューワーの実装

フレームワークの Rails の View に顔と飲料の表示の処理が書かれている(図:29). トップページ(/.html.erb)には顔の表示と, 飲料の表示をする. 顔の表示はデータベースにある顔 (faceid カラムに値)を持つ画像を 1 枚表示する. 飲料の表示はデータベースにある飲料 (drinktag カラムに値)を持つ画像を 1 枚表示する.

1 人の顔ページ(@Image.faceId.html.erb)はトップページの顔がクリックされると遷移される. 1 人の顔ページ内の飲料の画像を選択する. 選択された飲料が飲まれた場面の画像と, 日時を表示する.

1 本の飲料ページ(@Image.drinktag.html.erb)はトップページの飲料がクリックされると遷移される. 1 本の飲料ページ内の顔の画像を選択する. 選択された人の顔が飲料を飲んだ場面の画像と, 日時を表示する.

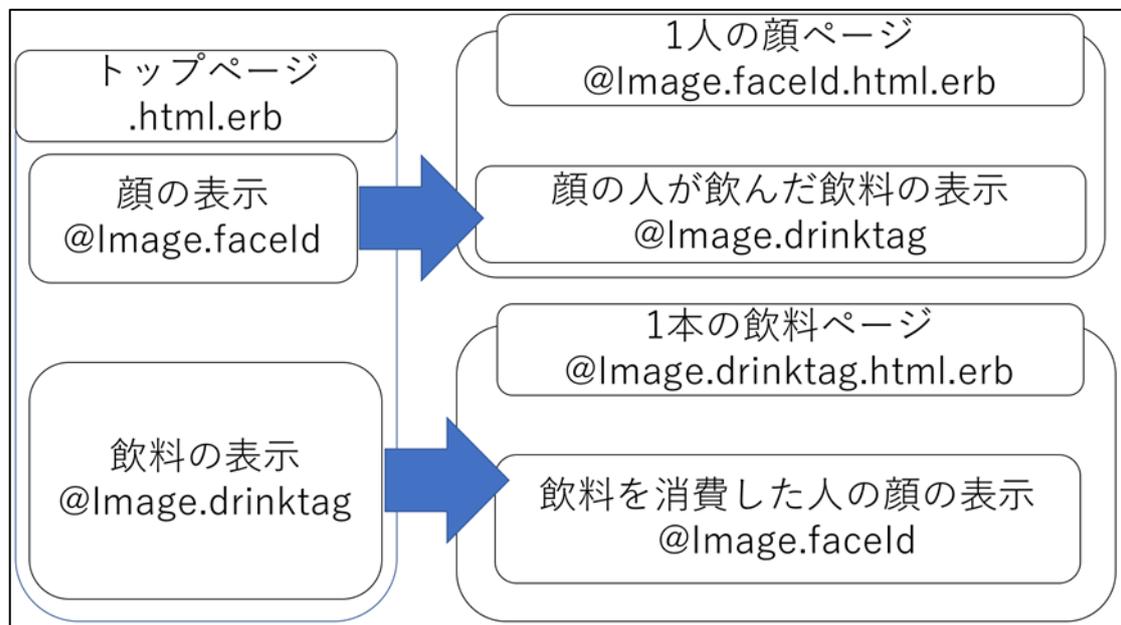


図 : 31 フレームワーク Rails の View に書かれている顔と飲料の表示の処理

4.5. 分析ビューワーの実装

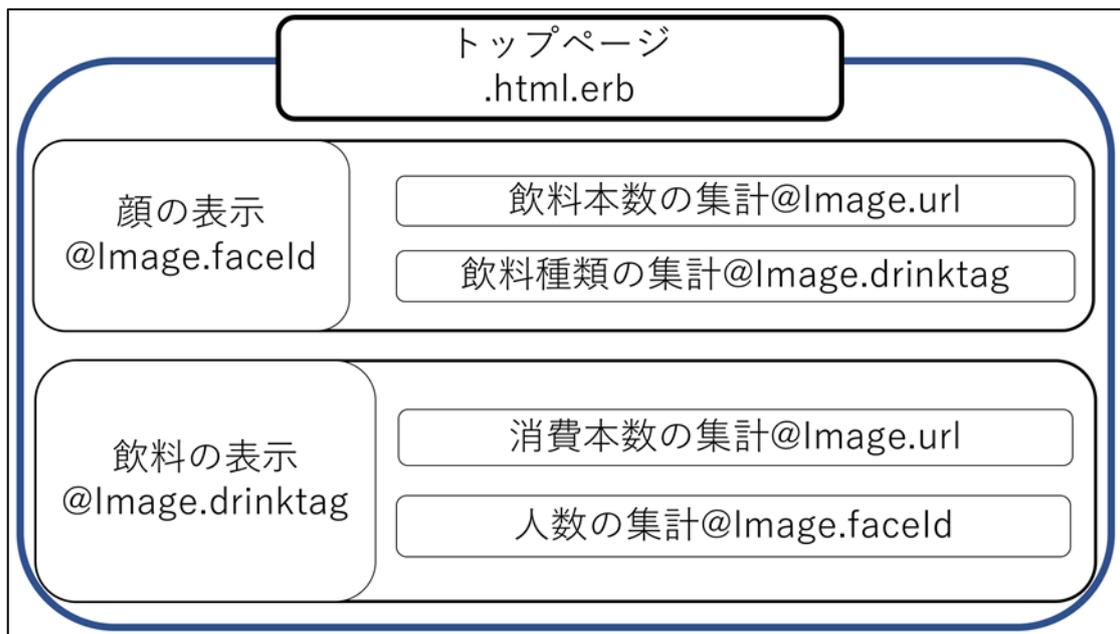
フレームワークの Rails の View に顔と飲料の表示，集計結果の処理が書かれている(図:30). トップページ(/.html.erb)には顔の表示と，飲料の表示をする.顔の表示はデータベースにある顔(faceid カラムに値)を持つ画像を 1 枚表示する.飲料の表示はデータベースにある飲料(drinktag カラムに値)を持つ画像を 1 枚表示する.

顔の表示には，顔の人が飲んだ飲料本数の集計と，飲料種類の集計を表示する.

飲料本数の集計はデータベースにある顔 (faceid の重複する値) が映っている画像(url カラムの値の数)を合計する. 飲料種類の集計はデータベースにある顔 (faceid の重複する値)と同じ画像に含む飲料(drinktag の重複する値と 1 個の値をまとめた数)を合計する.

