

ホームロボットにおける ディープラーニングによる物体認識

田向 権 (Hakaru Tamukoh)
九州工業大学 大学院 生命体工学研究科

tamukoh@brain.Kyutech.ac.jp

ひびきのAI社会実装研究会 実践的人工知能入門
ディープラーニングの社会実装を目指して
@ひびきの学研都市 2016年9月28日



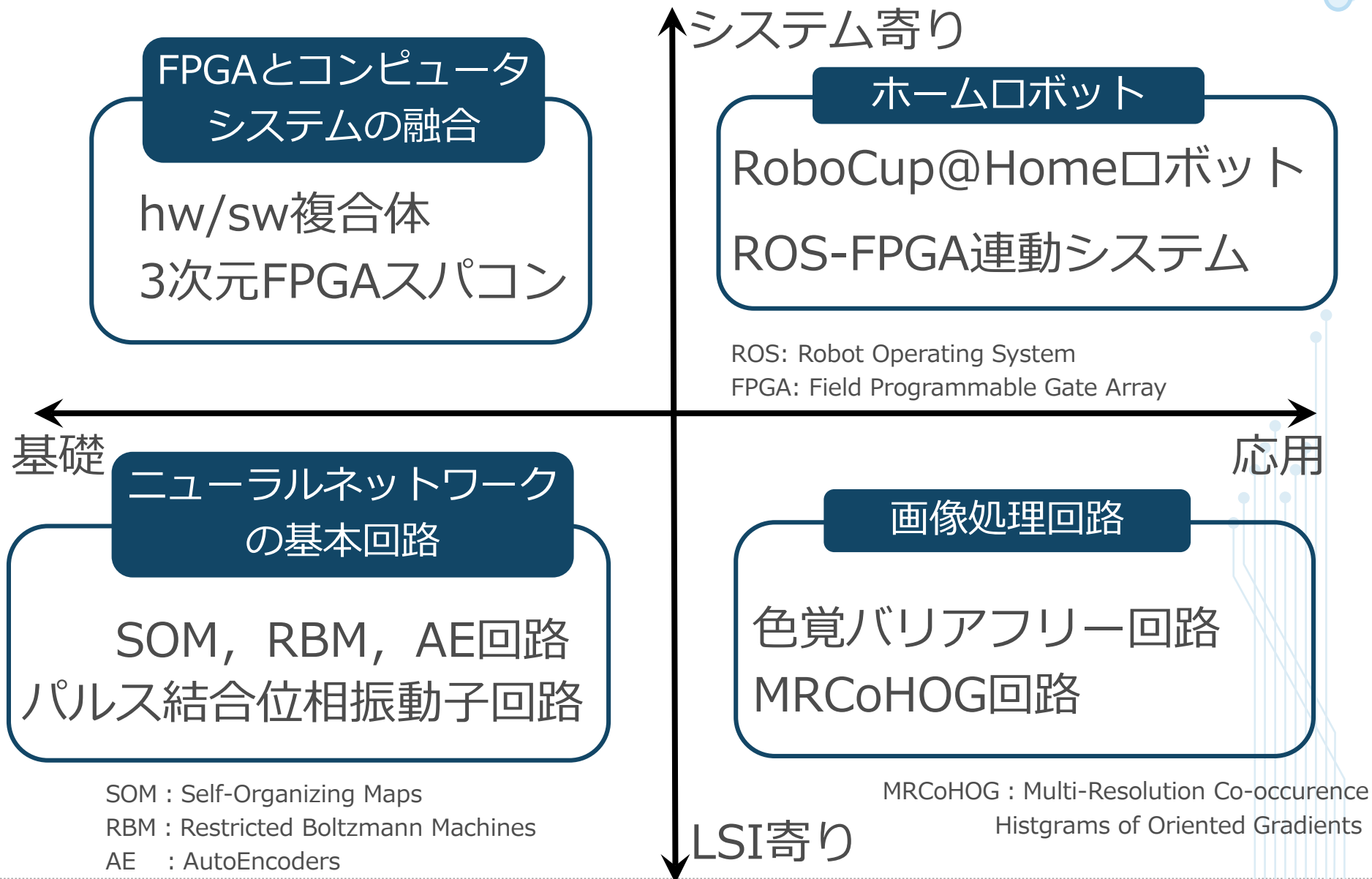
目次

1. 自己紹介
2. ロボカップ@ホームロボット
 - ✓ ロボカップ@ホームとは
 - ✓ ホームロボットの要素技術
3. ホームロボット向け物体認識・把持システムの構築
4. 実践！DCNNによる画像認識器の構築法
 - ✓ GoogLeNetと転移学習
 - ✓ 2016年9月時点でのお勧めGPU
5. 研究紹介：エッジ指向型脳型計算機システムの実現に向けて
 - ✓ ROS-FPGA連携システム
 - ✓ DNNの基本回路（時間があれば）
 - ✓ 組込指向DNNの世界的動向

自己紹介

- 学生時代より15年間一貫してニューラルネットワークのデジタル回路実装に関する研究に従事，**応用も重視**.
 - ✓ 2006年3月 九工大 脳情報専攻 博士課程修了
 - 博士論文「粗い勝者決定を用いた自己組織化マップおよび自己組織化関係ネットワークのハードウェア化」（指導教員：山川 烈 先生）
- 学会，他研究機関における活動
 - ✓ 電子情報通信学会スマートインフォメディアシステム研究専門委員会 幹事補佐，専門委員
 - ✓ 日本神経回路学会 理事
 - ✓ 東京大学生産技術研究所 協力研究員
 - ✓ 一般社団法人 ファジィシステム研究所 主任研究員
 - ✓ 一般社団法人 行動評価システム研究所 主任研究員

研究テーママップ





ロボカップ@ホームロボット

RoboCup@Homeとは

- RoboCupのリーグのひとつ
 - ✓ 他にはサッカー, レスキューがある
- 日常生活に役立つ機能を制限時間内に達成
 - ✓ 日用品の探索, 言われたものを取ってくる, 人を追従するなど
 - ✓ 画像や音声を用いて人間とコミュニケーションを取る
- 生活支援ロボットのコンペティションとしては世界最大
 - ✓ タスクはロボットのベンチマークテストとして明文化
 - ✓ ひとつのタスクに複数の技術課題を含むように設計

杉浦孔明, “ロボカップ@ホーム 人と共存するロボットのベンチマークテスト,”
人工知能, Vol.31, No.2, 2016年3月 (人工知能学会 学会誌 解説論文) .

RoboCup@Homeの立ち位置

×技術教育向けコンテスト(例: 高専／NHK大学ロボコン)

○ホームロボットの實用化を強く意識したコンテスト



RoboCup@Home Japan Open 2016 (23チーム, 163名 参加)

ロボカップ紹介ビデオ



- 2010年世界大会

- ✓ eR@sers（玉川大，電通大，NICT連合チーム）優勝

<https://youtu.be/2HldlwjK0TM>

ロボカップ関連の社会への波及力 1

- Quince（東北大・田所研究室） <http://www.rm.is.tohoku.ac.jp/>
 - ✓ ロボカップレスキューから福島第一原発原子炉建屋投入へ
- Kiva systems <http://raffaello.name/>
 - ✓ ロボカップ小型リーグで培った技術を元に創業，後にAmazonに7.75憶ドルで買収 ⇒ Amazon配送センターのロボットとして活用，年間4.5～9億ドルのコスト削減効果

Quince

Kiva systems

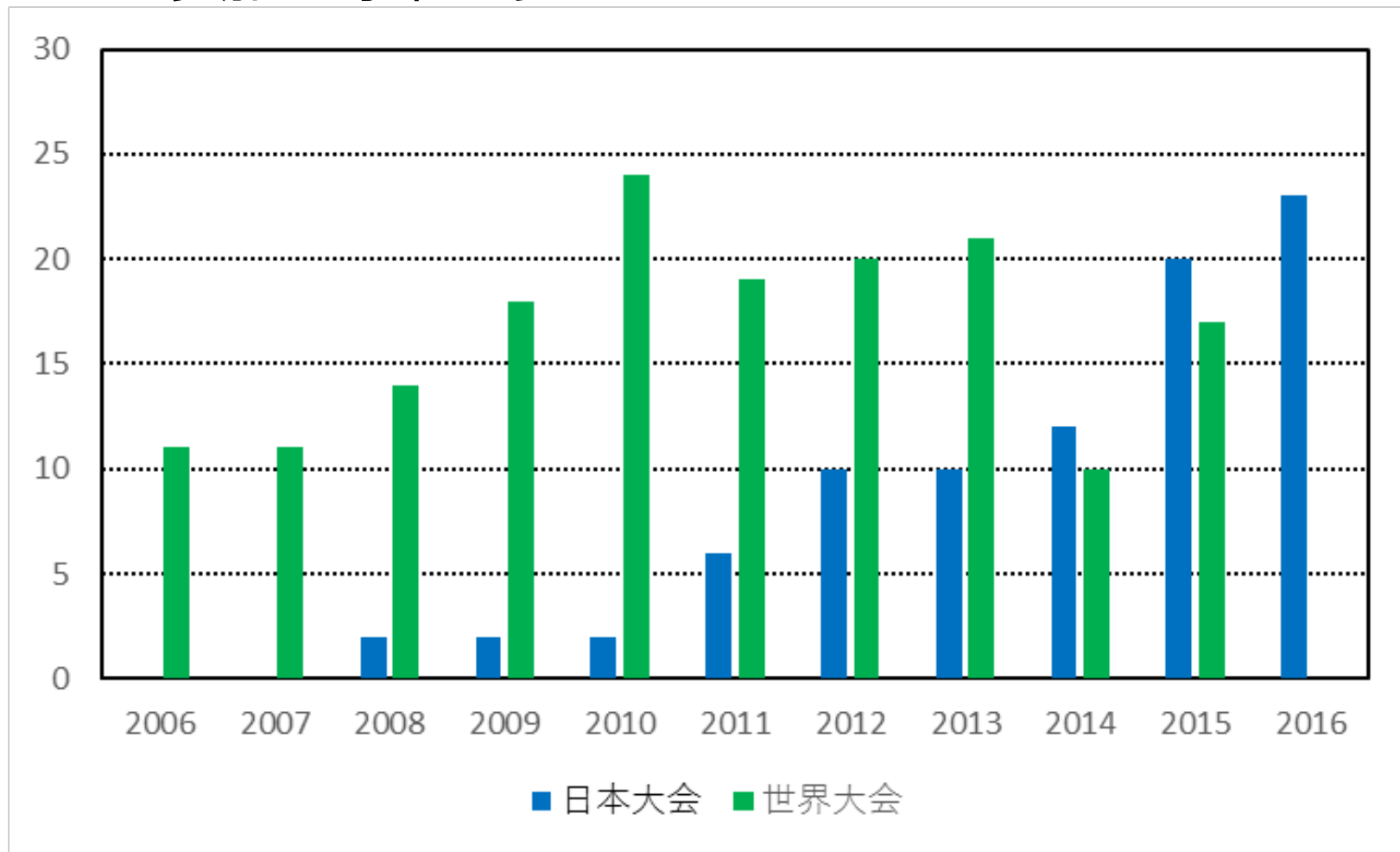
ロボカップ関連の社会への波及力 2

- NAO (Aldebaran) <https://www.aldebaran.com/>
 - ✓ ロボカップサッカー標準プラットフォームに選定されたことを契機に研究コミュニティに普及, 後にソフトバンクからの1億ドルの出資につながり, Pepperを構築するに至る
- HSR (トヨタ) http://www.toyota.co.jp/jpn/tech/partner_robot/
 - ✓ 2016年3月のRoboCup Japan Openよりトヨタのホームロボット HSRが@ホームリーグへ本格参入, 電通大, 玉川大で採用

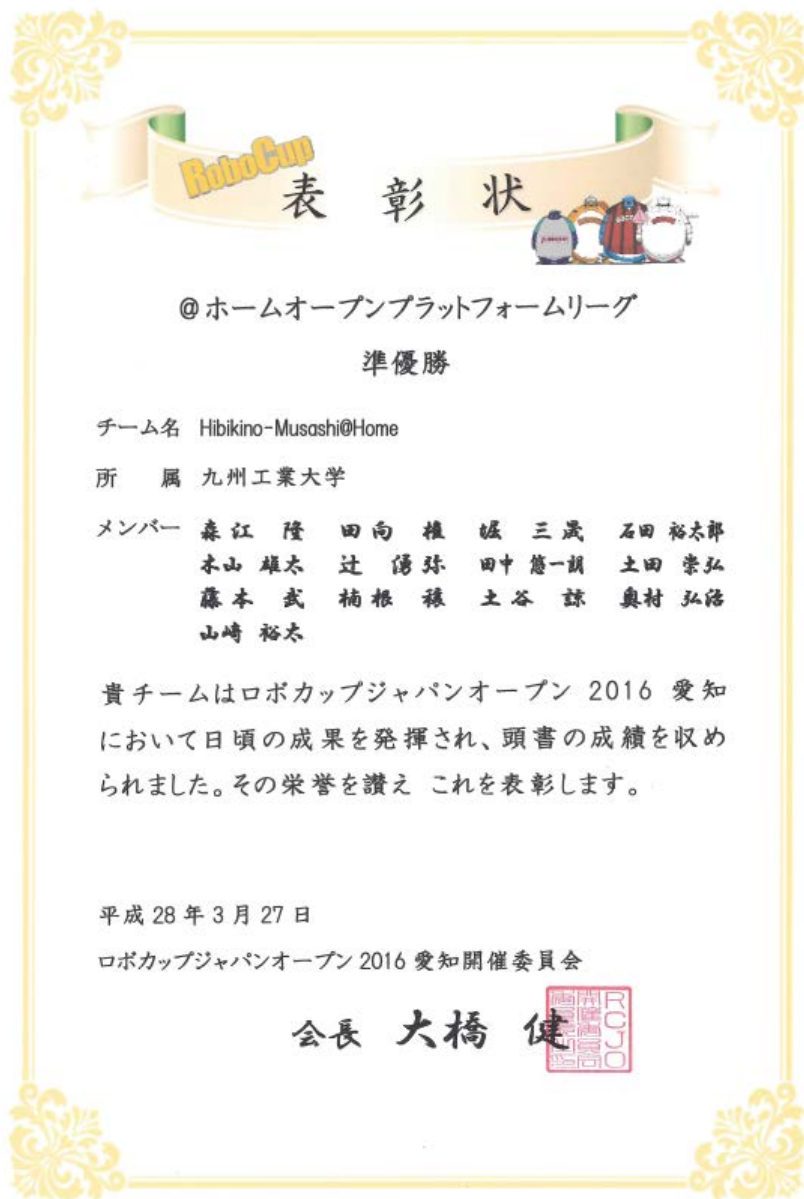
TOYOTA HSR

参加チーム数推移

- 日本大会の参加者は急激に増加中，台湾やタイ，ベトナムからの参加も毎年あり



Hibikino-Musashi@Homeチームの紹介



RoboCup Japan Open 2015
@ホーム実機リーグ **3位入賞**

RoboCup Japan Open 2016
@ホーム実機リーグ **準優勝**

九州工業大学の様々な研究室
の研究成果を集積

ロボティクス, 人工知能,
脳型チップの社会実装を目指
した研究プラットフォーム

Exi@:人間の生活空間で活躍する知能ロボット

ショットガンマイク：
声を聞き取る耳

ロボットアーム：
物を掴む腕

お茶を取ってきて！

分かりました

Laser Range Finder (LRF)：
周囲を見渡す目

SLAMによる
地図生成



RGB-Dカメラ × 2：
前を見つめる目

FPGAとPC × 2：
全体を制御する脳
ROSでコントロール

全方向移動台車：
全方向に動く足

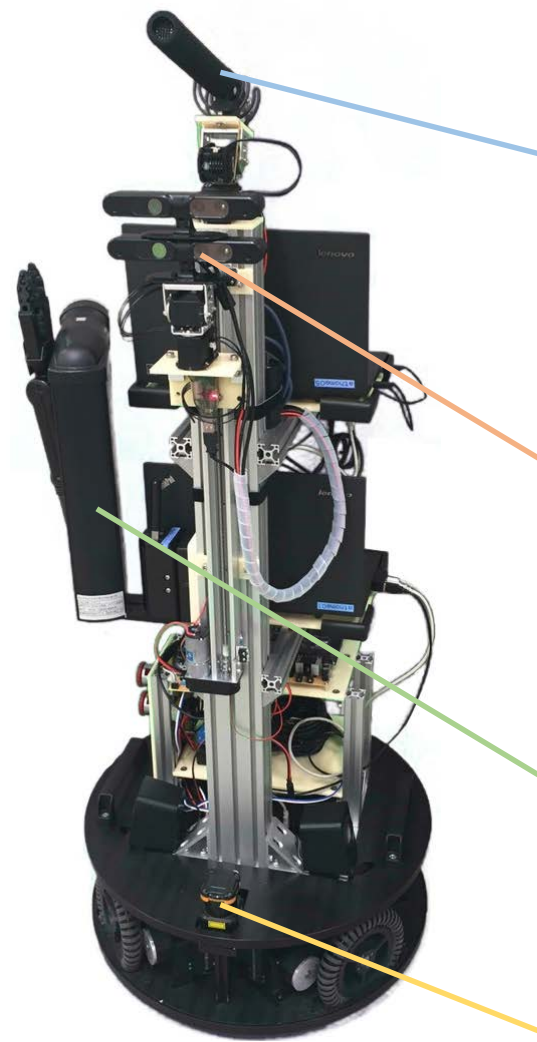


ロボカップに参加中！
2016年は準優勝！

「Hibikino-Musashi@Home」
でFacebookページを検索！



Exi@の主要機能



サービスロボット Exi@

音声インタラクション

音声認識

音声合成

画像処理

物体検出

人物検出・認識

物体認識 (Deep Learning)

