

農業における深層学習の活用

～Raspberry Piで実装するキュウリ選別システム～

#2018/08/31 SWEST20

ハウス栽培で年間を通してキュウリを栽培・出荷

- ほ場面積0.4ha
- 家族経営





機械化されたと言われる近代農業だが...



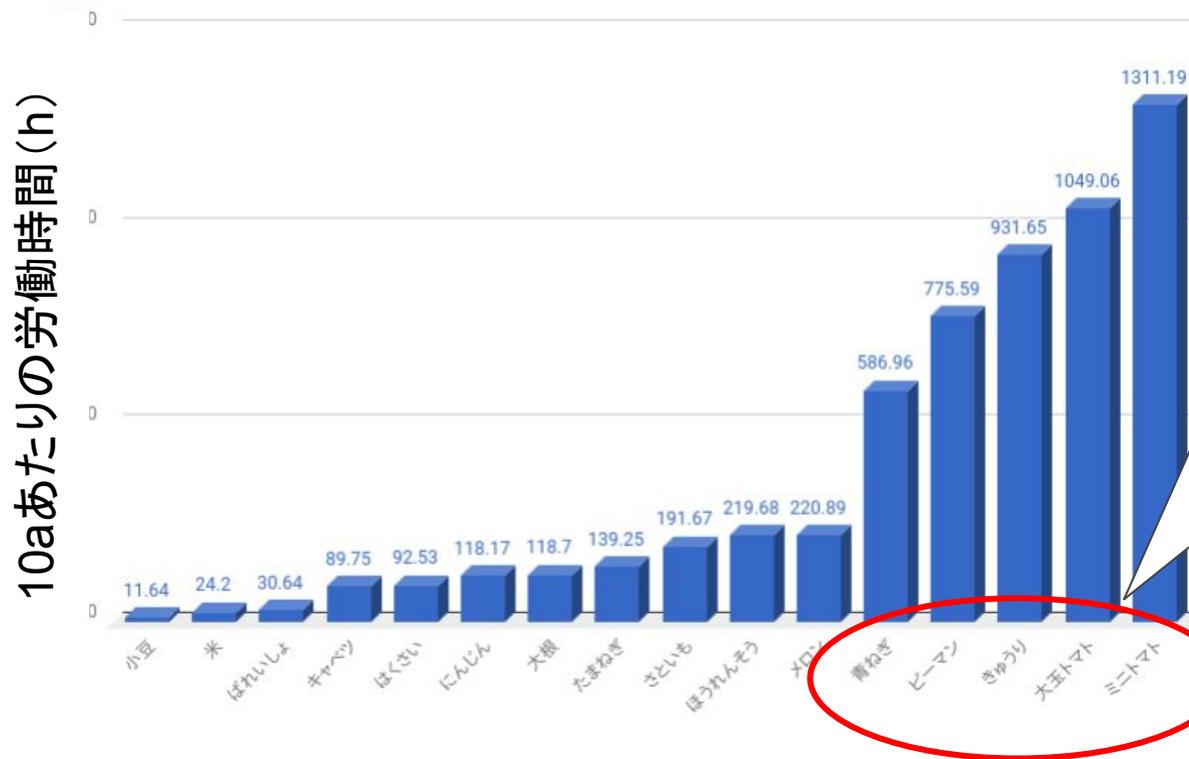
まだまだ手作業が多い！特に果菜類



農業の労働時間

● 品目別10Aあたりの労働時間

農林水産省:品目別経営統計(2007年)より



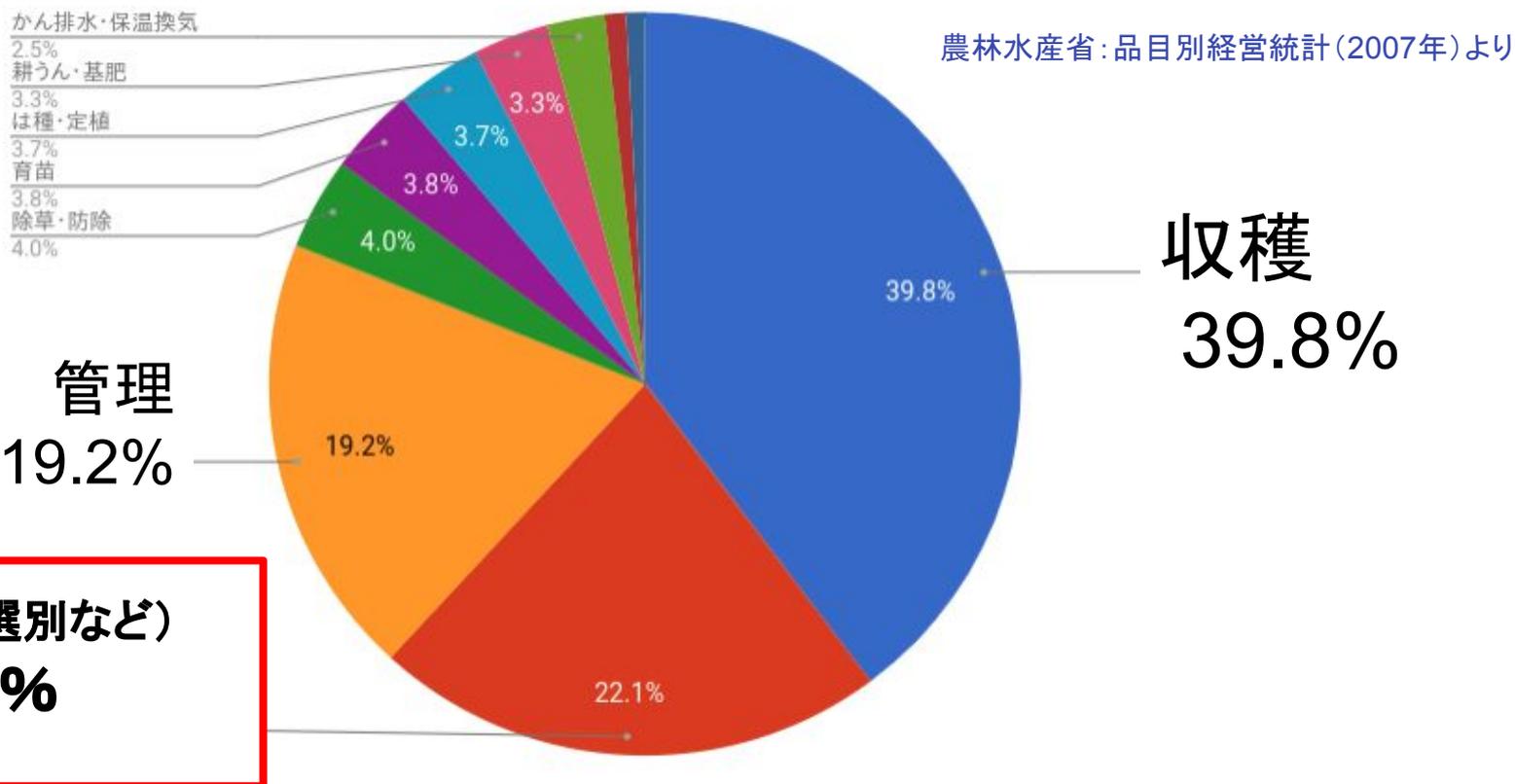
・ピーマン
・きゅうり
・トマト
・ミニトマト
果菜類は手間がかかる！

機械化できていない→大規模化もできない

特に細かい作業が多い果菜類は労働時間が多い傾向がある

きゅうり栽培の労働時間

- きゅうり栽培における作業別の労働時間の割合



きゅうり農家の仕事の約1/5は出荷作業に割かれている
出荷作業は頑張ったところで売上が大きく増えるわけではない

きゅうりの選別作業とは

- 長さ、太さ、曲がり具合、色などにより、等級・階級を分ける作業
- 我が家では9等級に分類

等級	秀	B	量目
階級	2L L M S 2S	大 中 小	C 太 5K

汐見胡瓜出荷組合



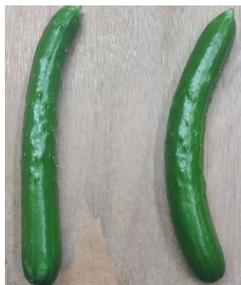
なかなか難しい選別作業

- 対象は自然物。定量的な選別基準があるわけではない
- 長年の経験(主観)、生産者のこだわり
→習得に時間がかかる
- 毎日同じ基準で分けている=信頼
→作業者間での統一が重要

特に品質の判断が難しい



秀品 B品



秀品 B品



秀品 B品



秀品 B品

今まで熟練農家が行っていた判断と同等のことをコンピュータにやらせることはできないか？

注目したのはディープ・ラーニング

- 世界的な画像認識コンペ (ILSVRC2012) で圧勝
 - 中間層が7層のニューラルネットワーク
- 画像認識能力では人間を超えた

	Team name	Filename	Error (5 guesses)	Description
Deep Learning	SuperVision	test-preds-141-146.2009-131-137-145-146.2011-145f.	0.15315	Using extra training data from ImageNet Fall 2011 release
	SuperVision	test-preds-131-137-145-135-145f.txt	0.16422	Using only supplied training data
長年の Feature engineering	ISI	pred_FVs_weighted.txt	0.26602	Weighted sum of scores from classifiers using each FV.
	ISI	pred_FVs_summed.txt	0.26646	Naive sum of scores from classifiers using each FV.
	ISI	pred_FVs_wLACs_summed.txt	0.26952	Naive sum of scores from each classifier with SIFT+FV, LBP+FV, GIST+FV, and CSIFT+FV, respectively.
	OXFORD_VGG	test_adhocmix_classification.txt	0.26979	Mixed selection from High-Level SVM scores and Baseline Scores, decision is performed by