飲料の表示には，消費本数の集計と，人数の集計をする.

消費本数の集計はデータベースにある飲料 (drinktag の重複する値) が映っている画像 (url カラムの値の数)を合計する. 人数の集計はデータベースにある飲料 (drinktag の重複する値)と同じ画像に含む人の顔(facetag の重複する値と 1 個の値をまとめた数)を合計する.



図： 32 フレームワーク Rails の View に書かれている顔と飲料の表示と集計結果の処理

4.6. 認識部の実装

認識部は Rails フレームワークで開発する.Controller に画像の取得, 顔の認識, 飲料の認識,人と飲料を含む画像の保存が書かれている(図:31).

画像群の取得は Flickr API で実装する.顔認識は Microsoft Face API で実装する.飲料認識は Microsoft Custom Vision API で実装する.[表示メソッド]は Rails の View を通して行う.

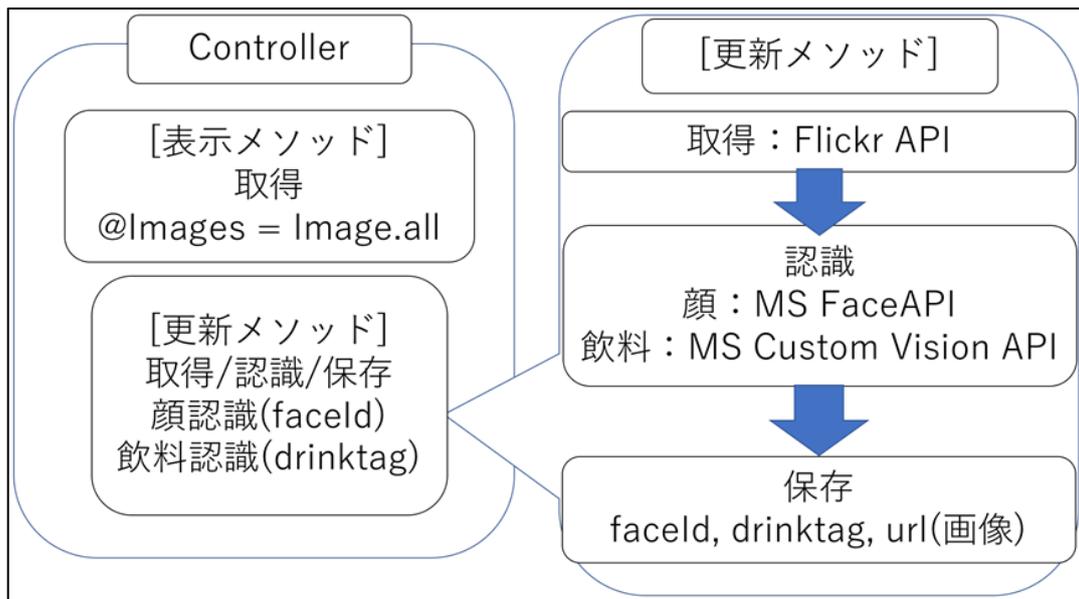


図 : 33 フレームワーク Rails の Controller に書かれている更新メソッドの処理

画像群を取得

画像の取得には FlickrAPI を使用する。(Ruby Gem があるため, gem をインストールする。)フリッカーAPIとは画像を取得する機能を持つ.画像群は 1 個 20 枚で保存されているため,全ての画像を取得した後に 20 枚ずつ配列で分割する.画像は URL を取得する.

Flickr API を用いた画像群の取得

Ruby Gem

Gem flickraw

ライブラリ

```
require 'flickraw'
```

Request Body

フリッカーから全ての画像を total_images に取得

```
total_images = flickr.photos.search
```

```
user_id: {your flickr id}
```

```
per_page: {How many getting photos}
```

表 2 FlickrAPI の Request Body

Field	値	辞書
User_id:	Your flickr id	フリッカーの ID を入力する.記録部の画像が保存されているプロジェクトの ID を記入する.
Per_page:	How many getting photos	取得する画像の枚数を記入する.

画像群から FaceId を検出

画像群から FaceId を取得するために Microsoft Face API を使用する.Face Detection(顔検出)機能を使用することで FaceID を見つける.FaceId とは画像に顔がある時に検出する顔の ID.検出する顔の位置を示す座標と FaceId で返答する.

Microsoft Face API FaceDetection 機能を用いた FaceId の取得

Request URL

<https://westus.api.cognitive.microsoft.com/face/v1.0/detect>

Request Parameters

'returnFaceId' => 'true'

表 3 Microsoft Face API の Request Parameters

Field	値	辞書
returnFaceId	ture	Faceid を取得する.

Request Body

View から取得する画像 url を params[:imageUrl]で受け取る.

```
imageUrl = params[:imageUrl]
request.body = '{"url":"' + imageUrl + '"}';
```

Response Body

表 4 Microsoft Face API の Response Body

Field	値	辞書
faceId	String	顔情報の特定をする.
faceRectangle	String	画像内の顔の位置を特定する.

画像群から顔を認識

画像群から顔を認識するために Microsoft Face API を使用する .Find Similar 機能を使用する .FaceId と FaceList の内部にある persistedFaceId と比較することで、最も類似している顔と類似度を取得する。

FaceList を生成

Create Face List

Request URL

<http://westus.api.cognitive.microsoft.com/face/v1.0/facelists/{faceListId}>

表 5 Create Face List

Field	値	辞書
FaceListId	String	FaceList の ID を表す
name	String	FaceList の名前を表す

FaceList に顔 ID を登録

Add a New Face

Request URL

http://westus.api.cognitive.microsoft.com/face/v1.0/facelists/{faceListId}
/persistedFaces[?targetFace]

表 6 Request URL

Field	値	辞書
FaceListId	String	FaceList の ID を表す
targetFace	String	画像の中にある顔の座標を表す

Request Body

url: imageURL

Response Body

表 7 Response Body

Field	値	辞書
persistedFaceId	String	個別の顔の特徴 ID を表示する

FaceList を取得

Get a FaceList

Request URL

`http://westus.api.cognitive.microsoft.com/face/v1.0/facelists/{faceListId}`

表 8 Request URL

Field	値	辞書
FaceListId	String	FaceList の ID を表す

Request Body

表 9 Request Body

Field	値	辞書
FaceListId	String	FaceList の ID を表す
name	String	FaceList の名前を表す
persistedFaces	Array	顔リストの顔 ID