人物追跡

ロボットアームビジュアルフィードバック

ARマーカ認識

把持経路設計

自律ナビゲーション

SLAM

移動経路設計

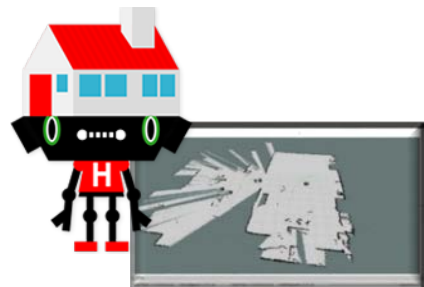
家具認識

RoboCup@Home タスク一覧（2015年ルール）

Stage	タスク名	内容
1	Manipulation and Object Recognition	棚に置いてある物体についてPick & Place タスクを複数回行う
	Navigation	フィールドの中を移動したのちフィールド外で人を追従する
	Person Recognition	複数の人物が居る状況において、顔画像の学習と性別推定を行う
	Speech Recognition & Audio Detection	音声認識および音源方向推定を計10回行う
	RoboZoo	並んだ展示ブースでロボットに自由にパフォーマンスを行わせ、観客に投票させる
	General Purpose Service Robot	ランダム生成された音声命令を解釈し、曖昧な指示を明確化する（例：飲み物を持ってきて ⇒ 何の飲み物ですか？）と共に実行する
2	Open Challenge	自由に選んだテーマに関する研究要素のデモンストレーション
	Restaurant	レストランにおいて、事前に場所を学習したのち、ロボットを呼んだ人を発見して注文を取り、注文されたオブジェクトを届ける
	Robo-Nurse	ユーザの音声命令に従い指示された薬瓶を探して持ってくると共に、ユーザの動作（倒れる等）を認識して必要な支援を行う
	Wake Me Up	ユーザに注文された朝食を運ぶと共にスマートホームの制御を行う
決勝	Final	自由に選んだテーマに沿ったデモンストレーション

タスクRestaurantの場合

①人間に教わりながら、事前に店の環境地図を作成する

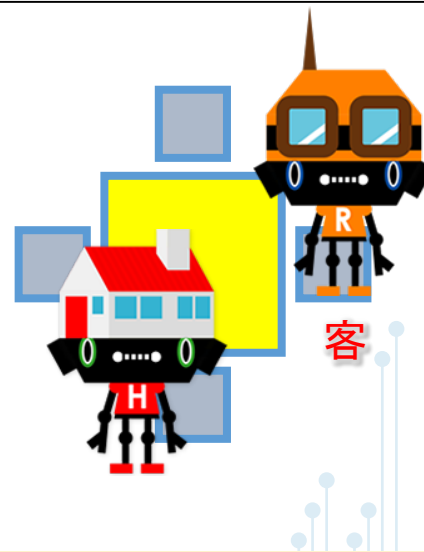


③呼ばれている事に気づき
注文を取りに行く(オプション)

Hey



②指定の場所で注文を取る

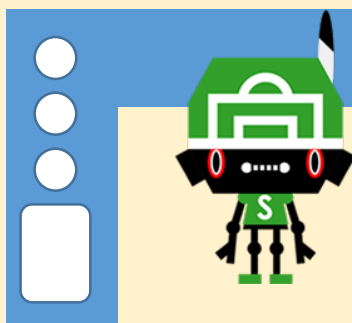


④バーテンに注文を
伝え、用意してもらう。



ロボット

⑤それをアームで把持し
⑥注文した人のところまで
持っていく



バーテン

キッチンバー

チェックポイント毎に点数が設定、制限時間内に出来た項目の合計点で勝負

タスクRestaurantのデモンストレーション

端末

/home/athome05/iyutaro_ws/src/restaurant_pkg/launch/restaurant.launch ht

Mono: Playback 102 [80%] [19.12dB] [off] Capture 13 [81%] [19.34dB] [on]

各種ログ

ステートマシン

```
graph TD
    WaitStartSig([WaitStartSig]) -- exit1 --> SRec_StartFollow([SRec_StartFollow])
    SRec_StartFollow -- exit1 --> Img_MemoWaiter([Img_MemoWaiter])
    Img_MemoWaiter -- exit1 --> FollowWaiter([FollowWaiter])
    FollowWaiter -- exit2 --> SRec_WhichSidesKitchen([SRec_WhichSidesKitchen])
    FollowWaiter -- exit1 --> SRec_TablePos([SRec_TablePos])
    SRec_WhichSidesKitchen -- exit1 --> SLAM_RecordKitchenPos([SLAM_RecordKitchenPos])
    SLAM_RecordKitchenPos -- exit1 --> SRec_WhichTableToAttend([SRec_WhichTableToAttend])
    SRec_TablePos -- exit1 --> SLAM_RecordTablePos([SLAM_RecordTablePos])
    SRec_WhichTableToAttend -- exit1 --> SRec_WhichTableToAttend
```

環境地図

物体認識

深度画像

RGB画像

28 fps

The screenshot displays a ROS environment. On the left, a terminal window shows the execution of a launch file and audio status. Below it, a state machine diagram illustrates the robot's task flow, starting from 'WaitStartSig' and moving through various states like 'SRec_StartFollow', 'Img_MemoWaiter', and 'FollowWaiter'. The 'FollowWaiter' state branches into 'SRec_WhichSidesKitchen' and 'SRec_TablePos', which lead to SLAM recording states. On the right, the RViz interface shows a 3D environment map with a robot's path. Below the map, two camera feeds are displayed: 'Image' (labeled 'No Image') and 'Image' (showing a depth map). At the bottom, an RGB camera feed shows a person in a grey hoodie. The bottom right corner indicates a frame rate of 28 fps.

<https://youtu.be/Bsrrk1pNPbc>

タスクRestaurantに必要な要素技術一覧

- ① ウェイター（人間）に教わりながら事前に環境地図を作成してテーブルの場所を覚える
 - ✓ SLAM (Simultaneous Localization And Mapping)
 - ✓ 自然な音声対話
 - ✓ ランドマーク登録
- ② 指定の場所で注文を取る
 - ✓ SLAM + 経路生成 + 移動台車制御
 - ✓ 自然な音声対話と意味理解
- ③ 呼ばれていることに気づき注文を取りに行く
 - ✓ 画像認識（手振りなど）
 - ✓ 音声認識 + 音源定位

タスクRestaurantに必要な要素技術一覧

- ④ バーテンに注文を伝え用意してもらう
 - ✓ 自然な音声対話
- ⑤ 用意された物体の中から注文されたものを把持する
 - ✓ 画像（RGB-D）からの物体切出しと物体認識
 - ✓ アーム制御
- ⑥ 注文した人のところまで持っていく
 - ✓ SLAM + 経路生成 + 移動台車制御

大会本番では多数の観客の声や顔・シルエット、
会場の照明・外光などが、音声・画像認識のノイズとなる
頑健な知的処理システムの構築が求められる

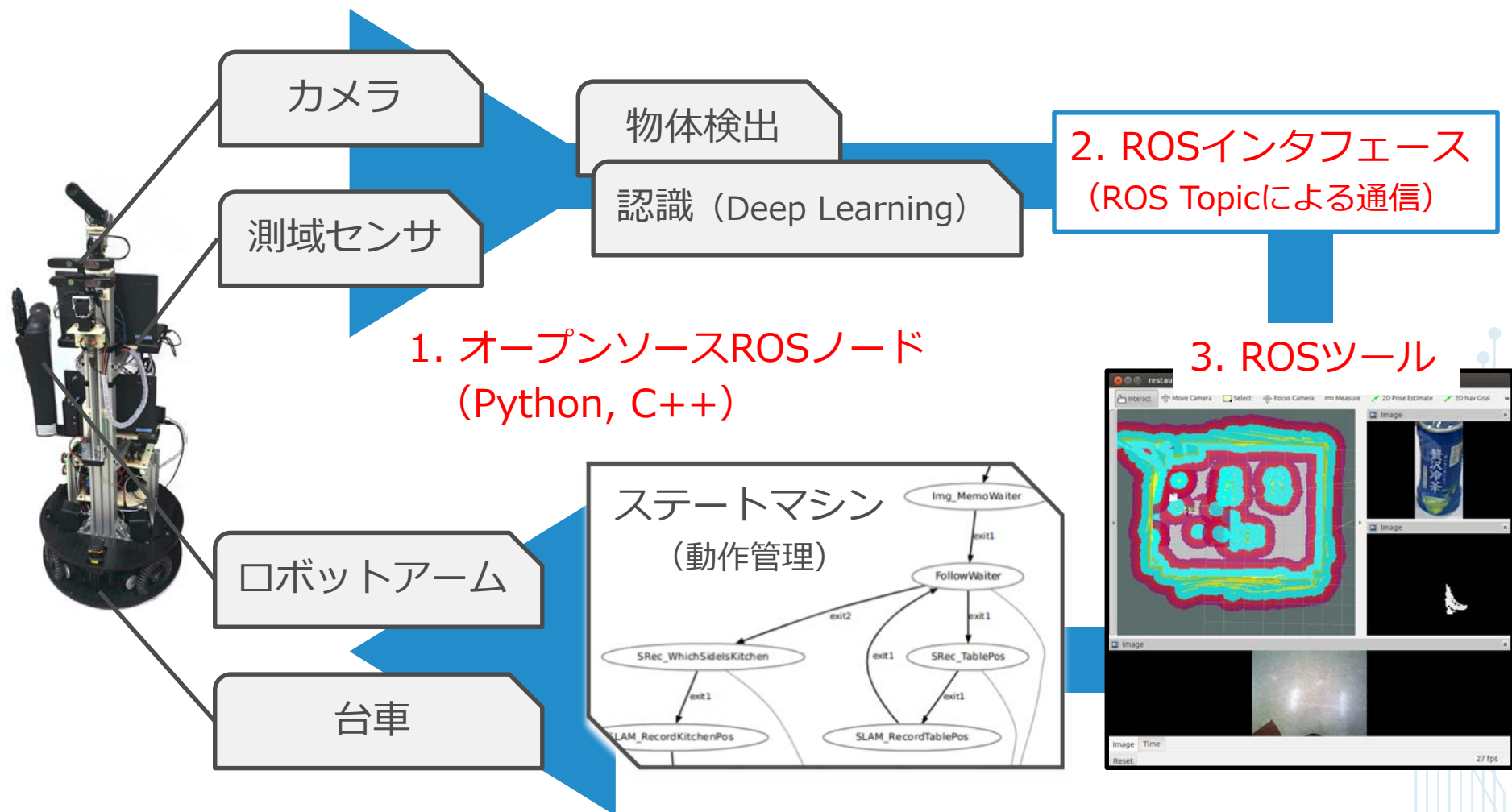
Robot Operating System (ROS)

- ロボット向けミドルウェア
 - 米国のロボットベンチャー企業 Willow Garage 社が開発
(現在は, Open Source Robotics Foundationが開発)
 - ロボットシステム構築のデファクトスタンダード
- ✓ ROS Wiki アクセス数: 110万アクセス/年 (2015) [1]
 - ✓ ROSに関する論文数: 1843本 (Google Scholar 2015/09/11現在) [1]
 - ✓ オープンソースパッケージ数: 3000パッケージ以上 [2]

[1] Community Metrics Report 2015,
<http://download.ros.org/downloads/metrics/metrics-report-2015-07.pdf>

[2] Community Metrics Report 2011,
<http://download.ros.org/downloads/metrics/metrics-report-2011-08.pdf>

ROSによるロボットシステムの構築



ROSノードの組換えでロボットシステムの変更・拡張が簡単かつ迅速に行える

ソフトコンピューティング・AIはどこに？

- Deep Convolutional Neural Networks (DCNN) と転移学習を用いた物体認識
- Particle Filter と深度画像を用いた人物追跡
- Gabor Filterとエラスティックグラフマッチングによる人物認識
 - ✓ M. Tanaka, et al., "Human Detection and Face Recognition Using 3D Structure of Head and Face Surfaces Detected by RGB-D Sensor," Journal of Robotics and Mechatronics, 2015.
- 自然な音声対話 （参考：記号創発ロボティクス）
 - ✓ 音声認識にはJuliusを利用，音声認識後の意味理解に機械学習（現状，我々チームは言葉揺れの統合のみ：取ってきて，持ってきて，取って）
- ARマーカ認識によるグリッパ位置のビジュアルフィードバックを用いた高精度なアーム制御
- 但し，全体のシーケンスはステートマシンで人間が記述

記号創発ロボティクス

- 人間と機械のコミュニケーションを伴った協調動作のために、機械が言葉などの記号の真の意味を獲得することを目指す

CREST「記号創発ロボティクスによる人間機械コラボレーション基盤創設」

代表：電通大 長井隆行先生

国内最強チームのひとつ

Kara@ge++の指導者

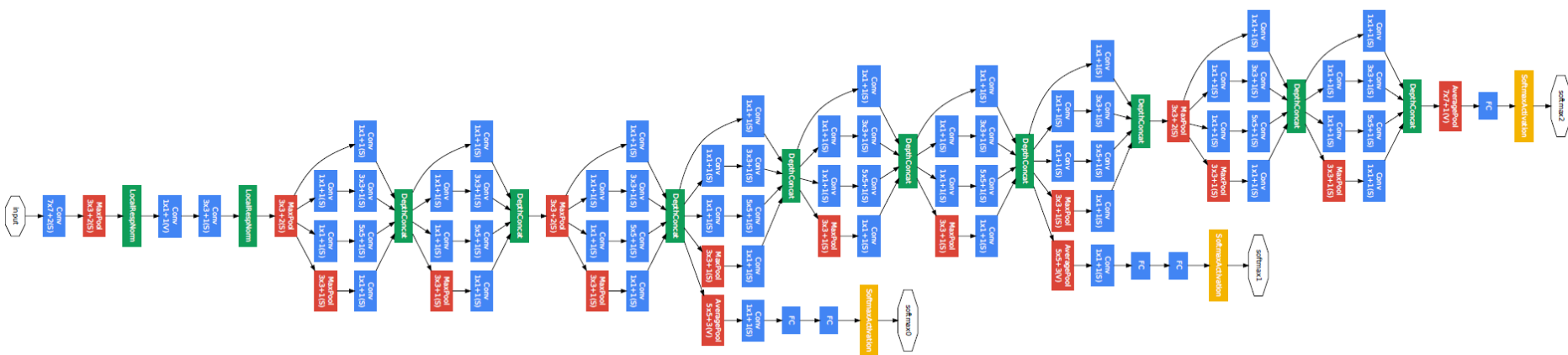
<http://sercrest.com/>



画像認識用DCNN

● GoogLeNet^[1] (ILSVRC 2014 優勝) & 転移学習

- ✓ 当初はLeNet (3層) を用いるも環境の変化に脆弱で実用に耐えない
- ✓ GoogLeNet であれば照明の変化等にも極めて頑健
- ✓ 学習済みのネットワークパラメタを転移学習し, 最終層のみを
@ホームロボット用にファインチューニング



GoogLeNet の構成 (22層) , Caffe^[2] model zooよりダウンロード可能

[1] C. Szegedy et al., “Going deeper with convolutions,” arXiv:1409.4842, 2014.

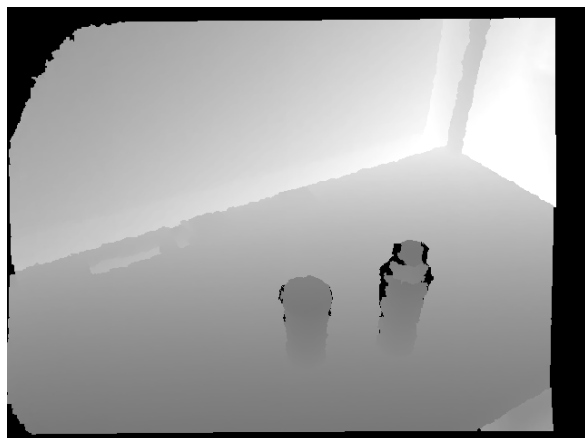
[2] Caffe <http://caffe.berkeleyvision.org/>



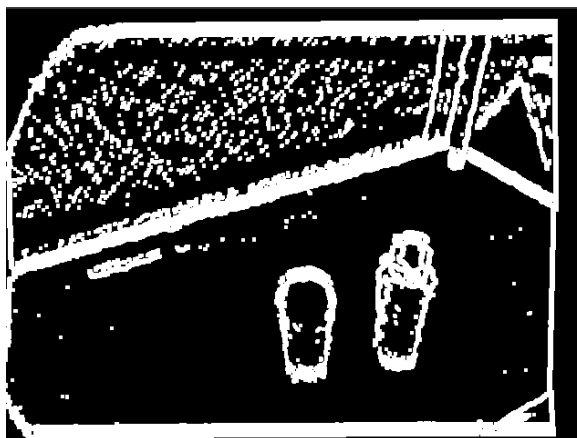
ホームロボット向け物体認識・ 把持システムの構築

石田裕太郎, 田中悠一朗, 森江隆, 田向権,
"ホームロボット向け物体認識・把持システムの構築,"
第34回日本ロボット学会学術講演会, 3G1-03, 2015年.