「ケタ」が違う

人工知能の未来 (http://www.soumu.go.jp/main_content/000400435.pdf)より

人間の目の代わりとなる技術として注目されている

Convolutional Neural Network による画像認識

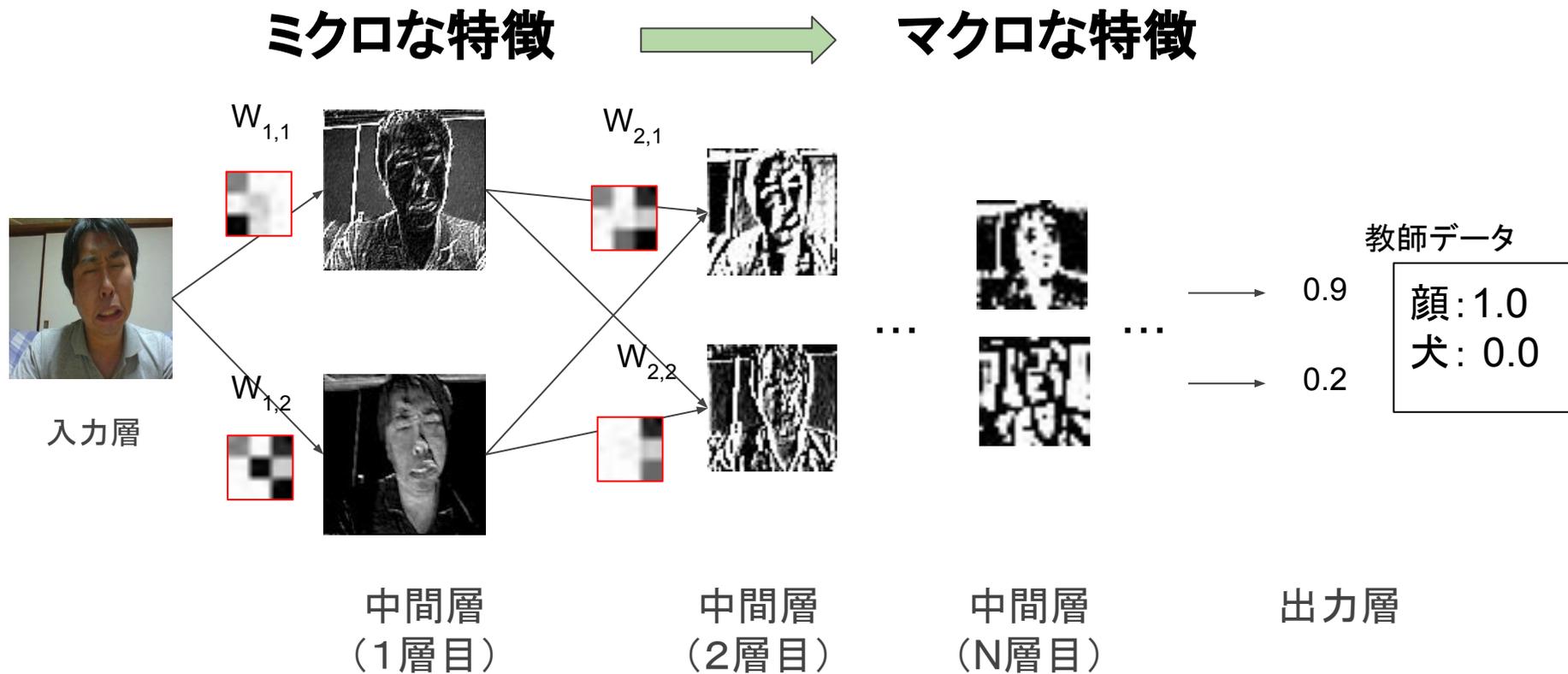
- 画像認識分野においてはConvolutional Neural Network (CNN)を用いたアルゴリズムが成果を出している
 - VGGNet, ResNet, DenseNet,...
- CNNを使って”きゅうり画像”から”等級”を予測する



CNNを使って「きゅうり画像」から等級を予測する

Convolutional Neural Networkとは

- 顔画像を入力した時の例 (VGG16の中間層出力)