**Microsoft Face API Find Similar 機能を用いた顔の特定
Request URL**

<http://westus.api.cognitive.microsoft.com/face/v1.0/findsimilars>

Request Parameters

表 10 Request Parameters

Field	値	辞書
faceId	String	顔情報の特定をする.
FaceListId	String	FaceList の ID を表す
maxNumber	Number	最も類似している顔を返す

Response Body

表 11 Response Body

Field	値	辞書
persistedFaceId	String	個別の顔の特徴 ID を表示する
confidence	Number	顔の類似度合いを 1 以下で表示する

画像から飲料を認識

飲料の認識に Microsoft Custom Vision API を使用する。事前に API を編集することで、画像内に特定のオブジェクトがあるかどうかを見つける。画像内の物体の情報を検出する。

Custom Vision API の編集(Prediction-Key と Iteration Id の発行)

編集手順 (1)

Custom Vision API の New Project をクリックし、プロジェクトを作成する。作成後にプロジェクトへアクセスする(図:32 破線内)。

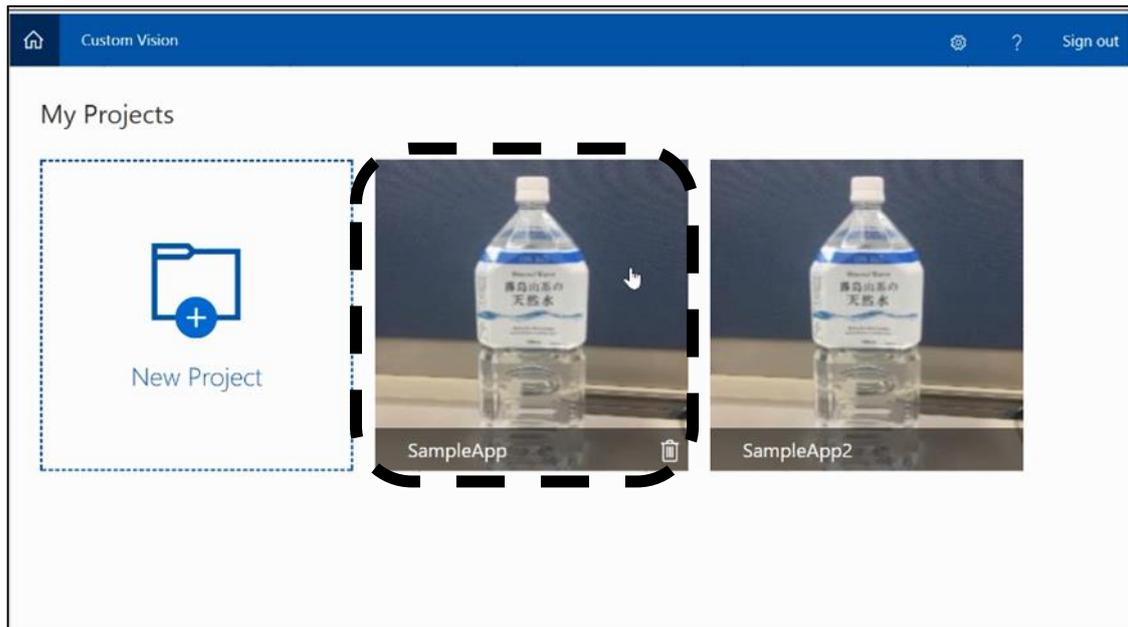
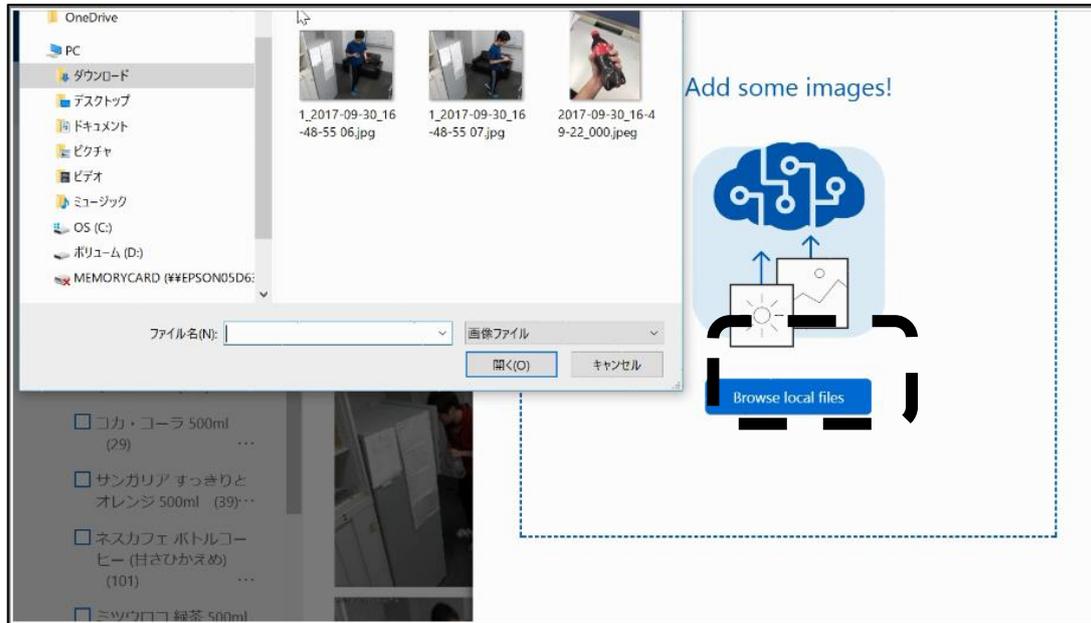


図 : 34 Custom Vision API の編集画面

編集手順 (2)