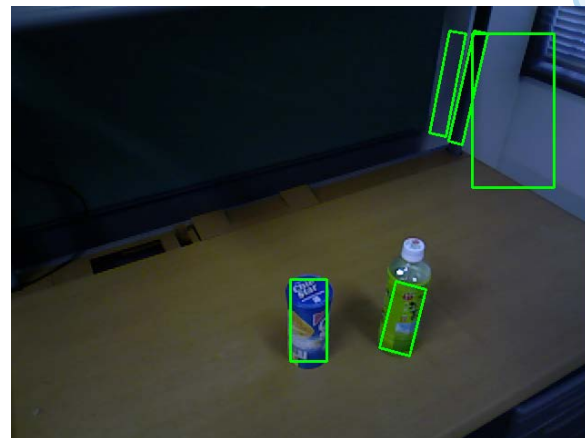
物体認識システムの流れ



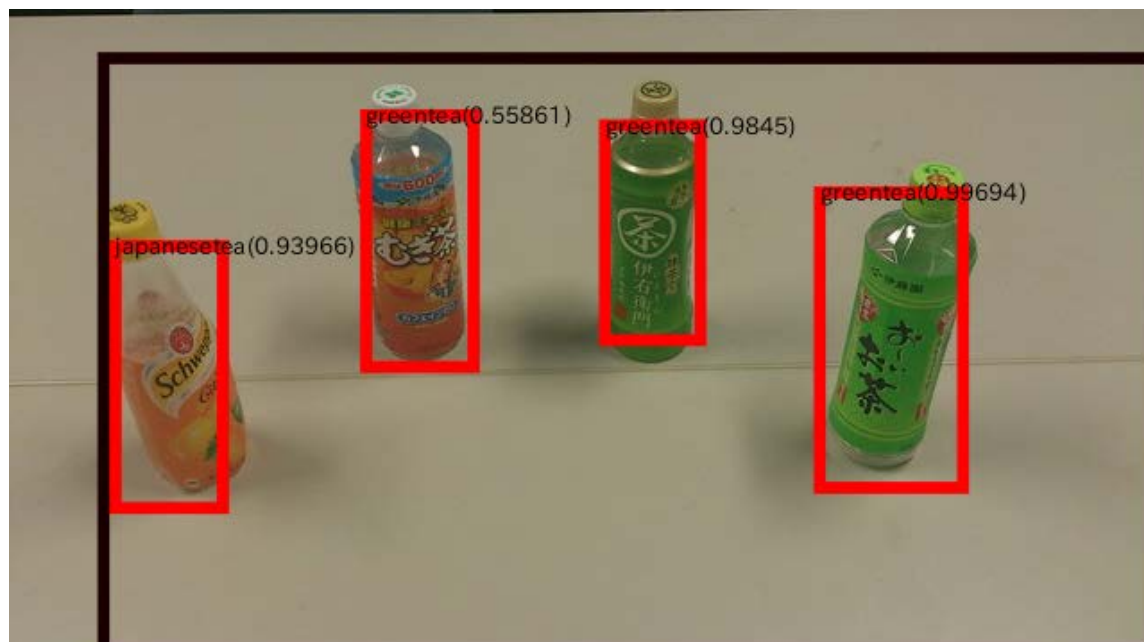
深度画像



フィルタ処理

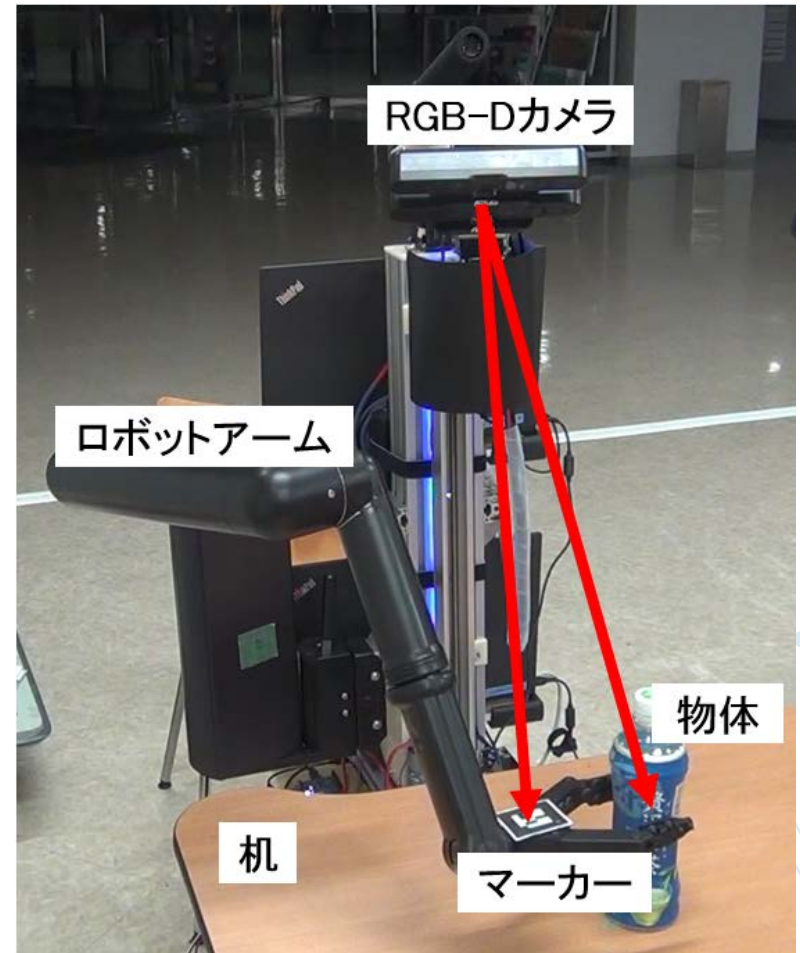
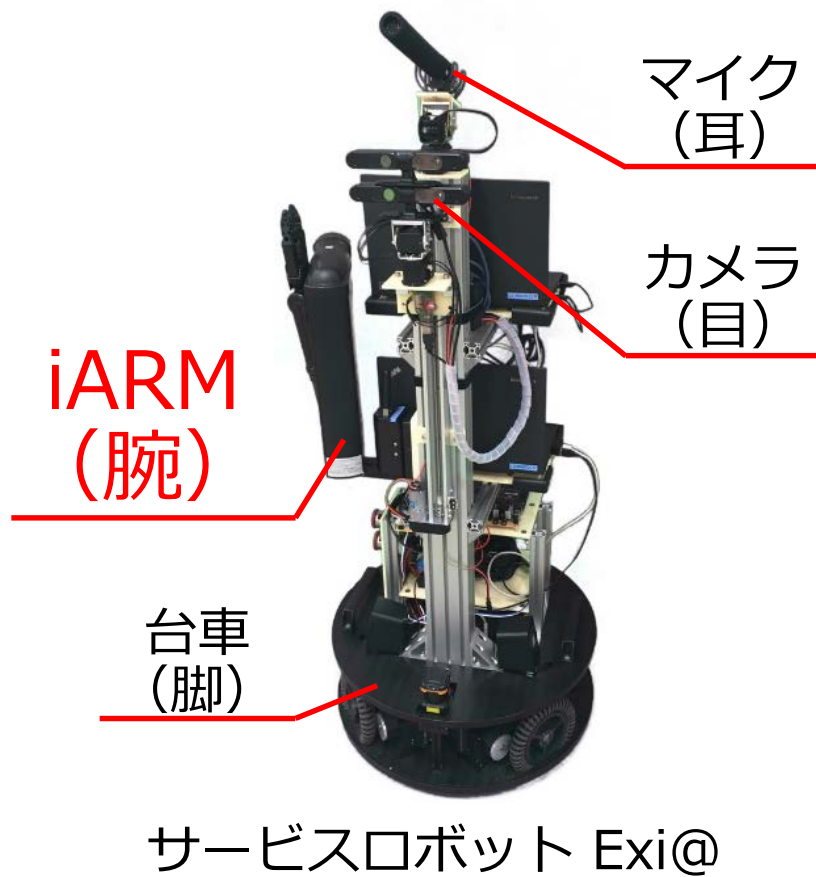


物体切り出し



Green Teaと認識

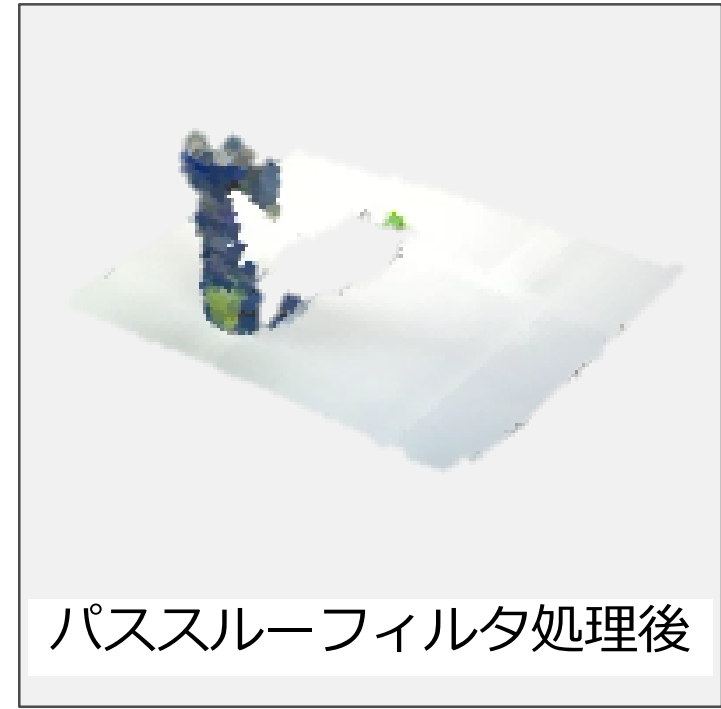
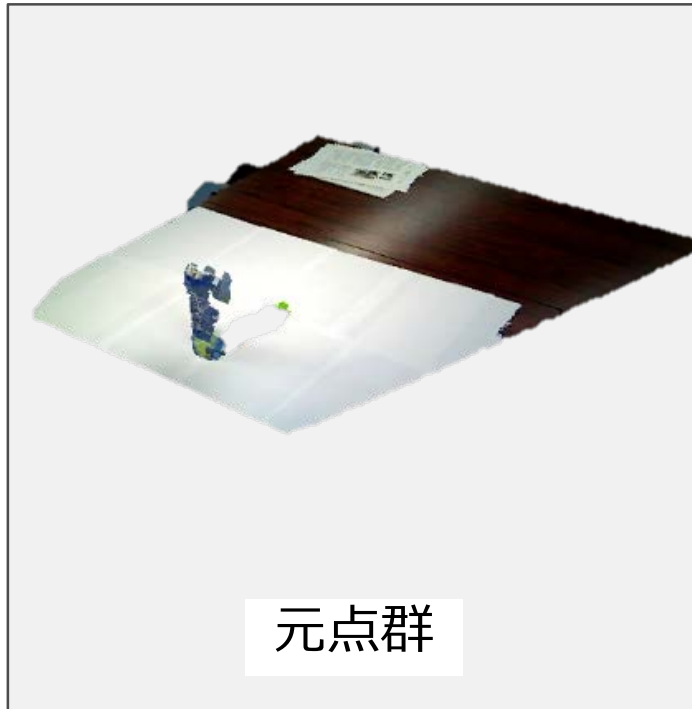
把持システムとの組み合わせ