試作1号機

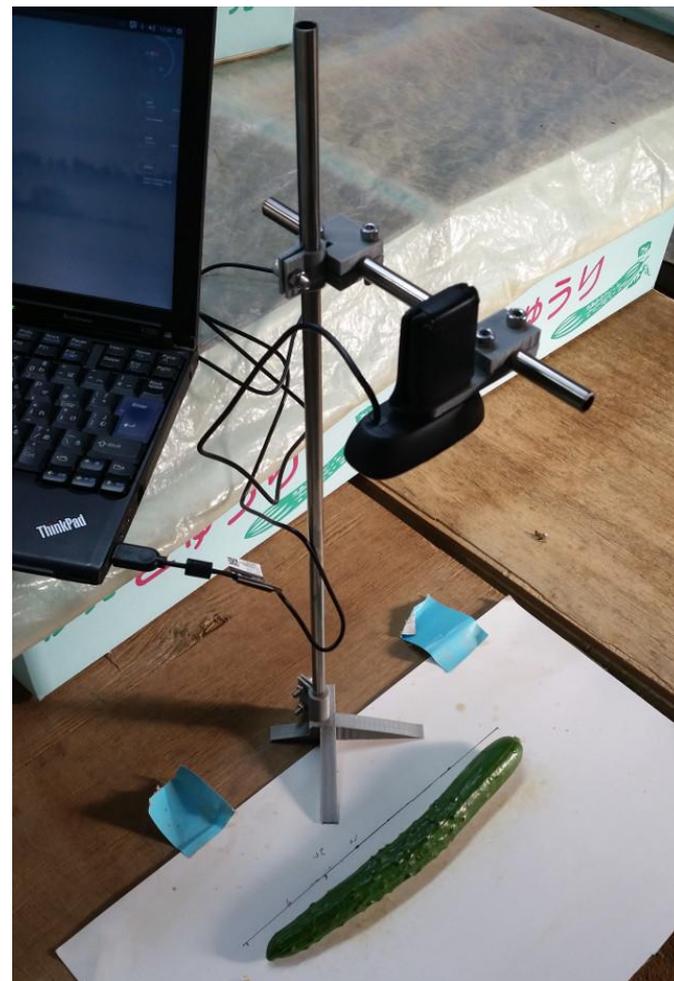
- とにかくキュウリ画像で試してみよう
- 試作1号機誕生
 - Webカメラをスタンドに固定
 - 上からキュウリの画像をとる

【ハード】

- ・Webカメラ Logicool C270(1,500円)
- ・アルミのパイプ(数百円)
- ・結合パーツ(3Dプリンターで印刷。数十円)
- ・固定用ボルト(数百円)

【ソフト】

- ・カメラ制御: OpenCV(オープンソース)
- ・機械学習: TensorFlow(オープンソース)



制作期間: 1週間 制作費: 約3000円

深層学習はデータが重要！！・・・とはいえ、

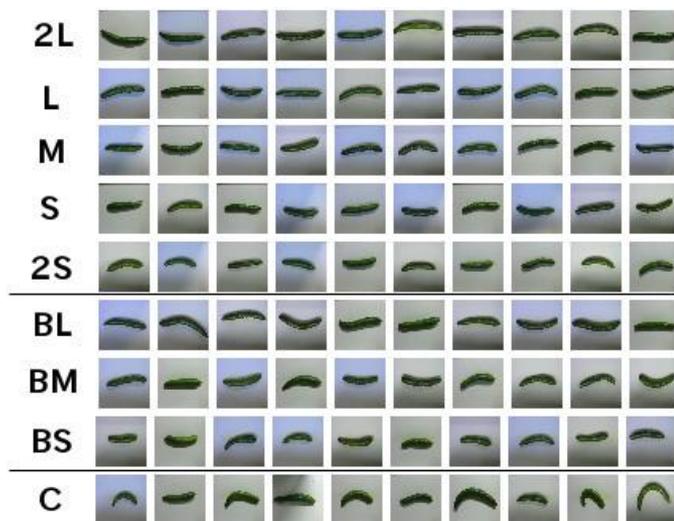
- 大量の良質な教師データが必要
 - 学習データが少ないと過学習により、実際に使用する環境では精度がでないという問題がある