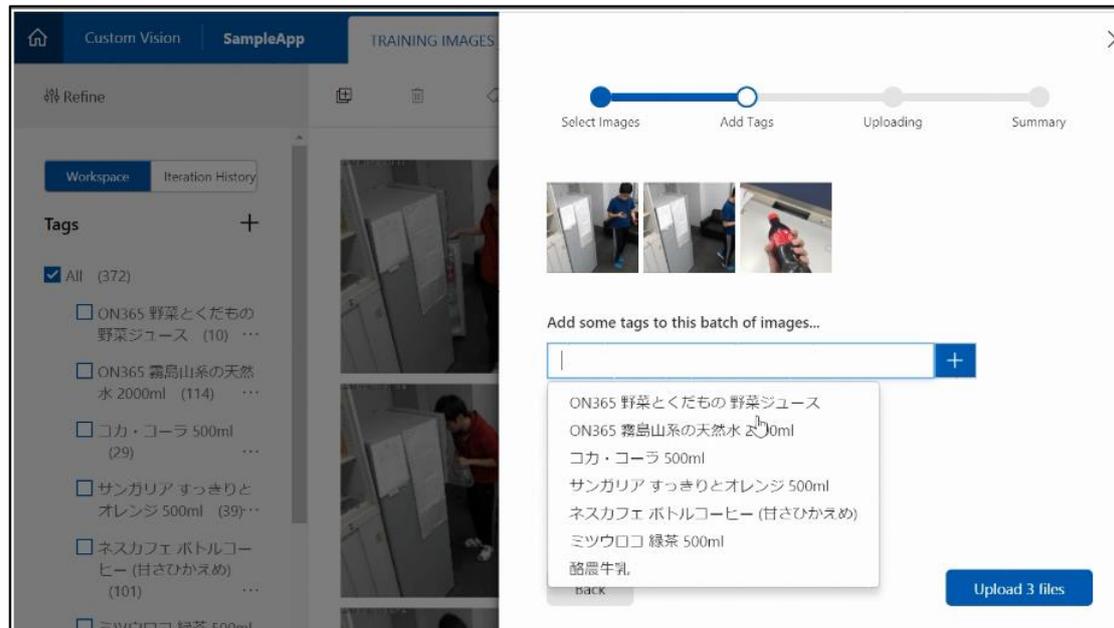
Browse local files をクリックする(図:33 破線内)。ローカルブラウザが起動する。ブラウザから人と物体が写っている画像と物体が大きく写った画像をアップロードする。画像の種類は飲料のみの写った画像か、人が飲料を持った画像を選択する。アップロード画像は Recorder で記録し始める前の日時の画像を選択する。



図： 35 人と物体が写っている画像と物体が大きく写った画像のアップロード画面

編集手順 (3)

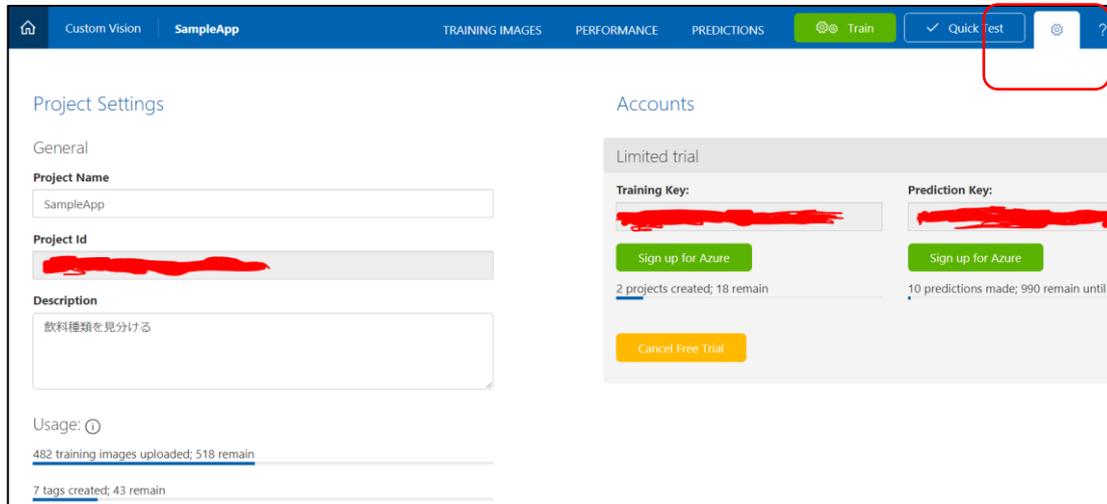
Add some tags to this batch of images...(図:34 破線内) のブランクへカーソルを置く、アップロードした画像に飲料の名前を記入する。Upload files ボタンをクリックする。プロジェクトの編集が完了する。



図： 36 アップロードした画像に飲料の名前を記入

Project-ID と IterationID を取得

プロジェクトのヘッダーにある設定（歯車マーク）をクリックする(図:35 線内)。設定画面(図:35)へ遷移する。Project-ID と Iteration ID をプロジェクトの設定画面から取得する。



図： 37 設定画面（Project-ID と IterationID の取得）

Microsoft Custom Vision API を用いて画像群から飲料名の認識

プロジェクトの設定画面で発行された Project-ID, IterationId を用いて, 画像群から飲料名を認識する.

Request URL

`https://southcentralus.api.cognitive.microsoft.com/customvision/v1.0/Prediction/{Project-ID}/url`

Request Parameters

`'iterationId' => ' {Your iteration Id} '`

表 12 Request Parameters

Field Id	値	辞書
iterationId	Your Iteration Id	飲料リストの使用を許可する.

Request Body

View から取得する画像 url を `params[:imageURL]` で受け取る.

`imageURL = params[:imageURL]`

`request. body = '{"url":"' + imageURL + "'}'`;

Response Body

表 13 Response Body

Field Id	値	辞書
TagId	string	物体の一意の ID を発行する
Tag	string	物体名を発行する
Probability	string	画像と物体の一致度合いを割合で表示する

表 14 Response Body 飲料リスト

TagId	Tag	Probability
Coca_cola Id	Coca_cola	String
Water Id	water	String
Greentea Id	greentea	String
Coffee Id	coffee	String
Orangejuice Id	Orangejuice	String
Vegetablejuice Id	Vegetablejuice	String
Milk Id	Milk	String

4.7. Recorder の実装

記録部の環境は Windows 環境の PC と Web カメラ，固定する三脚，USB-typeA ケーブルを使用する．ソフトウェアは動画生成に iSpy，画像群の生成に Free Video To JPG Converter を使用する．

Web カメラから iSpy によって，冷蔵庫の前に立つ人の飲料を取り出す場面が n 回撮影される．iSpy とは人の動作を検知して動画を撮影するソフトウェアである．Web カメラから n 個の動画を取得する．Free Video To JPG Converter を用いて，1 個の動画を画像 20 枚の画像群へ分割する．すなわち， n 個の動画を n 個の画像群へ変換し，認識部へ渡す．[図:36]