一般的な家に存在する机の上に置かれた物体を認識・把持

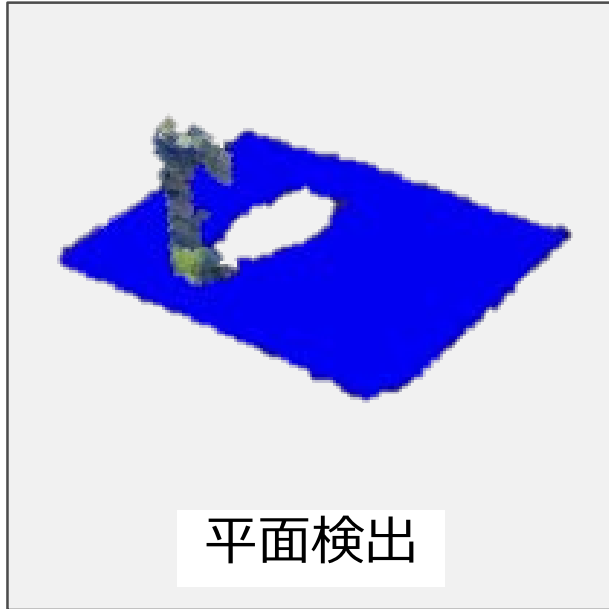
提案手法：物体検出（1）

RGB-Dカメラによって生成された3次元点群を用いる
PCL（Point Cloud Library）で処理



- ①パススルーフィルタで大まかに範囲を絞り込む
把持可能なところのみ注目，計算量削減

提案手法：物体検出（2）

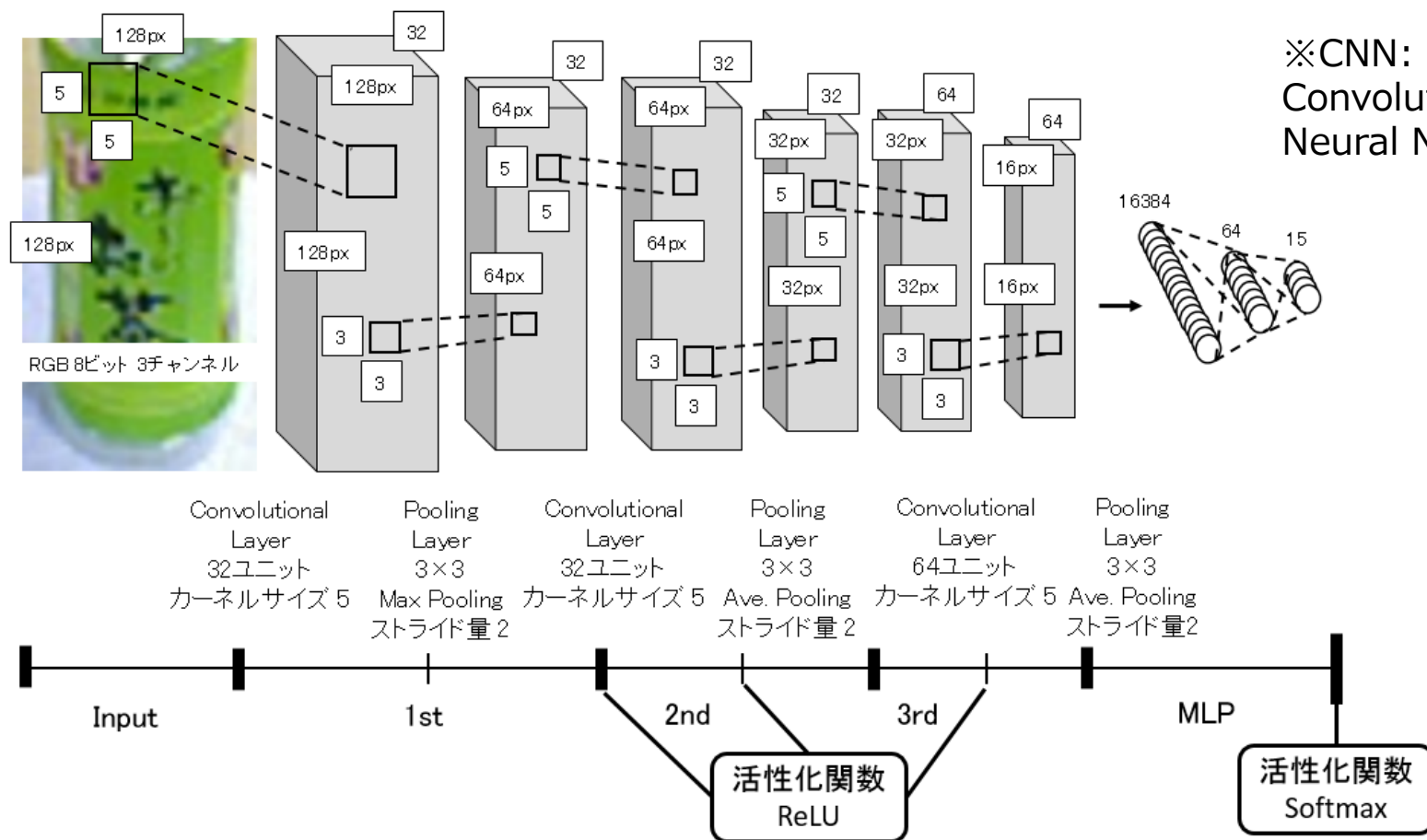


- ②RANSAC法を用いて平面検出・除去
- ③物体の3次元点群をもとに物体画像を切り出す

提案手法：物体認識（1）

Shallow CNN（畳みこみ層とプーリング層が3層，全結合層が3層）

※CNN:
Convolutional
Neural Networks

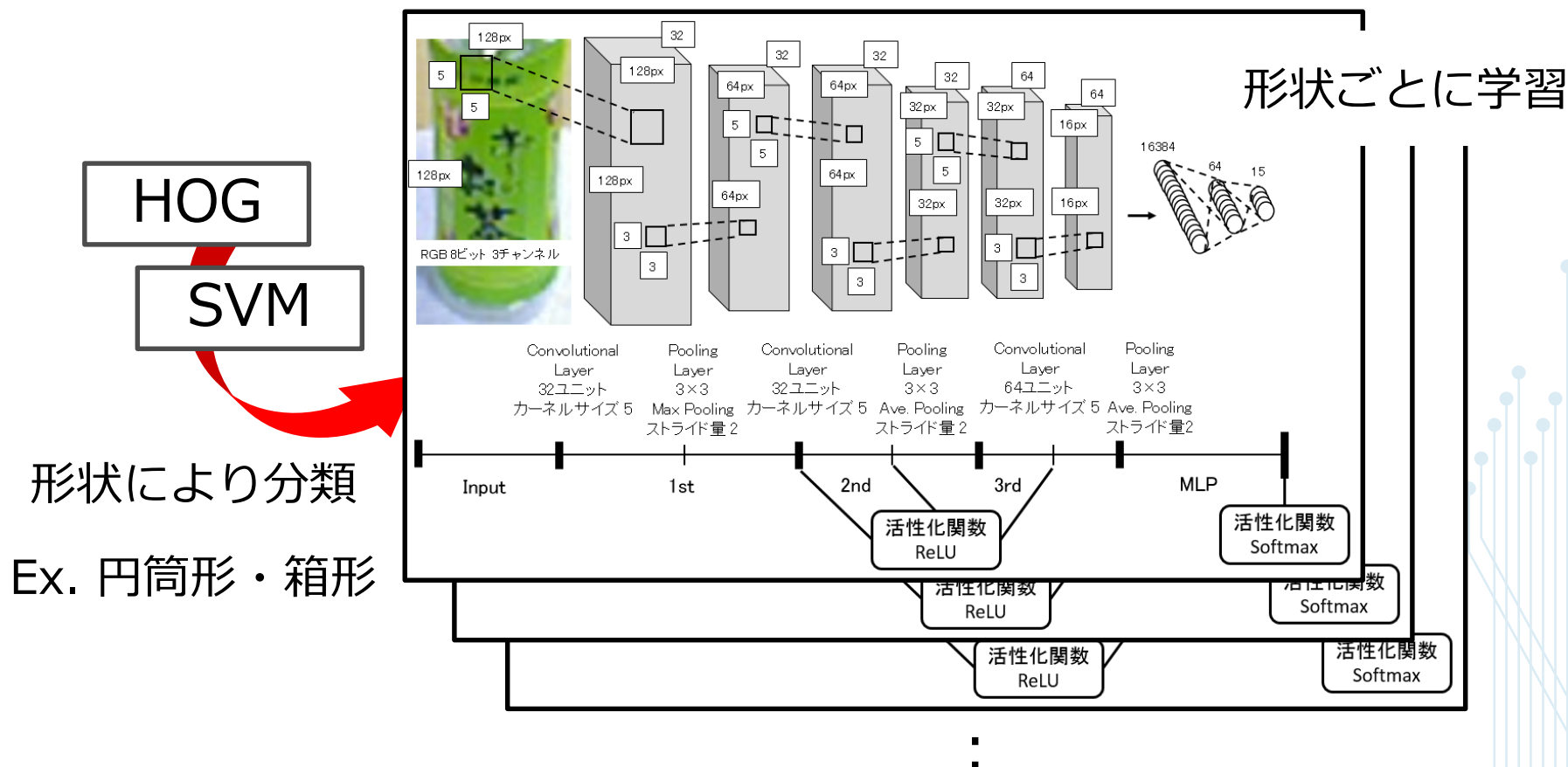


構造が浅く，入力から出力まで時間を要さない

提案手法：物体認識（2）

HOG + SVM + Shallow CNN

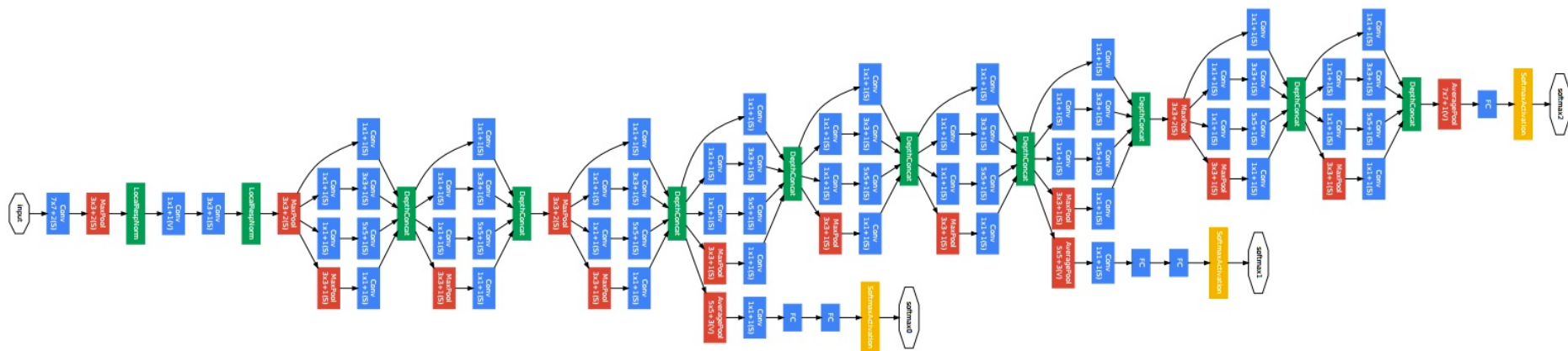
※HOG: Histograms of Oriented Gradients
※SVM: Support Vector Machine



1つのCNNで扱うクラス数を減らす→認識率の向上を狙う

提案手法: 物体認識 (3)

GoogLeNet[3] (22層のCNN)

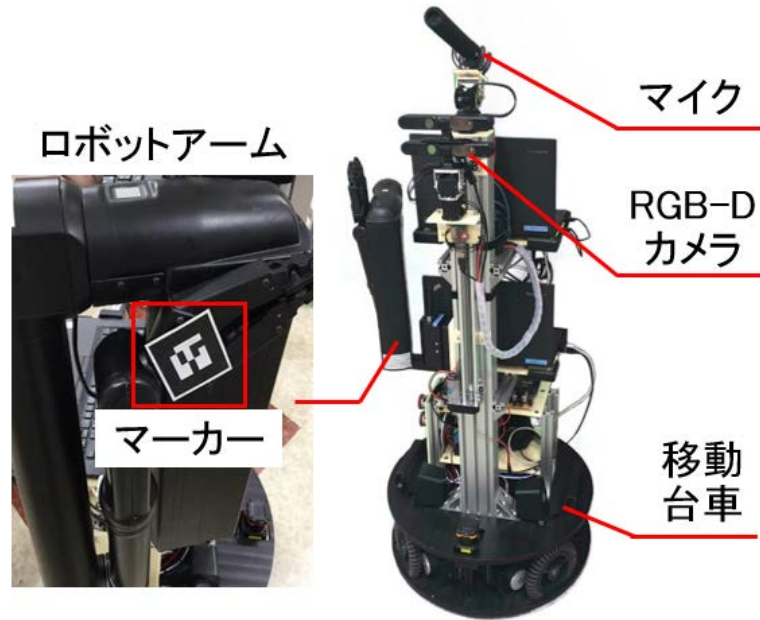


- ✓ ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [4]のデータセットにより学習済み
- ✓ 最終層のみを再学習 (転移学習)

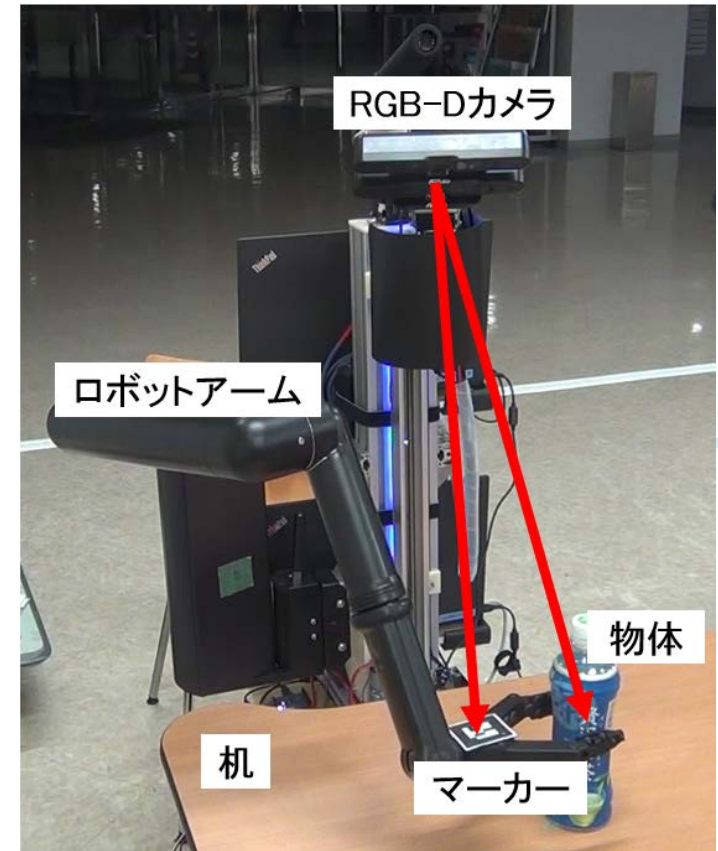
[3] "Model Zoo", http://dl.caffe.berkeleyvision.org/bvlc_googlenet.caffemodel

[4] "ImageNet Large Scale Visual Recognition Challenge 2012", <http://image-net.org/challenges/LSVRC/2012/>

提案手法：物体把持



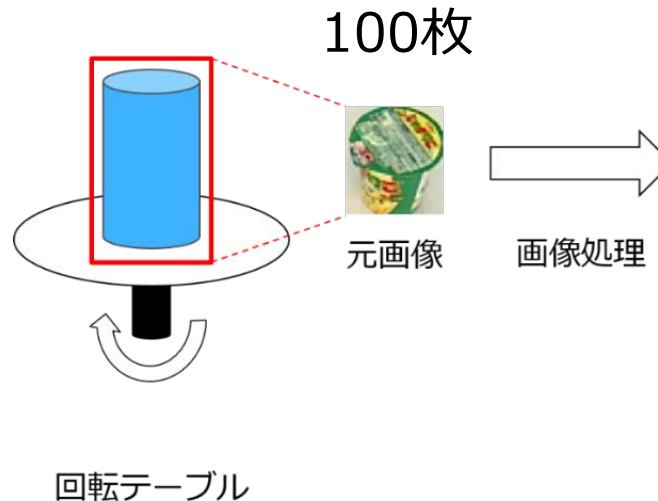
アームにマーカーを追加



マーカーを検出し目標座標に対して
誤差を最小化するフィードバック制御を行う

背景・物体に影響されない, エンドエフェクタの識別が可能

実験環境：データセット



27パターン



実験サンプル

1物体につき2700枚

- ✓ RoboCup@Home ジャパンオープン 2016の会場で15種の物体を撮影
(回転テーブルにより様々な角度から100枚ずつ)
- ✓ RGBそれぞれのチャンネルを0.9倍, 1.0倍, 1.1倍とした
27パターンのノイズを付加したデータセット

実験環境：学習と評価

- ✓ データセット2700枚中2000枚を学習
(Core i5-3470 DDR3 8GB GTX970)
- ✓ データセット撮影場所と異なる場所で物体1～12を評価
 - 光源などの環境が異なる
 - Core i5-5200U DDR3 12GB
- ✓ フレームワークとしてROS (Robot Operating System) とCaffeを使用

実験結果：物体認識

30度ずつ12方向から検証

方式		物体番号												正解率	処理速度
		1	2	3	4	5	6	7	8	9	10	11	12	[%]	[s]
Shallow CNN	正解数	8	10	1	10	0	0	10	12	3	12	11	8	59	0.2
	誤認数	4	2	11	2	12	12	2	0	9	0	1	4		
HOG + SVM + Shallow CNN	正解数	11	2	4	10	11	10	11	10	6	9	4	1	62	0.3
	誤認数	1	10	8	2	1	2	1	2	6	3	8	11		
GoogLeNet	正解数	12	12	12	11	12	12	12	12	12	12	12	12	99	4.6
	誤認数	0	0	0	1	0	0	0	0	0	0	0	0		

GoogLeNetは99%と頑健。しかし，処理時間が長い。

関連研究: ロボットアームのビジュアルフィードバック制御[1]

iARM

車椅子に取り付けユーザが操作することが前提
位置決め精度が悪い

手法

色空間ベースの画像処理により
エンドエフェクタ座標を算出

結果

フィードバックなしの把持成功率: 48%
フィードバックありの把持成功率: 72%

問題点

物体や背景の色によっては把持できない
SURF特徴量のマッチングでは物体認識に問題あり



(a) Projection of the space on the image



(b) Color matching pixels



(c) Space check



(d) Filtered ROI

実験結果：物体把持

各方式で
同じ場所・同じ物体を
30回ずつ把持する



	ビジュアルフィードバック制御なし	先行研究	提案手法
成功率[%]	48	72	93

21ポイント

45ポイント

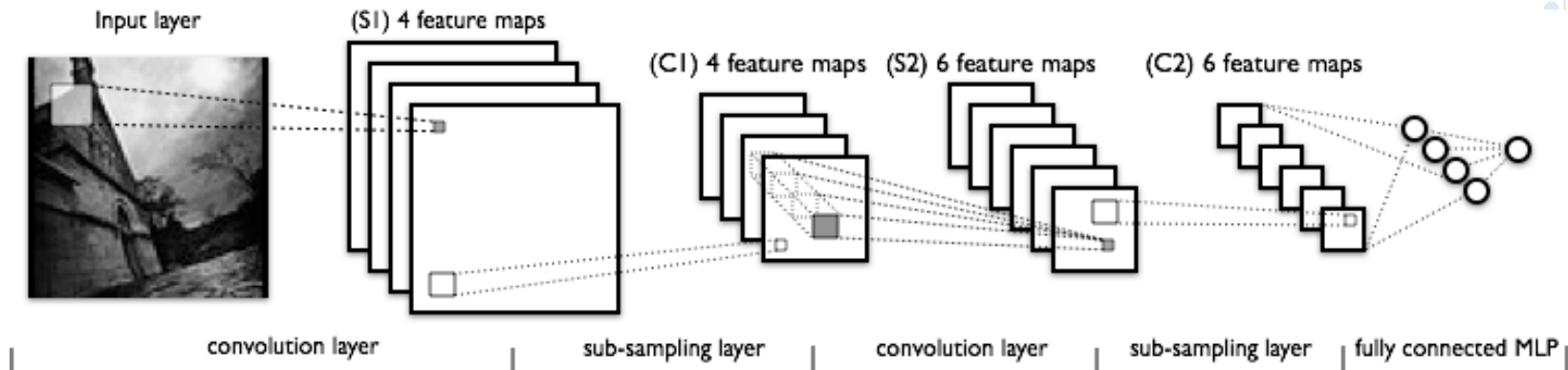
- ✓ マーカーベースにすることで物体や背景の影響を受けない
- ✓ 7ポイントは目標物体座標が誤って算出されたことにより失敗した