とはいえ、初めからコストをかけても・・・

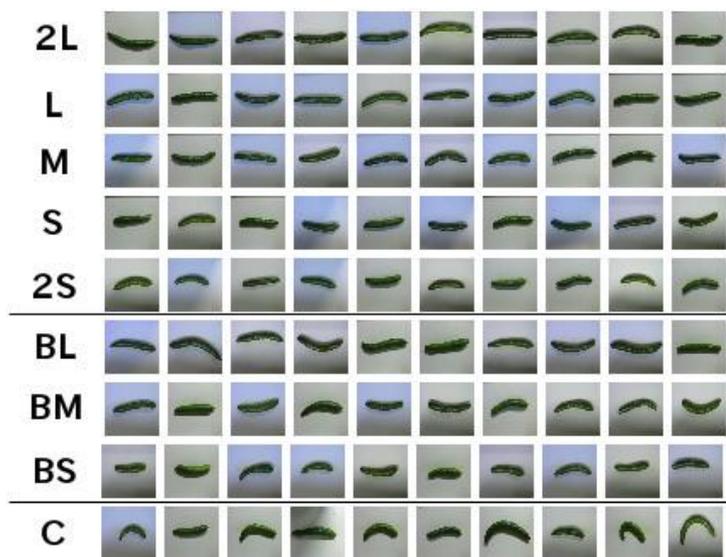


試作1号機では、

- 2475枚(1クラスにつき275枚)
- 背景に白い厚紙
- なるべく位置を揃える



試してみた結果



教師画像 : 2475枚

80%の正答率

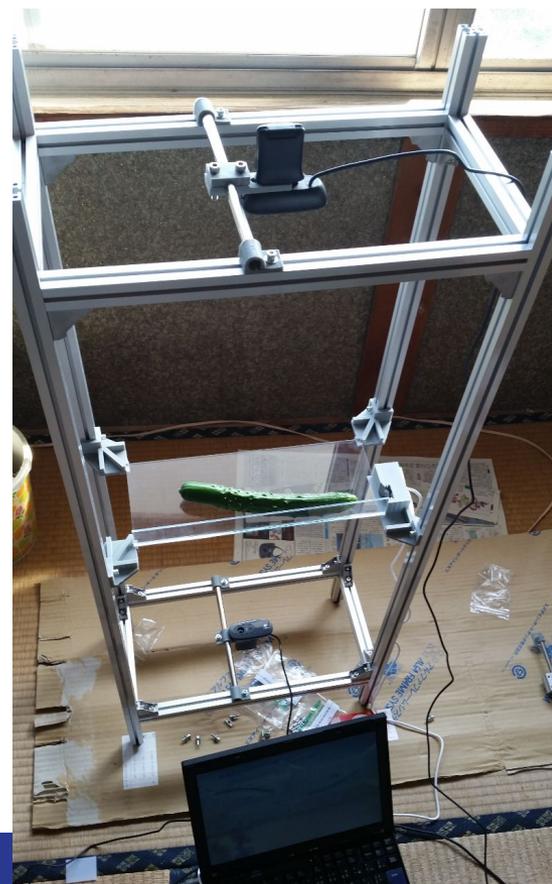
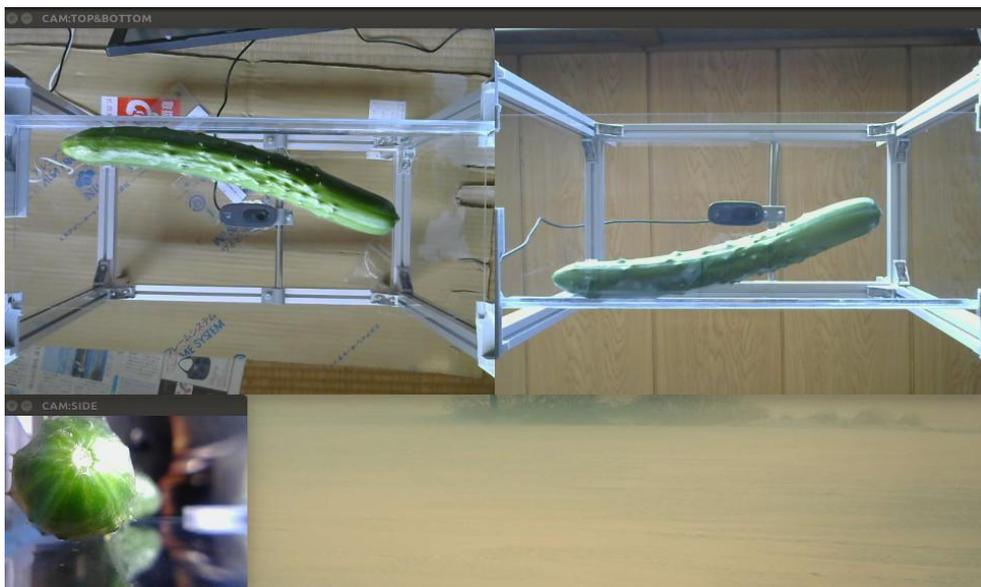
- 思ったよりいい結果だった
- おそらく…
 - 短期間データでの評価であったため
 - 置く位置を揃えたため