図： 38 Recorder の Web カメラから n 個の画像群を生成する流れ

人と飲料を含む動画を撮影

使用するソフトウェア

iSpy を使用する。iSpy とは、事前に設定を行うことで人の動作を検知する。検知をするとすぐ撮影が開始される。

動作設定の手順(1)

Web カメラとパソコンを接続し、iSpy を起動する。トップページが起動され、画面の歯車のアイコンをクリックする(図:37 破線内)。

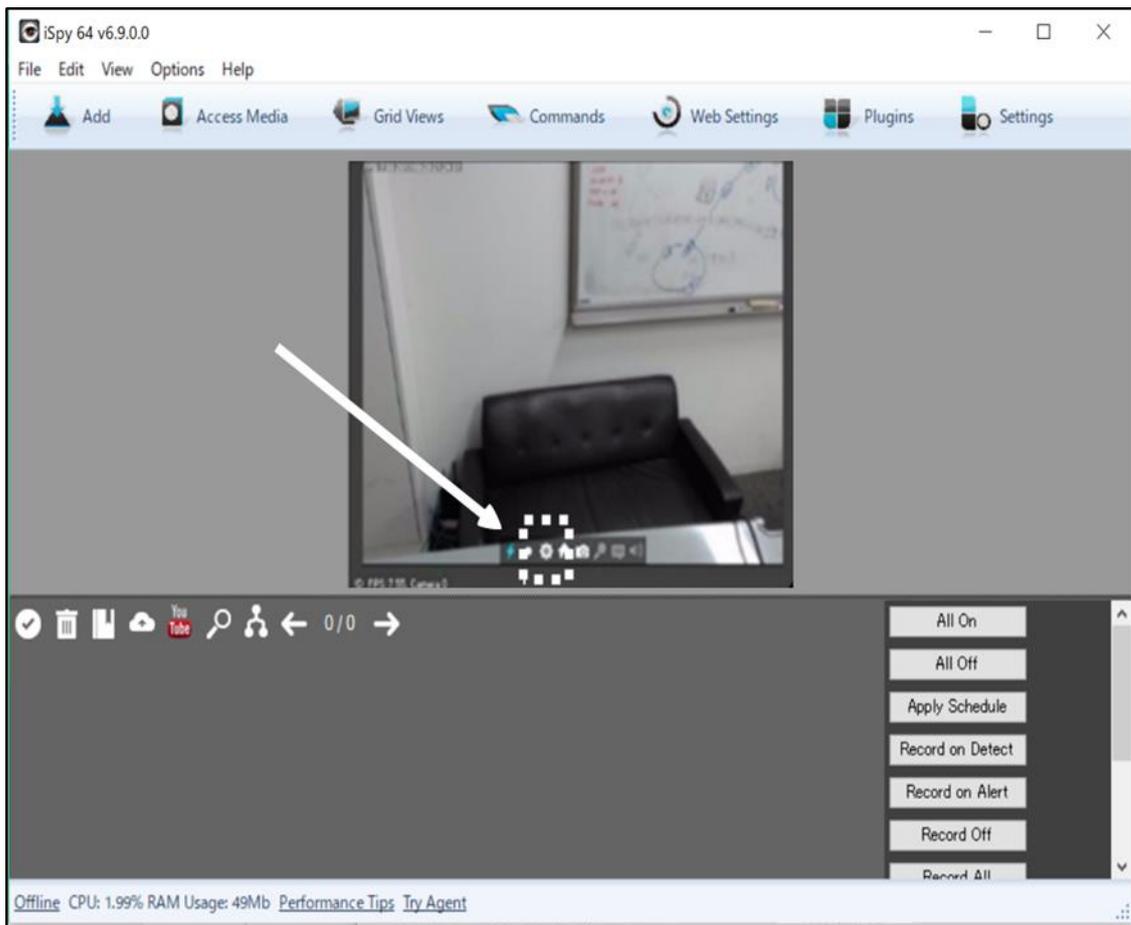


図 : 39 iSpy トップページ 破線内は歯車のマーク

動作設定の手順(2)

iSpy トップページの歯車のアイコンをクリックされると画面(図:38 破線内)が表示される.同画面内(図:39 破線内)の Recording タブをクリックする.

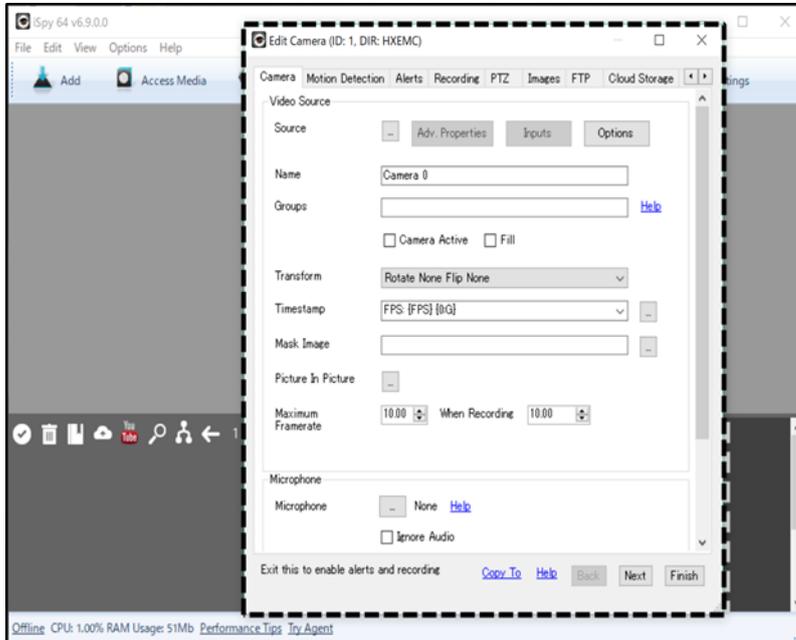


図 : 40 画面(図 : 37)の歯車マークをクリックすると遷移される画面

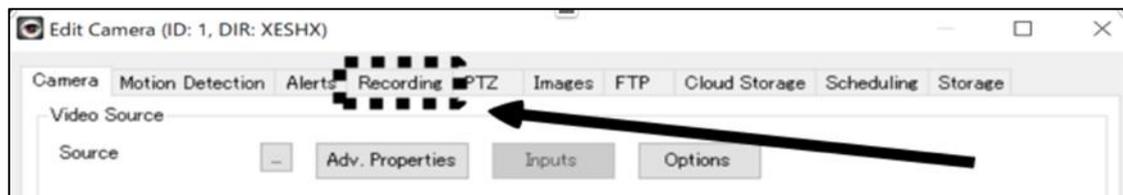


図 : 41 画面(図 : 38)内の Recording タブ

動作設定の手順 (3)