実践！ DCNNによる画像認識器の構築法

Convolutional Neural Networks (CNN) の構造

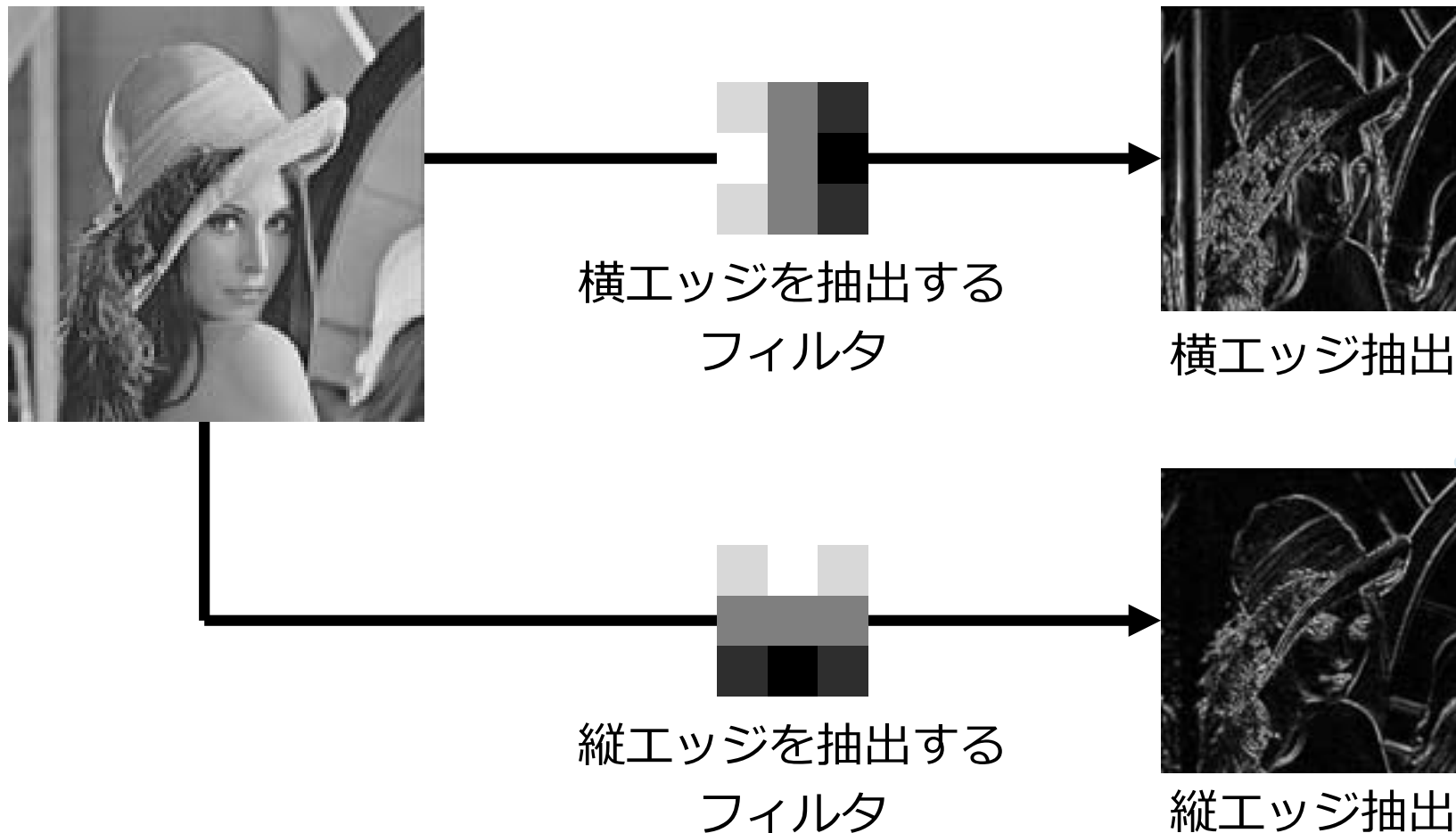
- 畳み込み + プーリング + 全結合層で構成
 - ✓ 畳み込み → フィルター処理
 - ✓ プーリング → 低解像度化
 - ✓ 全結合層 → MLP
- 下図はLecunが提案したLeNet



※LeCun, Proc. IEEE, 1990.

フィルタ処理とは

- 画像における特定の特徴をとりだす



CNNでは学習データから識別に有効なフィルタを学習

プーリングとは

- 画像を低解像度化し計算量を削減
- 2種類のプーリング
 - ✓ max pooling
 - ✓ average pooling



CNNの学習

- 誤差逆伝播法により
物体の識別に有効なフィルタを獲得する

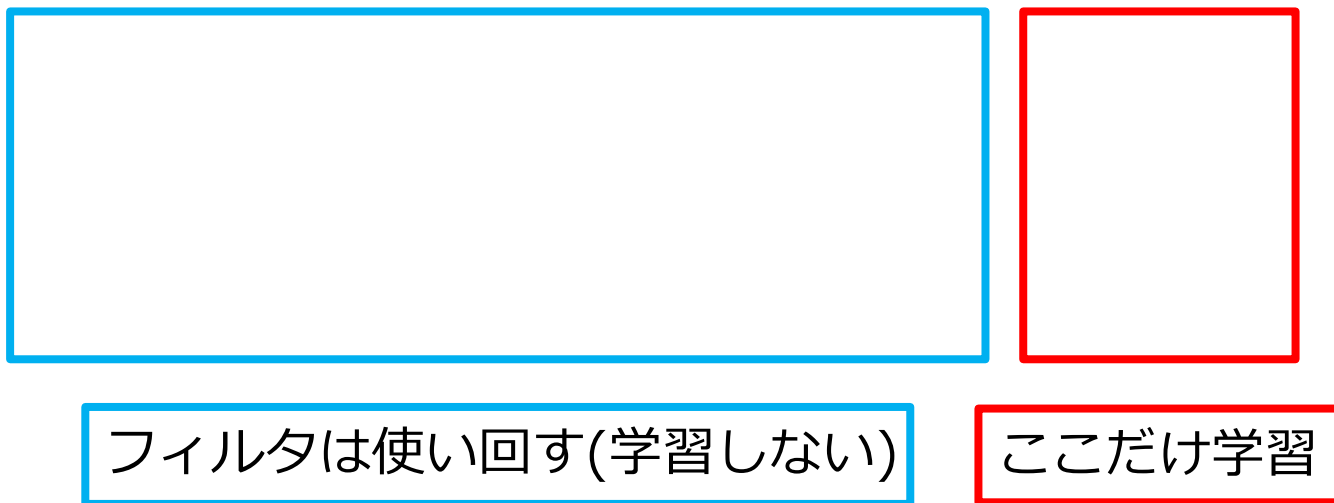
図引用元 : <https://devblogs.nvidia.com/parallelforall/accelerate-machine-learning-cudnn-deep-neural-network-library/>

H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng., "Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations." ICML 2009.

転移学習 (Transfer Learning)

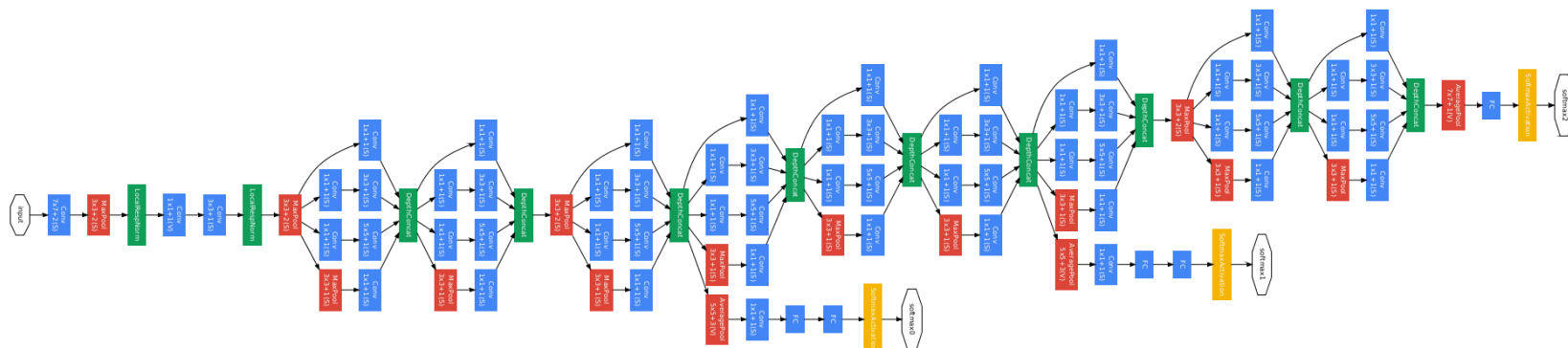
- あるタスクに特化した学習済みパラメータを読み込み
少しだけ改変することで他のタスクに特化させる
- ✓ 一から学習するより効率的かつ**高速**な学習が可能

転移学習の例



GoogLeNet※

- 2014年の画像認識コンテストILSVRCで優勝
- 22層から成るCNN→学習時間は膨大
 - ✓ prototxtが2000行を超えるほど
- 学習済みモデルの全結合層のみ更新
 - ✓ 2000クラス分類に対応できるほどのフィルタを持ち合わせている
 - ✓ 20クラス程度のペットボトル画像の分類は造作も無い



※Szegedy et. al., CVPR, 2015

Caffeとは

- Deep Convolutional Neural Network (DCNN) 用のフレームワーク
- 学習理論の理解やスクラッチからのプログラム実装が非常に困難なDCNNを手軽に取り扱える
 - ✓ OpenCV 感覚でDCNNを活用したアプリケーションを開発可能
- Caffe Model Zoo に 多数の実装済み&学習済みネットワークが存在, 自由にダウンロードして活用可能
 - ✓ グーグルのデータセンターでしか学習できないような巨大なネットワークも手軽に活用可能
- WEB上に多数の情報あり
 - ✓ LinuxとPythonが使える学生なら直ぐに実践可能
 - ✓ Linuxが苦手・・・ もっと簡単に使えるMATLABという選択肢もあります

Caffeの学習に必要なファイル

- データセット

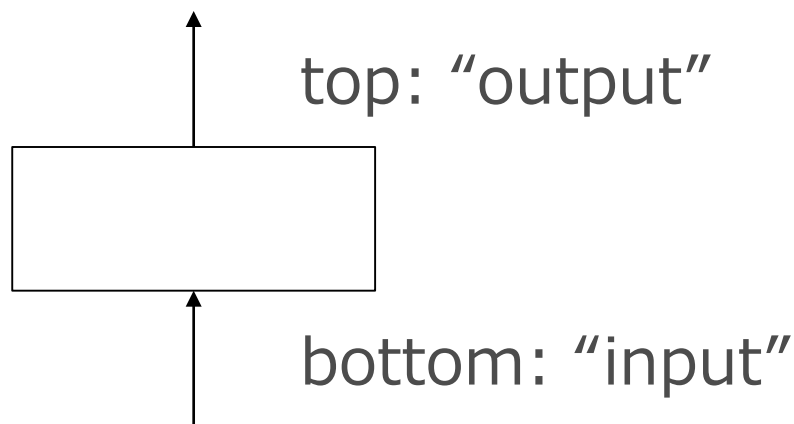
- ✓ 画像
- ✓ 画像のファイル名とラベル番号を定義するテキスト

- prototxt 2種類

- ✓ network_train.prototxt
ネットワークの構造を定義するファイル
- ✓ solver.prototxt
学習の設定を定義するファイル

構造の定義

- network_train.prototxtを編集
- レイヤーの種類はCaffeのTutorialページを参照
 - ✓ <http://caffe.berkeleyvision.org/tutorial/layers.html>
- layer{ ... }で各層を表現
 - ✓ bottomが入力で topが出力



prototxtの例：入力層(学習用)

```
layer {  
  name: "data"  
  type: "Data"  
  top: "data"  
  top: "label"  
  include{ phase: TRAIN }  
  transform_param {  
    scale: 0.0039215684  
  }  
  data_param{  
    source: "train_lmdb"  
    batch_size: 100  
    backend: LMDB  
  }  
}
```

#レイヤ名
#レイヤタイプ
#出力：学習データ
#出力：教師ラベル
#学習用と指定

#データの規格化($1 \div 255$)

#データベースのパス
#バッチサイズ
#データベースのタイプ

prototxtの例：入力層(テスト用)

```
layer{
  name: "data"
  type: "Data"
  top: "data"
  top: "label"
  include{ phase: TEST }
  transform_param {
    scale: 0.0039215684
  }
  data_param{
    source: "test_lmdb"
    batch_size: 100
    backend: LMDB
  }
}
```

#レイヤ名
#レイヤタイプ
#出力：学習データ
#出力：教師ラベル
#テスト用と指定

#データの規格化($1 \div 255$)

#データベースのパス
#バッチサイズ
#データベースのタイプ

prototxtの例：全結合層

```
layer{  
  name: "fullConnect1"           #レイヤ名  
  type: "InnerProduct"          #レイヤタイプ  
  bottom: "flatdata"            #入力：平滑化データ  
  top: "fc1"                    #出力  
  
  #重みの学習係数とdecay係数  
  param{ lr_mult: 1 decay_mult: 1 }  
  
  #バイアスの学習係数とdecay係数  
  param{ lr_mult: 1 decay_mult: 1 }  
  
  inner_product_param{  
    num_output: 10               #ユニット数  
  }  
}
```

prototxtの例：活性化関数と誤差関数

```
layer{  
  name: "loss"           #レイヤ名  
  type: "SoftmaxWithLoss" #レイヤタイプ  
  bottom: "fc1"          #入力：全結合層  
  bottom: "label"        #入力：教師ラベル  
  top: "loss"            #出力：誤差関数  
}
```

※SoftmaxWithLossレイヤーは
活性化関数：Softmax
誤差関数：クロスエントロピー
が複合されたレイヤーになる

GoogLeNetの入手方法

- caffe/models/bvlc_googlenet にある
 - ✓ deploy.prototxt : PyCaffeに読み込ませる
 - ✓ train_val.prototxt : 学習に使う
 - ✓ solver.prototxt : 学習の詳細設定
- caffemodelはサイズが大きいのので個別にDL
 - ✓ http://dl.caffe.berkeleyvision.org/bvlc_googlenet.caffemodel

実験サンプル

- RoboCup JAPAN OPEN 2016 AICHI @Homeリーグで使用
全15種類の物体, 各種類2,700枚の画像データセット
- ✓ リンゴ等のように形や色が一定でない物も含まれる



RoboCup JAPAN OPEN 2016 AICHI (<http://www.ait.ac.jp/robocup2016/>)

学習サンプルの取得法

- 回転テーブルを用いて様々な角度の物体を100枚撮影
 - ✓ 物体切出しには深度画像とPoint Cloud Library (PCL) を利用
- 1枚の画像にノイズを付加した27枚の画像を生成
 - ✓ 1クラスあたり2700枚の学習サンプル



データベース生成

- GoogLeNetは224x224の入力画像を想定している
データベースを作る際にはリサイズを忘れないこと

✓ createdb.sh 参照

✓ \$CAFFEROOT/build/tools/convert_imageset.bin

BOTTLE/

BOTTLE/train.txt

train_\$(DBTYPE)

1

-backend \$(DBTYPE)

-resize_height 224

-resize_width 224

Caffeの転移学習

- 学習させたくない層の学習係数を0にしておく
 - ✓ lr_mult: 0
 - ✓ 最後の全結合層(学習させたい層)の学習係数はそのまま
- 最後の全結合層のユニット数を変える
 - ✓ num_output: 15
- 学習済みパラメータを読み込んで学習を開始
全結合層のみが更新されていく
 - ✓ train.sh 参照
 - ✓ `$ ~/caffe/build/tools/caffe.bin --solver=solver.prototxt --weights=bvlc_googlenet.caffemodel`

各クラスの正解率 (2000枚:学習用, 700枚: テスト用)

ラベル名	正解率	ラベル名	正解率
Green tea	1.00	Potage soup	1.00
Café au lait	1.00	Egg soup	1.00
Iced tea	1.00	Orange	1.00
Orange juice	1.00	Apple	0.963
Strawberry juice	1.00	Bawl	1.00
Potato chips	1.00	Tray	1.00
Cookie	1.00	Cup	1.00
Potato stick	1.00	TOTAL	0.998

2016年9月現在のお勧めGPU

- 値段と性能のバランス良：GTX 1070：参考価格 64,800円
✓ もう少し予算が取れば， GTX 1080：参考価格 108,000円
- 私の研究室で最近複数台導入したGPUマシン構成例

合計 ¥

191,270 也(税込)

内 訳

品名	メーカー	品 名 / 規 格	数 量	単 価	金 額(税込)
SSD	サムスン	MZ-75E500B/IT (500GB)	1	16,980	16,980
SSDマウンタ	AINEX	HDM-34	1	570	570
CPU	intel	Core i7-6700K	1	42,000	42,000
マザーボード	ASUS	Z170-A	1	22,800	22,800
グラボ	MSI	GTX 1070 GAMING X 8G	1	64,800	64,800
メモリ	Crucial	CT2K8G4DFD8213(PC4-17000 8GB 2枚)	1	7,980	7,980
SATAケーブル	AINEX	AK-CBSA05-15BK	2	680	1,360
DVDドライブ	LG	GH24NSD1 BL	1	2,600	2,600
電源	COUGAR	HEC-GX800V2 (800W)	1	12,800	12,800
PCケース	クーラーマスター	Silencio 452 SIL-452-KKN1-JP	1	10,980	10,980
CPUファン	サイズ	SCKBT-3000 (KABUTO3)	1	5,300	5,300
マウス・キーボード		Wireless Combo MK270	1	3,100	3,100