可能性が見えた！試作2号機の開発開始

- 試作1号機の結果からディープラーニングで選別ができそうな気がしてきた
- もっと人間の仕分けに近づけたい
 - カメラの台数を増やしたら精度が上がるかも

撮影台の作成

- キュウリ画像を取得するための台
- カメラを上、下、横に設置し、キュウリの全面を捉える
- 照明を付けて明るさを一定にする

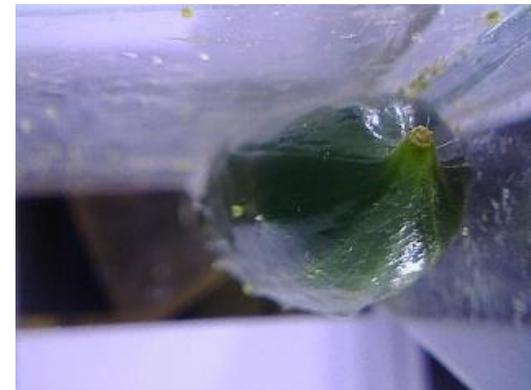


教師データ集め

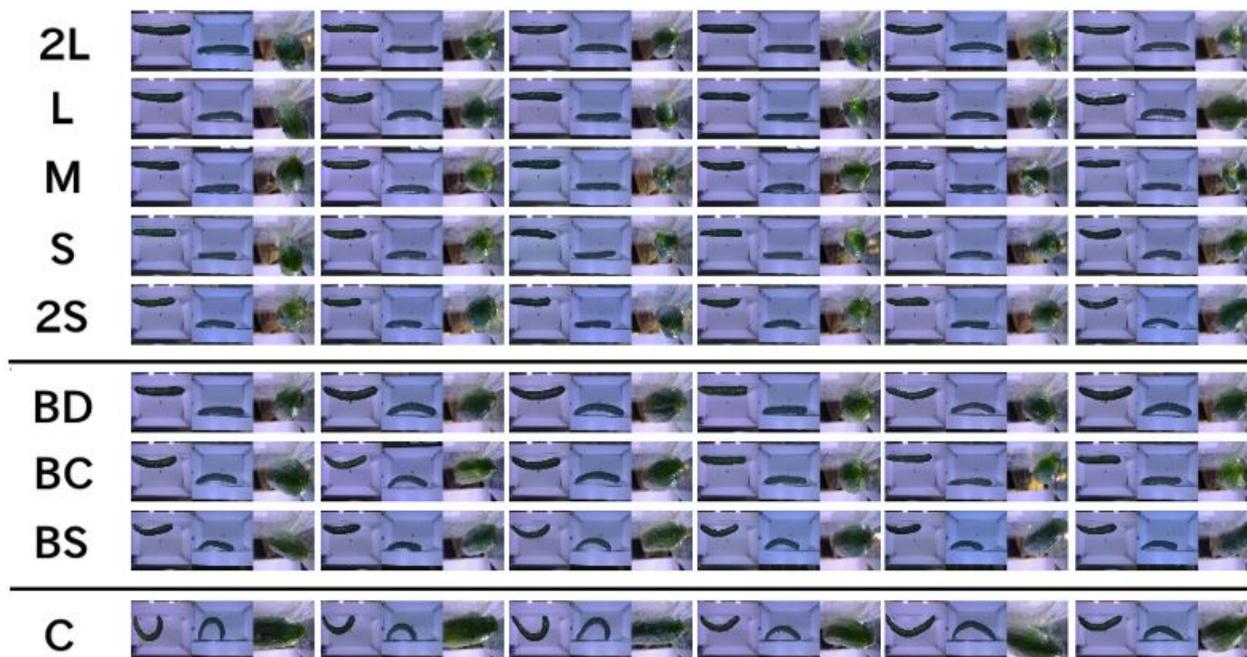
- 作ったキュウリ台で教師データとなる画像を撮影&ラベル付け



キュウリ1本につき上、下、横の3枚の画像を取得



教師データ集め



集めた画像 : 8500組

解像度 : 80x80x3

ラベル : 10種類

教師画像 : 7000組