画面(図 : 39)から Recording をクリックすると画面(図:40)へ遷移される。

Recording Mode(録画の方法)の欄から, Record on Movement Detection(動作検知)の横にあるラジオボタンにチェックを入れる。

Recording Settings(録画の設定)の欄から, Min Record Time(最小録画時間)の横にある空欄に 5 秒と記入する。また, Max Record Time(最大録画時間)の横にある空欄に 10 秒と設定する。

Section	Item	Value	Unit
Recording Mode	No Recording	<input type="radio"/>	
	Record on Movement Detection	<input checked="" type="radio"/>	
	Record on Alert	<input type="radio"/>	
Recording Settings	Max. Calibration Delay	15	seconds
	Buffer	0	seconds
	Inactivity Record	0	seconds
	Min Record Time	5	seconds
	Max. Record Time	10	seconds

図 : 42 画面(図 : 39)から Recording をクリックすると遷移される画面

画像 20 枚 n 個の画像群を生成

使用するソフトウェア

FreeVideoToJPGConverter を使用する。FreeVideoToJPGConverter とは、1 個の動画を画像へ分割する。

静止面枚数の設定

Free Video to JPG Converter を起動する。画面（図:42 破線内）にて、合計の横にあるラジオボタンにチェックを入れる。「動画内のフレーム」に 20 と記入する。

取得方法

画面(図:42)の「ファイル追加」をクリックする。動画を選択し、入力する。1 動画を 20 枚の画像へ分割する。画像 20 枚で構成された n 個の画像群を生成する。



図 : 43 Free Video to JPG Converter の画面

第 5 章 関連研究

5.1. How to Browse through my Large Video Data? Face Recognition & Prioritizing for Lifelog Video[7]

ウェアラブルカメラで撮影した 1 人称視点の動画から人の顔を認識する研究である。

ライフログカメラで撮影した動画の容量が大きいため閲覧が困難である。そのため、自動的に顔を認識し、認識した人の顔を含む動画を閲覧するユーザインターフェースの構築を狙う。その際にビデオブラウザを開発し、実際に顔を検出することを試みた。

次の場面①ビデオ撮影中②ビデオ撮影中(動的)③会議④食事から誰が含まれているか表示する。結果として個人の顔を検出する理由の各カテゴリー一覧と計測回数の累積を表示することに成功した。次の属性で評価を行った。①各理由の累積/②各理由の説明/③個人の顔。各累積を元に誰か認識し、正確性の評価を行った。

複数人が動画に含まれる場合、ある個人の信頼度が極端に落ち込む場面が見られた。一画面に複数の人が映り込むと全員の判定が難しいことが報告されている。

5.2. Emotional Video Scene Retrieval System for Lifelog Video Based on Facial Expression Intensity [9]

固定カメラから獲得する動画を画像へ変換し、画像から人の表情を認識する研究である。

1 動画あたり 300-600 枚の画像から人の表情 6 種類の内、撮影している人々の表情がどれにあたるのか自動で判定する。最終的な評価値は 31 項目を 5 属性で分けることで表にまとめた。すると全評価 3.5/5.0 を上回った。

表情の判定にヒートマップの利用を行っている。顔表情の判定のスコアは 0-15 段階を 6 表情中どれが最も高いスコアかどうかで判定を行う。

動画の検索を行う項目は次からなる。「動画のシーン長」「シーン開始時刻」「動画撮影日」「表情強度レベル」「表情の種類：喜び：怒り：悲しみ等」である。膨大な画像群を自動的にシーン別に判別することに成功したと報告されている。

5.3. Lifelogging for 'observer' view memories: an infrastructure approach [15]

ウェアラブルカメラを用いた一人称視点のライフログと、固定カメラを用いた三人称視点のライフログアプローチ法の分類についての研究である。

一人称視点の Narrative Clip を、また三人称視点の FOSCAM をそれぞれ使用している。主に三人称視点の持つメリットが報告されている。特に費用対効果が高い点が挙げられている。固定カメラは比較的低コストで提供されている。外出先、会議室、公共スペース、監視カメラなどさまざまな場所に存在している。記憶容量も本体機器に依存しないため、追加費用を想定しない利点がある。また、三人称視点のデメリットが報告されている。一例では、プライバシーの課題がある。人々が記録するデータを集めることはプライバシーの問題が存在する。pawS のようなシステムはユビキタスコンピューティングに一般的に存在する課題と記述されている。

5.4. Evaluation of a System to Analyze Long-Term Images from a Stationary Camera[10]

固定カメラから長時間（例えば、数ヶ月）にわたって撮影された画像を記録し分析することにより、記録された目標に関する様々な情報を明らかにすることができる。しかし、画像から貴重な情報を迅速に得ることが困難である。

この研究では、上記の問題を解決するため、ヒートマップベースの解析システムを開発した。

ヒートマップ操作画面と時刻操作画面から入力する。出力は 3 人称視点の画像へ赤と緑のヒートマップを重畳させて表示させる。赤のヒートマップは特定の時刻を意味している。緑は別の時刻を意味している。

評価実験を行い、結果的には 2 人の参加者は、「自分の人生のパターンを振り返り、研究室に行く意欲が高まった」とアンケートに答えた。したがって、システムはユーザ自身を分析するのに役立つと推測すると報告されている。

5.5. The Design and Implementation of a Wireless Video Surveillance System [16]

複数の固定カメラから同じ人を撮影し、複数の動画から同じ人を認識する研究である。想定する利用は小売店舗の監視カメラのような状況である。例えば何人がどんな商品に興味を持っているのかを認識することである。

この研究は 3 台のカメラで同じ人の顔を別の角度から撮影した。同じ時系列で人が映った複数の動画から、同じ人の顔画像を抽出することに成功したと報告されている。次に、動画内で撮影した人数を集計している。

5.6. Predicting Important Objects for Egocentric Video Summarization [17]

ウェアラブルカメラから一人称視点の動画を撮影する。動画から重要な物体と撮影者にとって、重要な人を含むシーンを動画で抽出する研究である。

この研究は人か、もしくは物体、あるいは両方を含む動画にアノテーションした人や物体を表示する。認識方法は、手元の近さ、動画に映った時間の長さ、動画に出現する頻度で、撮影者に関わりある人や物体をシーンごとに分けている。