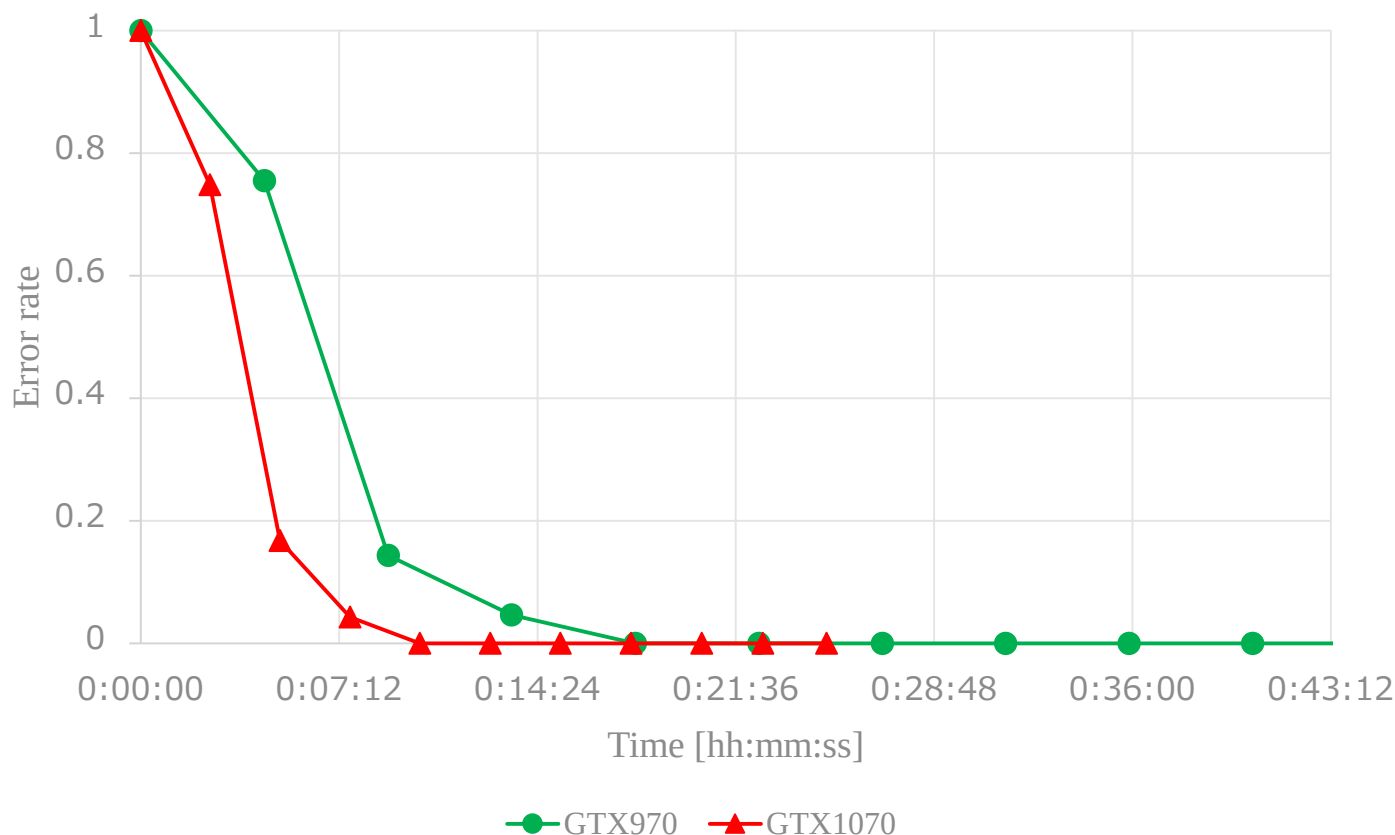
GoogLeNet最終層のファインチューニングに必要な時間

- エラー率0に落ちた時間（RoboCup@Homeの物体画像）

- ✓ GTX970 : 約18分

1世代前のGPUで1070と同価格帯

- ✓ GTX1070 : 約10分



研究紹介

エッジ指向型脳型計算機システム の実現に向けて

エッジ指向型脳型計算機システムに向けて

- 深層学習を中心とする人工知能を全ての“モノ”へと実装するために、エッジ指向型脳型計算機システムの構築と応用を目指す。特に、ハードウェア化に適した脳型人工知能アルゴリズムと書き換え可能半導体FPGAと組み合わせて用いる仮想回路の研究開発に注力する。



DCNNによる物体認識精度と処理時間（再掲）

● 順伝搬時（実行モード）

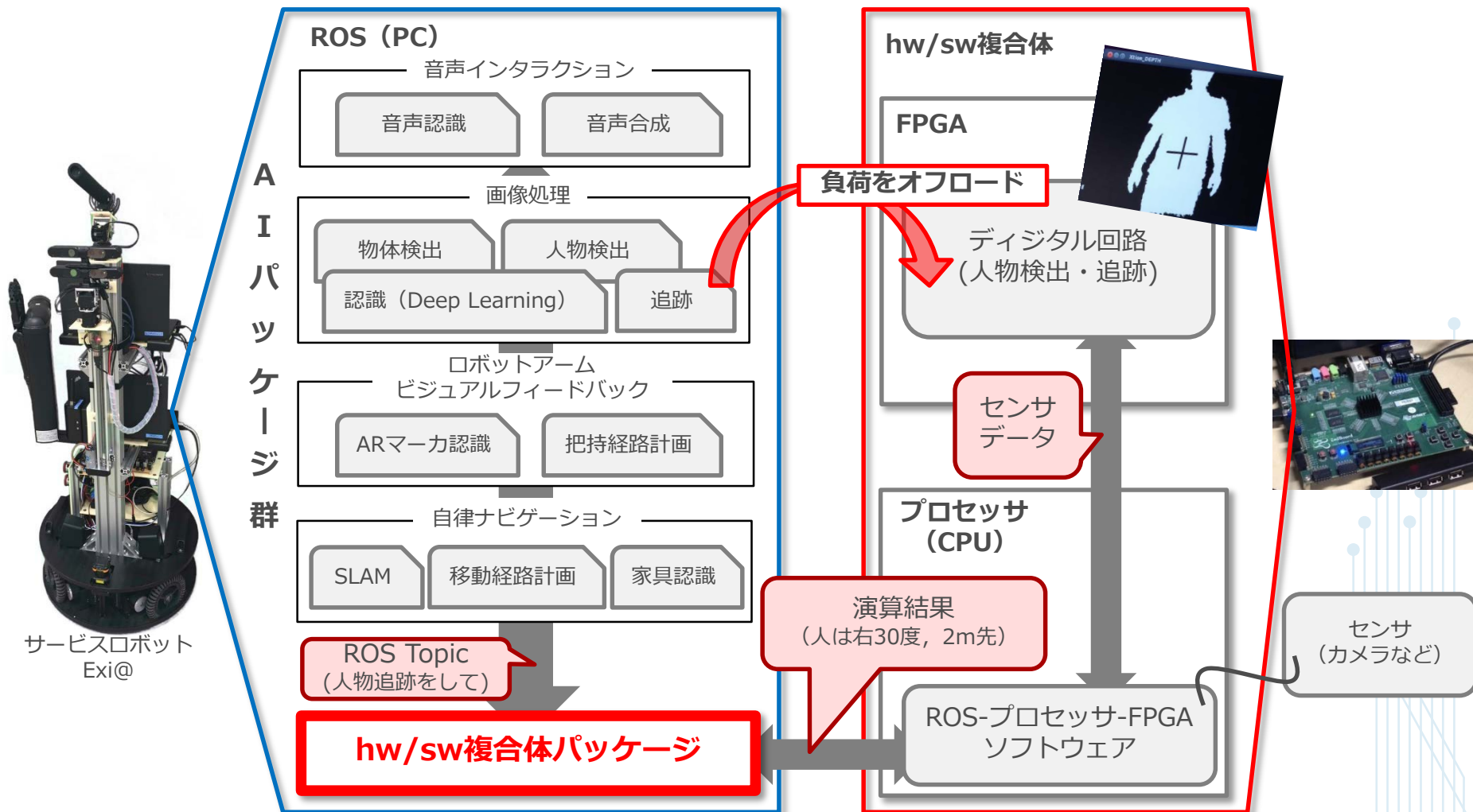
方式		物体番号												正解率	処理速度
		1	2	3	4	5	6	7	8	9	10	11	12	[%]	[s]
Shallow CNN	正解数	8	10	1	10	0	0	10	12	3	12	11	8	59	0.2
	誤認数	4	2	11	2	12	12	2	0	9	0	1	4		
HOG + SVM + Shallow CNN	正解数	11	2	4	10	11	10	11	10	6	9	4	1	62	0.3
	誤認数	1	10	8	2	1	2	1	2	6	3	8	11		
GoogLeNet	正解数	12	12	12	11	12	12	12	12	12	12	12	12	99	4.6
	誤認数	0	0	0	1	0	0	0	0	0	0	0	0		

GoogLeNetは99%と頑健。しかし、処理時間が長い。

コンピュータシステムとLSIに求めること

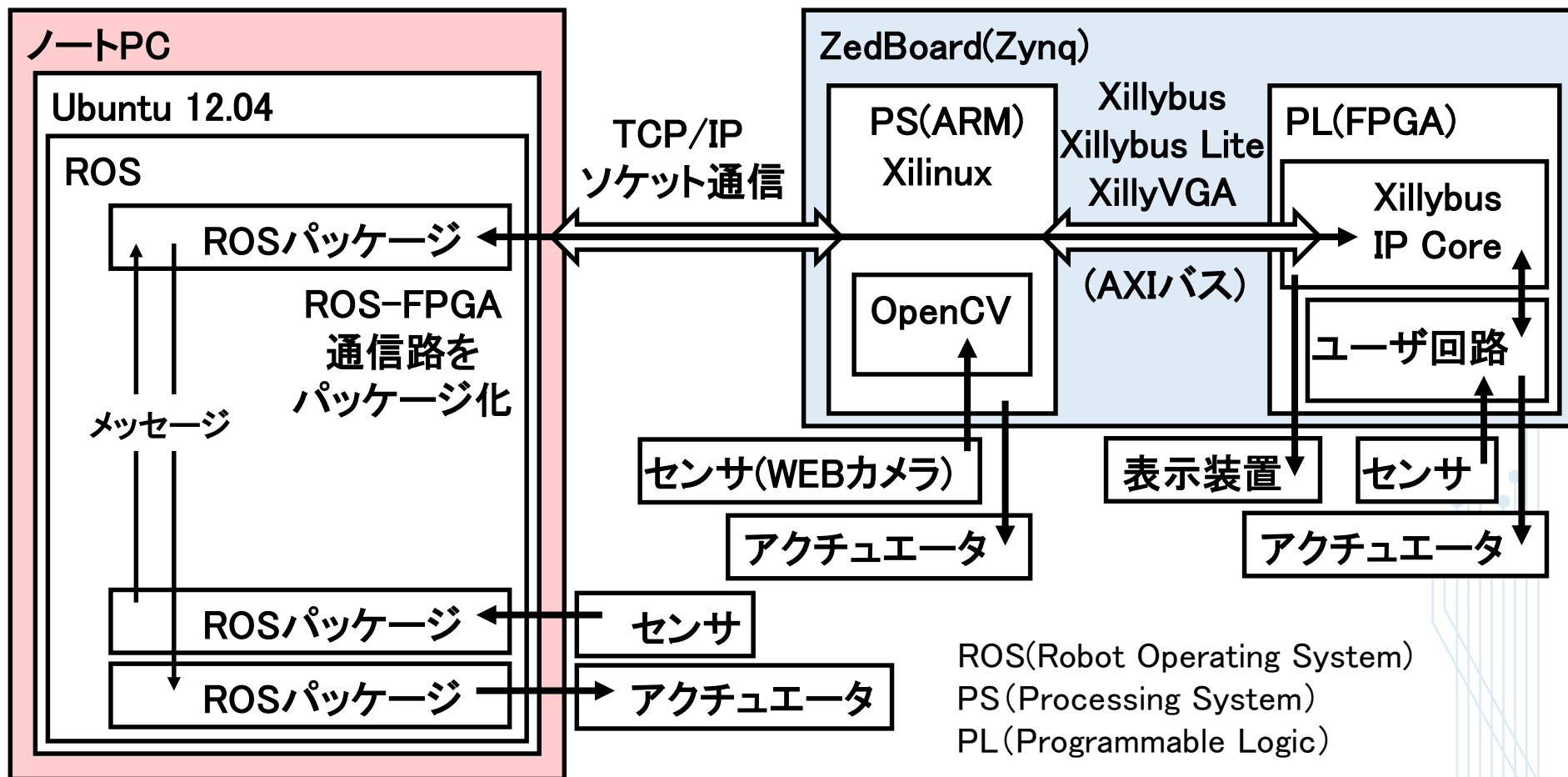
- 常時10以上のROSトピックが並列動作
 - ✓ ノートPC搭載の4コアCPUのパワーは常に不足
 - ✓ ステートマシンで起動するトピックを切り替えつつやりくり
 - ✓ 車載ネットワークのように多数のCPUをROSでネットワーク化
- DCNNや各種画像処理には高速化が必須
 - ✓ Point Cloud Libraryによる深度画像からの物体切出し, DCNNによる物体認識, パーティクルフィルタによる人物追跡など画像処理を多用
 - ✓ GPUは必須, 但し消費電力に目をつぶっても組込みには熱が凄い
- 小型化, 高速化, 省電力化が必須, 同時に柔軟性も必要
 - ✓ 各種知的処理のLSI化 (但しアルゴリズムは日進月歩) とROS対応
 - ✓ 実行モードだけでもDNNの専用チップが欲しい

ROS-FPGA 連携システム：コンセプト



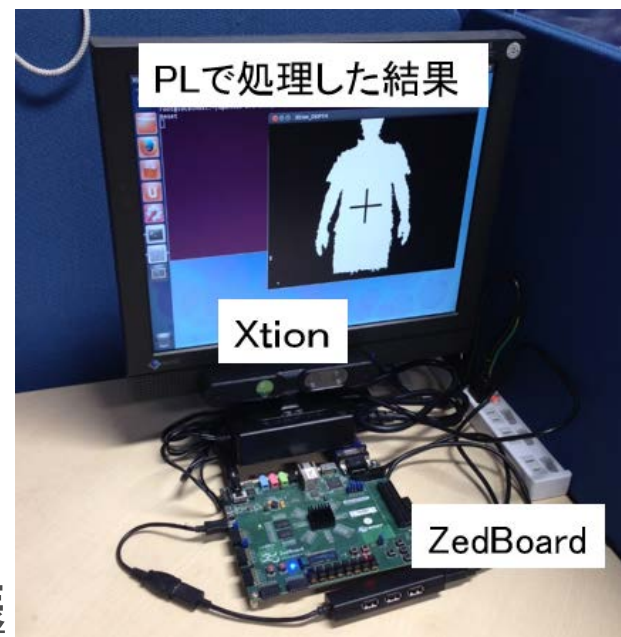
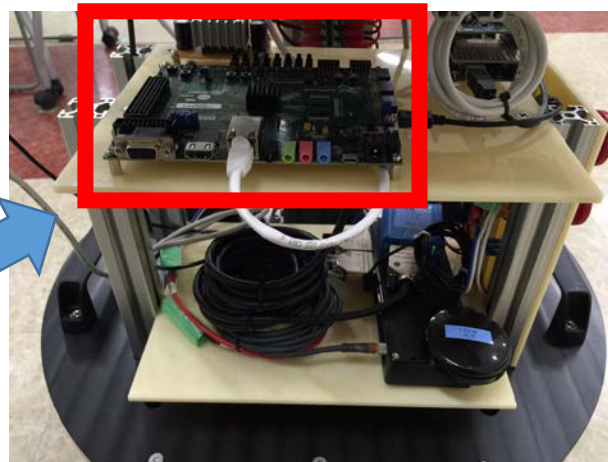
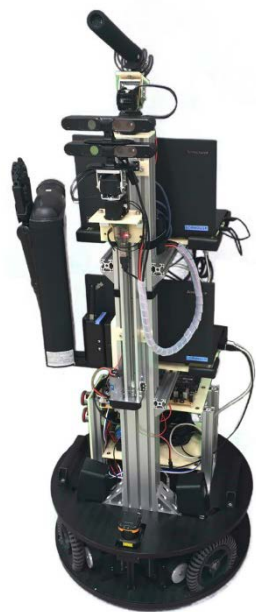
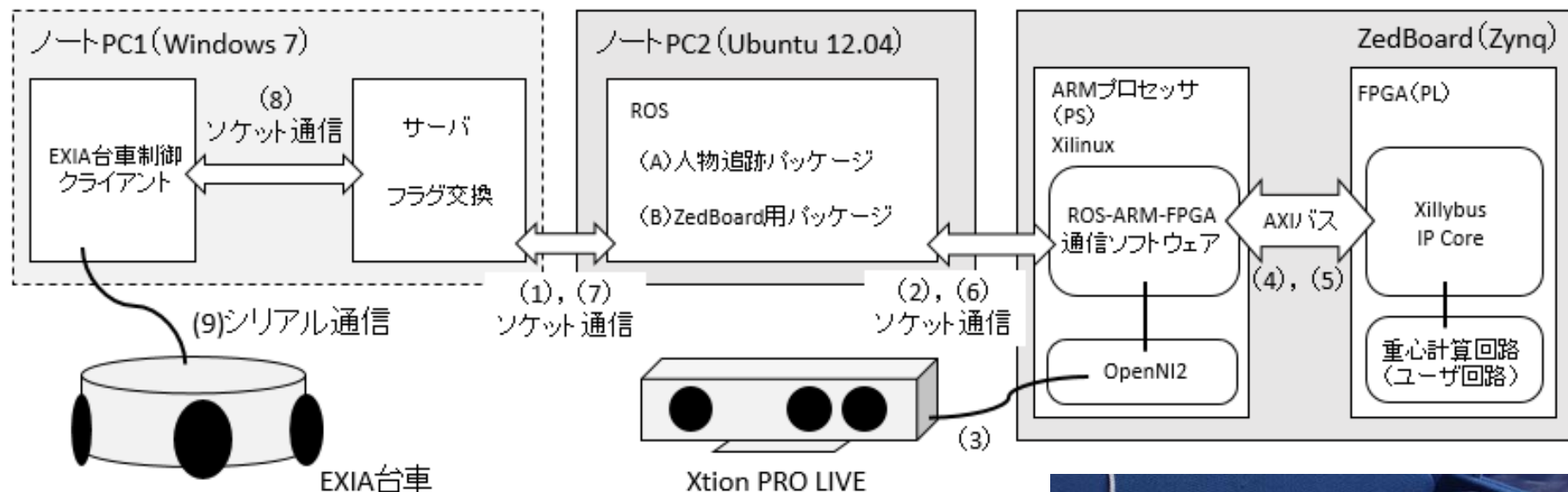
高負荷な知的処理をFPGAへオフロード，ROSで簡便にアクセス