テスト用画像 : 1500組

2ヶ月間ほどかかった



学習結果



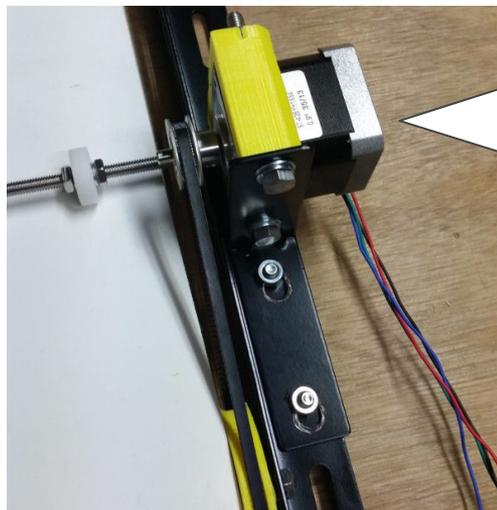
教師画像：8500枚

91.6%の正答率

- データを増やした効果
- カメラを増やした効果
- 照明を追加した効果

ベルトコンベア作り

- 判断したキュウリを指定の箱まで運ぶベルトコンベアがほしい



部品は3Dプリンター

ArduinoMicro
を使って制御

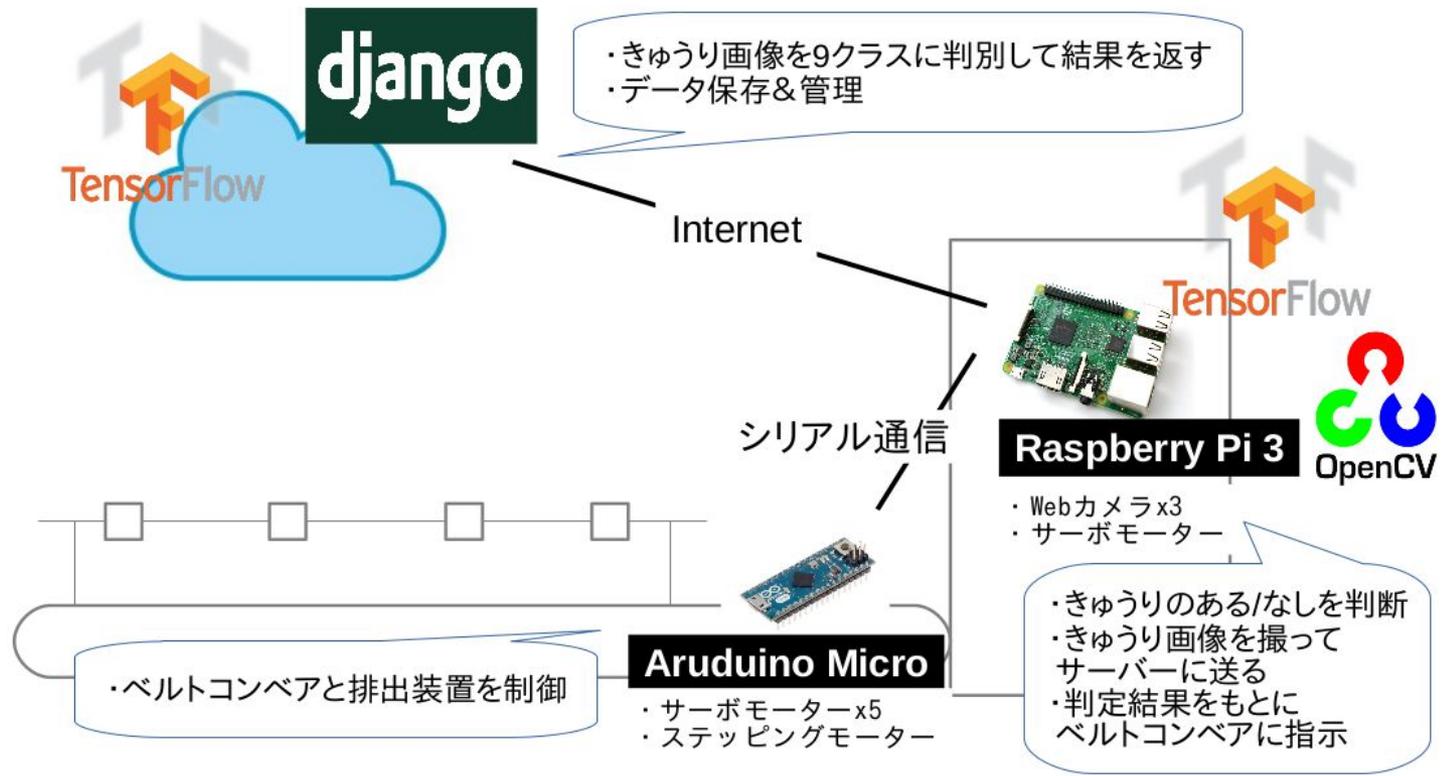


完成



RaspberryPiを使って実装

- 学習は上手く行った→次は実際に現場に持ち込む
- 農作業現場は屋外→壊れることを考えてラズパイ使おう
- ラズパイ遅い→クラウド使おう(IoT万歳...という時期もありました)