人や物体を含めた動画は次の(a)-(c)のように表示させた。(a)動画に含む人と物体の名前(動画は最長 3 分、動画 1 個は 5 個(1-5)重要な人や物体が含む)。(b)5 個までの物体か人にそれぞれ説明文を表示する。(一例：(1)米の入った黒いポット)。(c)動画に含む 5 個の重要な物体や人にアノテーションを重畳する。

結果的には 1 日中撮影した長時間の 1 人称視点の動画を要約することに成功したと報告されている。また、オブジェクトの認識は、カメラから見える画像に依存しているため、実際の物体と異なって認識される課題があると報告されている。

5.7. ライフログの閲覧と分析に人の表情を活用するシステム[18]

パソコンの前にいる人を Kinect とそれに付随するライブラリによって撮影し、その人の表情を認識した。人の表情を基にライフログの閲覧を支援するシステムを提案した研究である。

システムはデータを記録する記録部と、そのデータを使用して情報の表示を行う分析部からなる。記録部は Kinect から顔画像と表情データ、ライフログデータを記録させる。分析部は記録部からライフログデータの閲覧や、表情を基に似ている表情のシーンを選び出して分析を行う部分である。分析部は閲覧ソフトウェアと分析ソフトウェアからなる。閲覧ソフトウェアは、表情データがグラフで表示される。閲覧者が見たい表情データを見るためのソフトウェアである。分析ソフトウェアは記録部によって保存された表情データを基に、似ている表情シーンを検索しランキングを表示したソフトウェアである。

結果的にはコンピュータは表情を厳密に認識せずとも、ユーザが「心地よい」と感じているという情報さえあれば、同様に心地よいであろうシーンを探してくることに繋げると報告されている。

第 6 章 まとめ

第 1 章ではライフログの伝統的な方法として、一人称視点の画像を取得する方法と、三人称視点の画像を取得する方法をそれぞれ述べた。

一人称視点の画像とは、撮影者の向いた方向の画像が見える。一人称視点の特徴は、同じ人が見る画像を記録し続ける。不利な点は、撮影者自身を記録できないことである。

三人称視点の画像とは、少し離れた位置から見下ろす視点で画像が見える。三人称視点の特徴は、画像に人と物体を撮影する点である。不利な点は、カメラが固定されているため同じ場所しか撮影できないことである。

第 2 章では目的とアプローチを述べた。

本研究の目的は、固定カメラからわかる「さまざまな知見」を「自動的に」得ることであり、固定カメラに人と物体を含めた画像を自動的に撮影した。次に、画像に含む人は誰か、物体は何かを認識した。

次に、人と特定の物体を含む画像を、ビューワーへ自動的に表示した。ビューワーは 2 つからなる。1 つは、「人と飲料に関する記録」を表示させるビューワーである。2 つ目は「人と飲料に関する集計結果」を表示させるビューワーである。

本研究のアプローチは、目的達成のための具体的事例として、研究室の冷蔵庫と人を撮る位置に Web カメラを設置した。冷蔵庫から取り出されるペットボトルの飲料と人の顔を撮影することで、誰が何の飲料を取ったかを判定した。冷蔵庫を選択した理由は 2 つある。1 つは冷蔵庫の前は研究室の中のため、撮影する人数が限定的である点。2 つ目は冷蔵庫にある飲料の本数が限られている点である。そのため、本研究における最初のステップとして冷蔵庫の前の飲料を取り出す人を撮影した。

第 3 章では、システムデザインを述べた。本システムは次の 3 部からなる。

Recorder 部は、Web カメラから iSpy によって、冷蔵庫の前に立つ人の飲料を取り出す場面が撮影される。iSpy とは人の動作を検知して動画を撮影するソフトウェアである。Web カメラから動画を取得し、各動画を画像 20 枚の画像群へ変換し、認識する部分へ渡す。画像の分割には Free Video To JPG Converter というソフトウェアを用いた。このソフトウェアは動画を複数の画像へ分割する。

認識部は、撮影する部分から画像群を取得する。画像の取得には FlickrAPI を用いる。FlickrAPI とはフリッカーから画像を取得する。各画像群から顔と飲料を含む画像 1 枚を抽出し、画像をビューの部分へ渡す。顔の認識には Microsoft Face API を用いる。FaceAPI とは顔を認識する。飲料の認識には Microsoft Custom Vision API を用いる。この API は特定の物体を認識する。

Viewer 部は、人に関する記録と飲料に関する記録を閲覧する検索ビューワーと、人と飲料に関する集計結果を閲覧する分析ビューワーからなる。

3 ヶ月間、Web カメラから測定を行い、人と飲料に関する記録、人と飲料の集計結果をビューワー(検索ビューワーと分析ビューワー)へ自動的に表示させることに成功した。

検索ビューワーでは、人に関する記録(誰がどんな飲料を飲んでいるのかの記録)と、飲料に関する記録(飲料が誰から飲まれてるかの記録)を表示させることに成功した。

分析ビューワーでは、人に関する集計結果(どの人が飲料を何本飲むか、どの人が何種類の飲料を飲むか)と、飲料に関する集計結果(どの飲料が合計何本消費されたか、どの飲料が何人から消費されたか)を表示させることに成功した。

第4章ではシステムの実装について述べた。

本システムは Rails フレームワークで作成した。ブラウザから http の入力が行われると、Rails の遷移順 (ブラウザ→View→Root→Controller→Model→Controller→View→ブラウザ) でブラウザへレスポンスが返される。誰が何の飲料を飲んだかの画像の表示と更新をした。View にはブラウザへ人と飲料を含む画像を表示する処理が記述されている。View の処理は Html, Scss, Javascript, Ruby で記述した。Controller には 20 枚ずつ画像の取得/人と飲料を含む画像 1 枚の認識/画像の保存の処理が記述されている。Controller の処理は Ruby で記述した。Model にはデータベースのデータ処理が記述されている。Model の処理は Ruby で記述した。Root には検索ビューワーと分析ビューワーのブラウザから http のアクセスに応じてコントローラー内の各アクション(メソッド)が応じるルート処理が記述されている。Root は Ruby で記述した。