ROS-FPGA 連携システム：ブロック図



**FPGA内部に構成された知的処理回路はROSノードにぶら下がる
ひとつのパッケージとして簡便に取り扱える**

Exi@への実装例



人物追跡用画像処理回路を実装

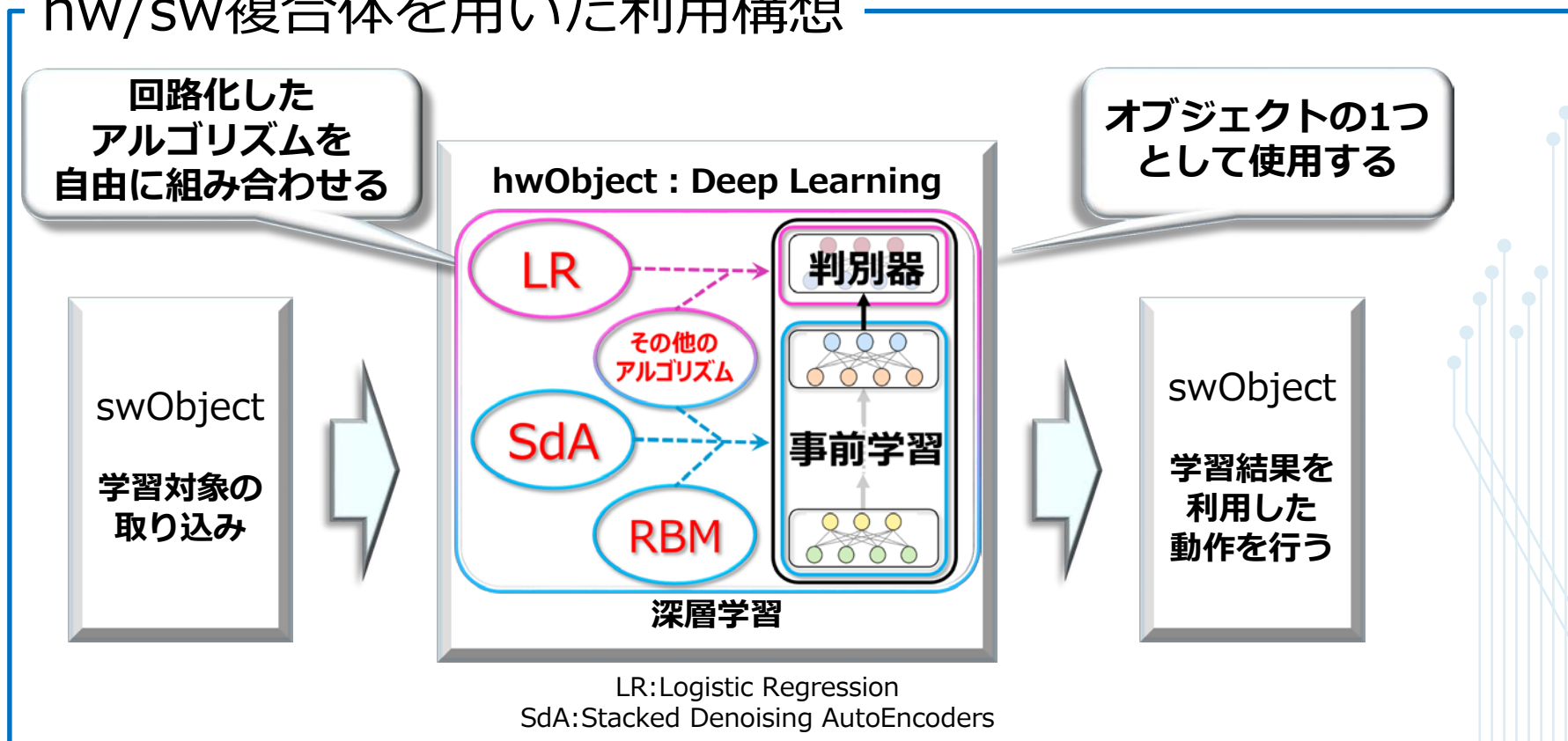
石田 裕太郎, 他, "FPGAによるROS向け高速分散処理システムの実装,"
第60回システム制御情報学会研究発表講演会 (SCI'16), 2016年5月.

Deep Neural Networks の基本回路

- 基本回路 + ハードウェアオブジェクトによる拡張

✓ 回路パラメタで規模変更, 基本回路 (オブジェクト) を組合せて階層化

hw/sw複合体を用いた利用構想



現在取り組んでいるDNNの基本回路

- Restricted Boltzmann Machines
- AutoEncoders
- Chaotic Boltzmann Machines
- Deep Self-Organizing Maps Networks and Advance Propagation
- 上記以外に, スパイクパルスタイミング演算に着目した Pulse-coupled phase oscillator networks

AutoencodersベースのDNNの場合



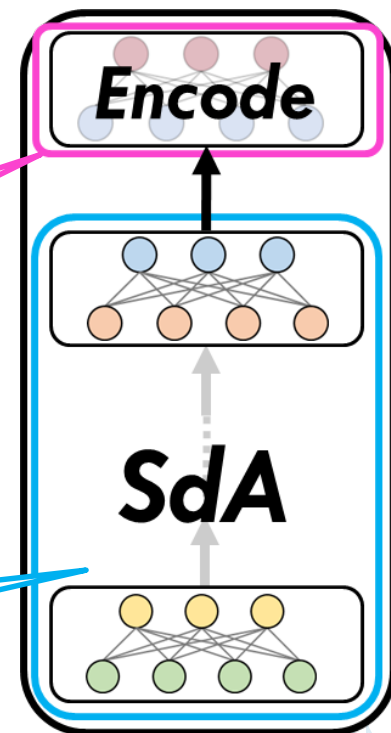
AutoEncoder Denoising AutoEncoder

Stacked (Denoising) AutoEncoder

深層学習を構成

判別器 : Encodeのみ適用
(Logistic Regressionと等価)

事前学習器 : Stackしたもの



Shared Synapse Architecture によるAEの回路化

FPGAの資源は有限

ディジタル回路設計として…

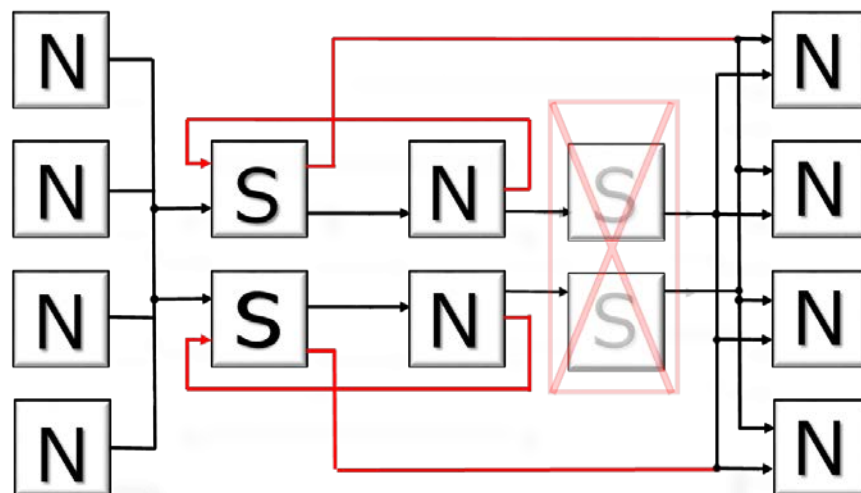
SharedSynapse構造

複数データの再構成
→重み行列の転置

資源の節約

更新値算出時間の短縮

更新値書き込み時間の短縮

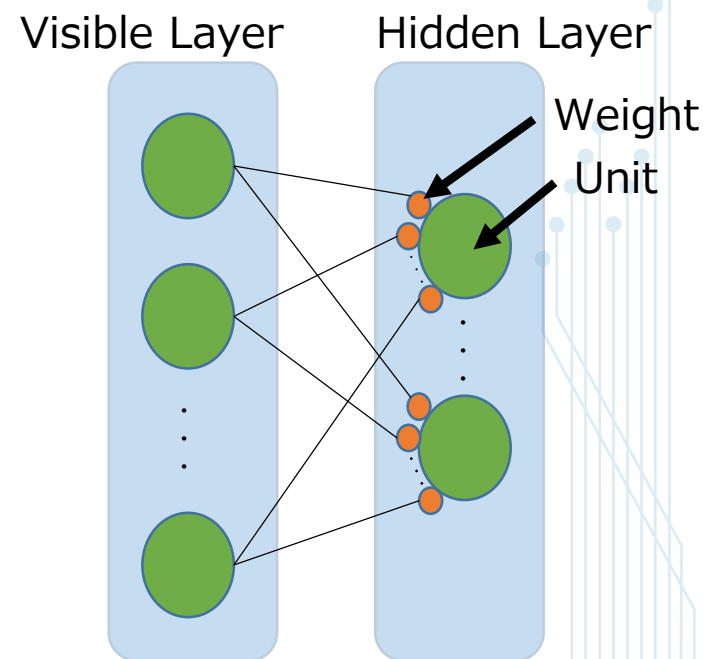


Restricted Boltzmann Machines (RBM)

G.E.Hinton, et al., Neural Computation, 2006

Deep Neural Networks を構成する
ニューラルネットワークの一つ。

- 可視層と隠れ層を持ち，同じ層内のユニット同士は結合しない
- 生成モデルの一つで，データが生成される確率分布を学習する
- 確率的な動作を行う



RBMの学習

Asja Fischer, Christian Igel, Lecture Notes in Computer Science, 2012.

隠れ層と可視層ユニットの発火確率

$$P(h_j = 1 | \mathbf{v}) = \sigma \left(b_j + \sum_i w_{ij} v_i \right)$$

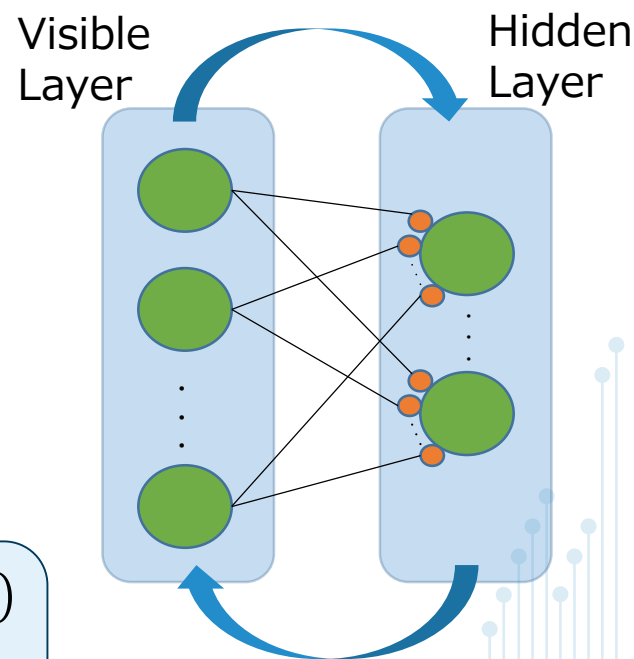
$$P(v_i = 1 | \mathbf{h}) = \sigma \left(a_i + \sum_j w_{ij} h_j \right)$$

重みとバイアスの更新式

$$W \leftarrow W + \eta (P(h^{(0)} = 1 | v^{(0)}) v^{(0)} - P(h^{(k)} = 1 | v^{(k)}) v^{(k)})$$

$$b \leftarrow b + \eta (v^{(0)} - v^{(k)})$$

$$c \leftarrow c + \eta (P(h^{(0)} = 1 | v^{(0)}) - P(h^{(k)} = 1 | v^{(k)}))$$



$$\sigma(x) = \frac{1}{1 + e^{-\frac{x}{T}}}$$

T : 温度, $P(h_j = 1 | \mathbf{v})$: 隠れ層のニューロン発火確率, $P(v_i = 1 | \mathbf{h})$: 可視層のニューロン発火確率,
 b_j : 隠れ層ニューロンのバイアス, a_i : 可視層のニューロンのバイアス, w_{ij} : 重み, v_i : 可視層ニューロンの状態,
 h_j : 隠れ層ニューロンの状態, w : 重み, η : 学習係数, b : 可視層バイアス, c : 隠れ層バイアス

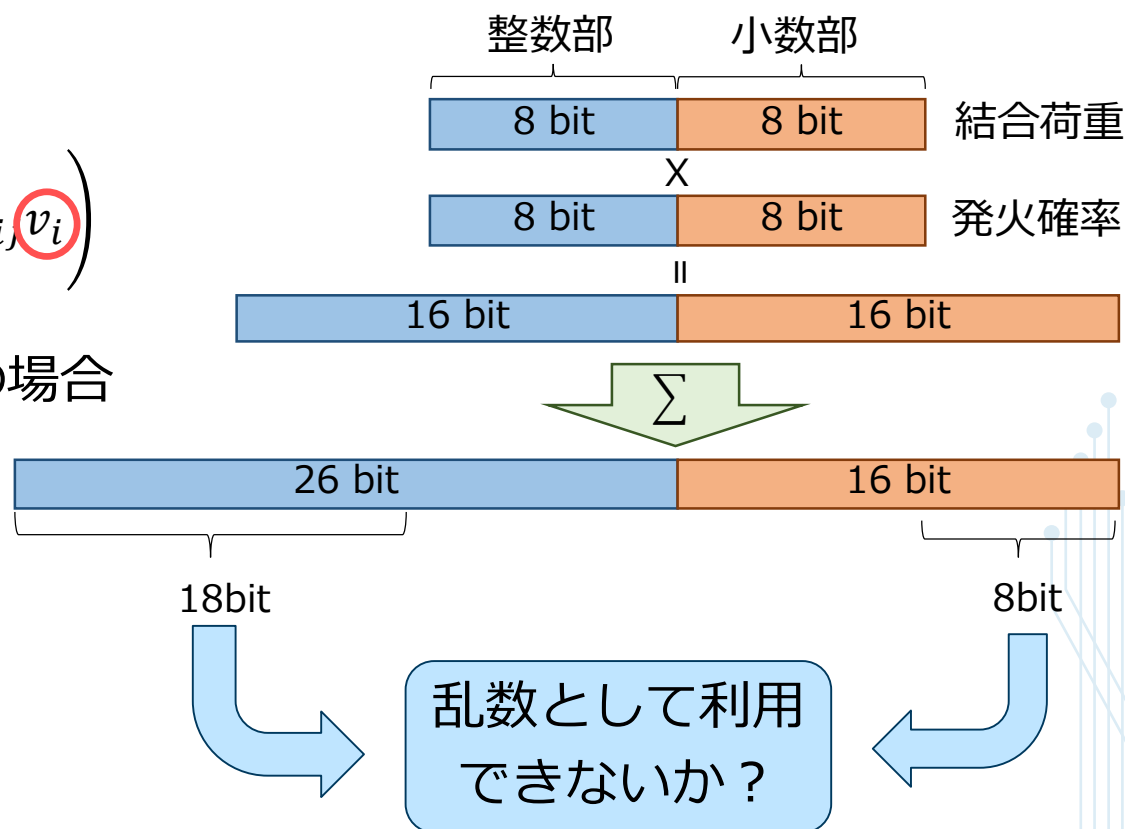
可視層, 隠れ層ユニットを交互に計算し更新.

乱数生成器を用いないRBM

発火確率算出時の切り捨てビットを乱数の代わりとして利用

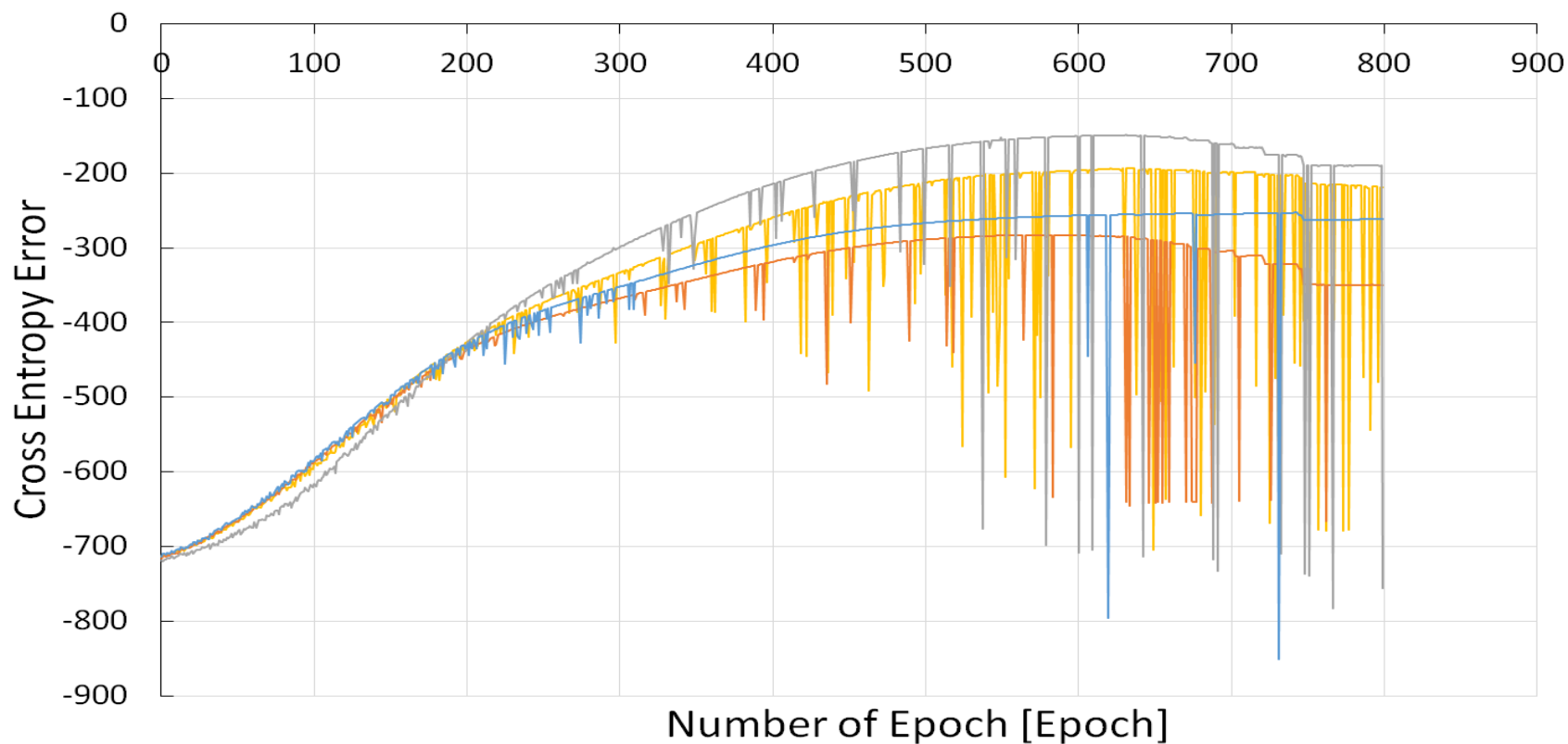
$$P(h_j = 1 | \mathbf{v}) = \sigma \left(b_j + \sum_i w_{ij} v_i \right)$$

可視層ユニット1024個の場合



回路化の際に生じる演算誤差を積極的に活用する

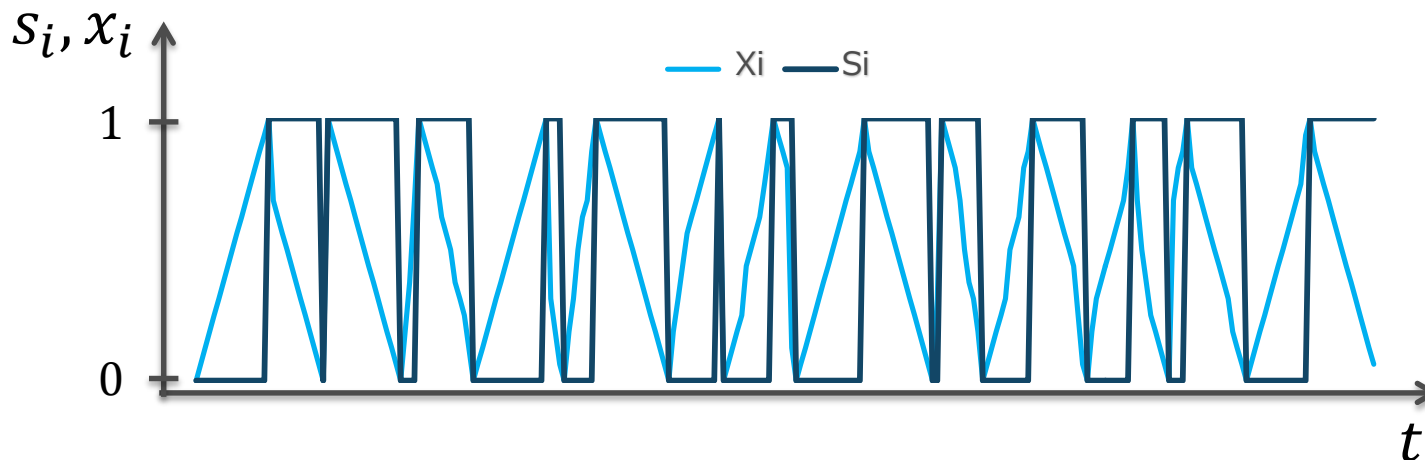
学習結果 - 切り捨てビットを乱数として利用



- 1: すべてソフトウェアの乱数
- 2: 学習初期のみソフトウェアの乱数
- 3: 学習初期のみ非切り捨てビットの8bitを使用
- 4: 学習初期のみ非切り捨てビットの4bitを使用

カオスボルツマンマシン[1]

[1] H. Suzuki, J. Imura, Y. Horio and K. Aihara,
"Chaotic Boltzmann Machines,"
Scientific Reports, 2013.



- ボルツマンマシンの確率的挙動をカオスダイナミクスで決定的に実現 $\Rightarrow$ 乱数生成不要

- 内部状態 $x_i \in [0, 1]$ の変化量

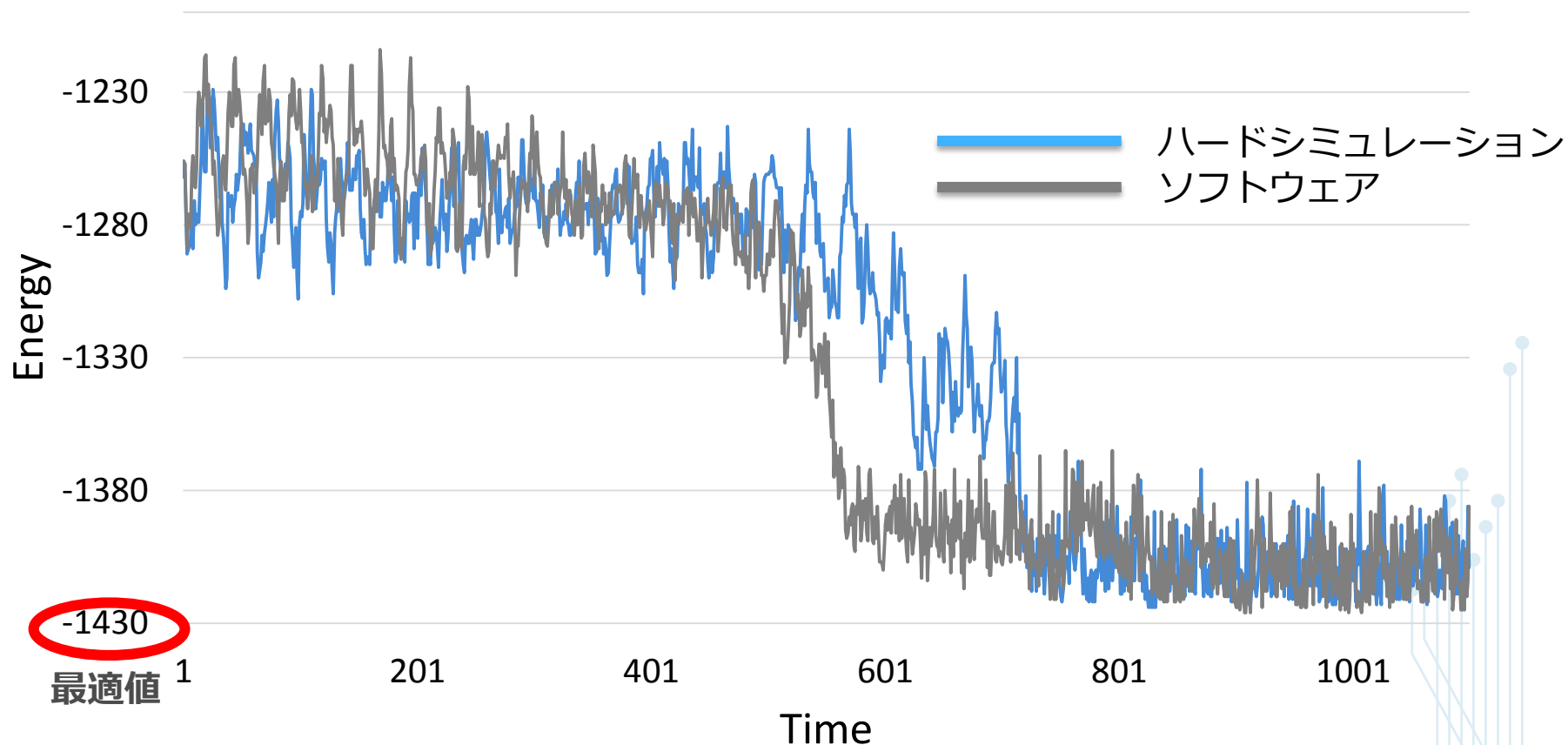
$$\checkmark \frac{dx_i}{dt} = (1 - 2s_i) \left(1 + \exp \frac{(1-2s_i)z_i}{T} \right) \Rightarrow \text{振動子回路と挙動が類似}$$

- ニューロンの発火状態

$$\checkmark \begin{cases} s_i := 0 & \text{when } x_i = 0 \\ s_i := 1 & \text{when } x_i = 1 \end{cases}$$

s_i : ニューロン i の状態,
 z_i : ニューロン i への入力
 T : 温度,

100ニューロンCBMの最大カット問題への適用例^[1]



- ハードウェア化に際し振動子回路を導入すると共に、固定小数点化や演算簡略化を行った
- 双方とも最適解が得られた

組込指向型DNNの世界的動向

- BinaryConnectを皮切りに組込指向型DNNの発表多数
 - ✓ BinaryNet, SqueezeNet などなど
 - ✓ IBM TrueNorthがBinaryNetをベースにDCNNを実装（2016年4月）
 - ✓ arXivにどんどん投稿, Githubでソースコード共有, TheanoやCaffeで手元ですぐに追実験可能, まさに日進月歩
- CNNも単にDeep化するのではなく, 新しい様々な手法が提案, パラメタ数を減らしつつ性能を向上させている
 - ✓ Inception Layer（畳みこみ層の分岐, GoogLeNetに導入）
 - ✓ Global Average Pooling（パラメタ数が爆発する全結合層の代わり）
 - ✓ ResNet（ショートカットとボトルネックの導入, 152層もの深層化を達成しつつ, AlexNet（8層）よりパラメータ数小かつ超高性能）

BinaryConnect (2015年11月)

- Bengio先生のグループよりBinaryConnectが発表 (NIPS2015, arXivに11月にアップロード)

✓ 2値化の条件は2つ

① 決定的

乗算 0

$$w_b = \begin{cases} +1 & \text{if } w \geq 0 \\ -1 & \text{otherwise.} \end{cases}$$

② 確率的

高精度

$$w_b = \begin{cases} +1 & \text{with probability } p = \sigma(w) \\ -1 & \text{with probability } 1 - p \end{cases}$$

$\sigma(x) = \max(0, \min(1, \frac{x+1}{2}))$: ハードシグモイド関数

- 決定的手法の場合, 乗算は完全にゼロ (実行時) になる
 - ✓ 学習時は正確な重みを保持する必要あり, 乗算は完全には無くならない
- CNN, MLP共に効果あり

SqueezeNet (2016年4月)

- CNNモデルの改良（ 1×1 のフィルタが重要） ,
枝刈（1980年代の手法のリバイバル+ α ） ,
圧縮（Deep Compression; Weight Sharing (k-means)
& ロスレス圧縮）により,
AlexNetのパラメタ（240 MB）を510倍軽量化（0.47 MB）

F. N. Iandola, "SqueezeNet: AlexNet-level accuracy ---", <https://arxiv.org/abs/1602.07360>
<http://www.slideshare.net/embeddedvision/techniques-for-efficient-implementation-of-deep-neural-networks-a-presentation-from-stanford>

まとめ

1. ロボカップ@ホームロボット

- ✓ ホームロボットの要素技術

2. ホームロボット向け物体認識・把持システムの構築

3. 実践！DCNNによる画像認識器の構築法

- ✓ GoogLeNetと転移学習
- ✓ 2016年9月時点でのお勧めGPU

比較적으로手軽に
試せます！

4. 回路化に関する研究紹介

ホームロボットおよびそこに搭載される知的処理全般は
ひびきのAI社会実装研究会の強力なアプリケーションです！

謝辞

- 九州工業大学 森江 隆 先生（カーロボ連携大学院 事業推進責任者 & Hibikino-Musashi@Home 創設者）
- Hibikino-Musashi@Home の皆さん，OBの皆さん



九州工業大学
森江 隆 先生



Hibikino-Musashi@Home 2016年メンバー 一同

脳型計算機の実現

ハードウェア/ソフトウェア/ ネットワーク複合体

最先端デバイス・ソフトウェア
のデザイン法と利用技術探求



ソフトコンピューティング

脳の情報処理様式を
工学的に具現化



アプリケーション

知的動画処理, 自律型
ロボット 等を通した応用
領域の作りこみ



本研究室の魅力

1. 最先端ハードウェア, ソフトウェア, ネットワークの実践的技術とシステム化に関するノウハウを習得できます！ 企業が欲しがると技術人材を育成します。
2. “脳型計算機”という21世紀最後のフロンティアを工学的立場から開拓します。
3. 魅力あるアプリの研究開発に注力しています。基盤技術(How)を押さえた上で“何を作るか？”を重視した What型・アプリ指向の研究推進を行います。

生命体工学研究科・脳型知能ハードウェアグループ



人間知能システム工学専攻

脳型知能ハードウェア グループ

田中
研究室

森江
研究室

田向
研究室

知能デバイス
ナノ材料

人工知能処理モデル

脳型集積回路

脳型計算機システム
デジタル集積回路

分子アーキテクニクス

知的画像処理・システム

新ナノデバイス開発

ロボット制御

理論, ハードウェアアルゴリズム, デジタル, アナログ, 新奇デバイス, 応用を
横断した脳型ハードウェアの研究が出来る稀少な研究グループです！

基礎と応用をつなぐ人材教育

- 基礎（AI）と応用（ロボット・自動車を含む社会実装）を繋ぐ人材教育
- カーロボ連携大学院 <http://jgs.kyutech.ac.jp/>
 - ✓ 九州工業大学，早稲田大学，北九州市立大学の連携で平成25年開設
 - ✓ **自動車・ロボットの高度化知能化分野**において先端研究開発を主導する高度専門人材（T字型人材）を継続的に育成することを目指す
 - ✓ 実際の自動車やロボットを使った大学院レベルの演習にも注力
- ひびきのAI社会実装研究会を主体にAIコース新設を検討中



ホームロボット演習



自動運転演習

Thank you!

ご清聴ありがとうございました。

<http://www.brain.kyutech.ac.jp/~tamukoh/>