第5章では本研究に関連する研究について述べた。

参考文献

- [1] C. Gurrin, A. F. Smeaton, and A. R. Doherty, “LifeLogging: Personal Big Data,” *Foundations and Trends*, vol. 8, no. 1, pp. 1–125, 2014.
- [2] A. J. Sellen, A. Fogg, M. Aitken, S. Hodges, C. Rother, and K. Wood, “Do life-logging technologies support memory for the past?,” *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, p. 81, 2007.
- [3] K. Aizawa and T. Hori, “Context-Based Video Retrieval for Life-Log Applications,” *Multimedia Content and the Semantic Web*, pp. 373–387, 2005.
- [4] X. Sun, H. Yao, W. Jia, and M. Sun, “Eating Activity Detection from Images Acquired by a Wearable Camera,” *International SenseCam & Pervasive Imaging Conference*, no. 3, pp. 80–81, 2013.
- [5] A. R. Doherty, N. Caprani, C. Ó. Conaire, V. Kalnikaite, C. Gurrin, A. F. Smeaton, and N. E. O’Connor, “Passively recognising human activities through lifelogging,” *Computers in Human Behavior*, vol. 27, no. 5, pp. 1948–1958, 2011.
- [6] D. McDuff, A. Karlson, A. Kapoor, A. Roseway, and M. Czerwinski, “AffectAura: An Intelligent System for Emotional Memory,” *ACM Conference on Human Factors in Computing Systems(CHI)*, p. 849, 2012.
- [7] K. Wolf, Y. Abdelrahman, M. Landwehr, G. Ward, and A. Schmidt, “How to browse through my large video data,” *Proceedings of the 15th International Conference on Mobile and Ubiquitous Multimedia (MUM)*, pp. 169–173, 2016.
- [8] A. R. Doherty, A. F. Smeaton, K. Lee, and D. P. W. Ellis, “Multimodal segmentation of lifelog data,” *RIAO LargeScale Semant. Access to Content Text Image Video Sound*, no. 2006, pp. 21–38, 2007.
- [9] M. Maeda, H. Nomiya, S. Sakaue, T. Hochin, and Y. Nishizaki, “Emotional video scene retrieval system for lifelog videos based on facial expression intensity,” *IEEE/ACIS International Conference on SNPD*, pp. 551–556, 2017.
- [10] A. Ishii, T. Abe, H. Hakoda, B. Shizuki, and J. Tanaka, “Evaluation of a System to Analyze Long-Term Images from a Stationary Camera,” *HIMI Information, Design and Interaction*, pp. 275–286, 2016.
- [11] H. Joho, J. M. Jose, R. Valenti, and N. Sebe, “Exploiting facial expressions for affective video summarisation,” *Proceedings of the ACM International Conference on Image and Video Retrieval*, 2009.
- [12] T. Zhao and R. Nevatia, “Tracking multiple humans in crowded environment,” *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 406–413, 2004.
- [13] M. Romero, J. Summet, J. Stasko, and G. Abowd, “Toward Visualizing Video through,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 14, no. 6, pp. 1261–1268, 2008.
- [14] C. Vondrick, A. Shrivastava, A. Fathi, S. Guadarrama, and K. Murphy, “Tracking Emerges by

- Colorizing Videos,” Google Research, 2018.
- [15] S. Clinch, P. Metzger, and N. Davies, “Lifelogging for ‘Observer’ View Memories: An Infrastructure Approach,” *ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pp. 1397–1404, 2014.
- [16] T. Zhang, A. Chowdhery, P. Bahl, K. Jamieson, and S. Banerjee, “The Design and Implementation of a Wireless Video Surveillance System,” *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking(MobiCom)*, pp. 426–438, 2015.
- [17] Y. J. Lee and K. Grauman, “Predicting Important Objects for Egocentric Video Summarization,” *International Journal of Computer Vision*, vol. 114, no. 1, pp. 38–55, 2015.
- [18] 高井 啓, “ライフログの閲覧と分析に人の表情を活用するシステム,”筑波大学 システム情報工学研究科 修士論文, 45 pages, 2014.

概要

本研究の目的は、固定カメラからわかる「さまざまな知見」を「自動的に」得ることであり、固定カメラに人と物体を含めた画像を自動的に撮影した。次に、画像に含む人は誰か、物体は何かを認識した。

最後に、人と特定の物体を含む画像を、ビューワーへ自動的に表示した。ビューワーは2つからなる。1つは、「人と飲料に関する記録」を表示させるビューワーである。2つ目は「人と飲料に関する集計結果」を表示させるビューワーである。

目的達成のための具体的事例として、研究室の冷蔵庫と人を撮る位置に Web カメラを設置した。冷蔵庫から取り出されるペットボトルの飲料と人の顔を撮影することで、誰が何の飲料を取ったかを判定した。

本システムは次の3部からなる。

Recorder 部は、Web カメラから iSpy によって、冷蔵庫の前に立つ人の飲料を取り出す場面が撮影される。iSpy とは人の動作を検知して動画を撮影するソフトウェアである。Web カメラから動画を取得し、各動画を画像 20 枚の画像群へ変換し、認識する部分へ渡す。画像の分割には Free Video To JPG Converter というソフトウェアを用いた。このソフトウェアは動画を複数の画像へ分割する。

認識部は、撮影する部分から画像群を取得する。画像の取得には FlickrAPI を用いる。FlickrAPI とはフリッカーから画像を取得する。各画像群から顔と飲料を含む画像 1 枚を抽出し、画像をビューの部分へ渡す。顔の認識には Microsoft Face API を用いる。FaceAPI とは顔を認識する。飲料の認識には Microsoft Custom Vision API を用いる。この API は特定の物体を認識する。

Viewer 部は、人に関する記録と飲料に関する記録を閲覧する検索ビューワーと、人と飲料に関する集計結果を閲覧する分析ビューワーからなる。

3ヶ月間、Web カメラから測定を行い、人と飲料に関する記録、人と飲料の集計結果をビューワー(検索ビューワーと分析ビューワー)へ自動的に表示させることに成功した。

検索ビューワーでは、人に関する記録(誰がどんな飲料を飲んでいるのかの記録)と、飲料に関する記録(飲料が誰から飲まれてるかの記録)を表示させることに成功した。

分析ビューワーでは、人に関する集計結果(どの人が飲料を何本飲むか、どの人が何種類の飲料を飲むか)と、飲料に関する集計結果(どの飲料が合計何本消費されたか、どの飲料が合計何人から消費されたか)を表示させることに成功した。