試作2号機



MakerFaireTokyo2016で展示

- ベルトコンベアをコンパクトにしてMakerFaireで展示



コレがきっかけでInterface誌の記事執筆のお仕事を頂けるようになった

試作2号機完成



お蔵入り

遅いし、キュウリに傷が
付くから使えんな！



認識部分はいい感触
しかし、まだまだ実用には程遠い！

現場で動かしてみても分かったこと

- 作業場での使用を考えると、毎回画像をアップロードするのは厳しい
 - 気軽にインターネットに繋げない環境
 - レイテンシの問題 → 人間並みの判断速度(500ms程)を目指そうとするとネックになる
 - 通信費が…
- 設置環境による外乱が厳しい
 - 午前中に仕分けをすると認識精度が落ちる 約90% → 約70%ほど
 - 西日が差し込むと使えなくなる
 - 汚れにより精度が落ちる
- 成長時期や季節ごとに作物の出来が異なる
 - 出来始めため、後半細め
 - 季節によるばらつき

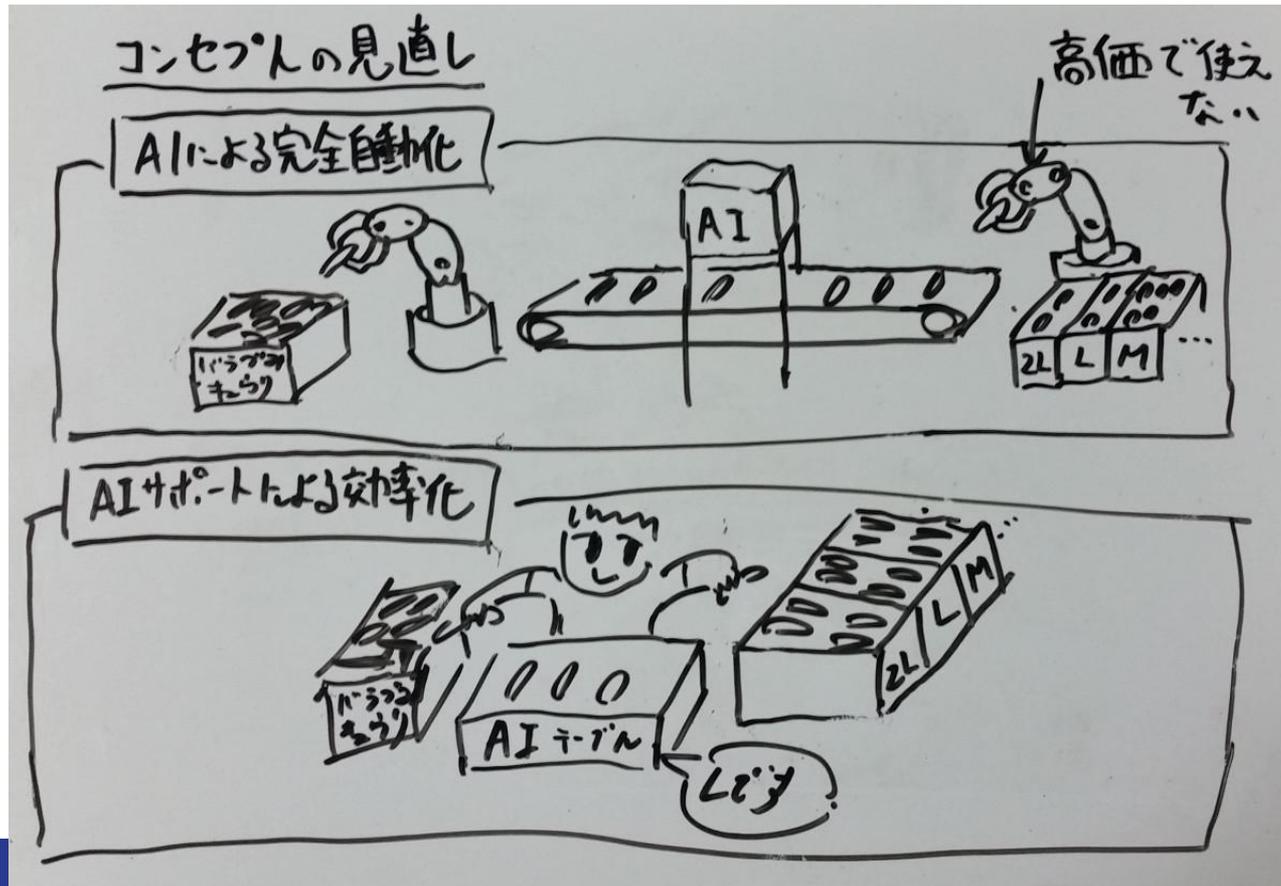


農業現場で使うとなると、

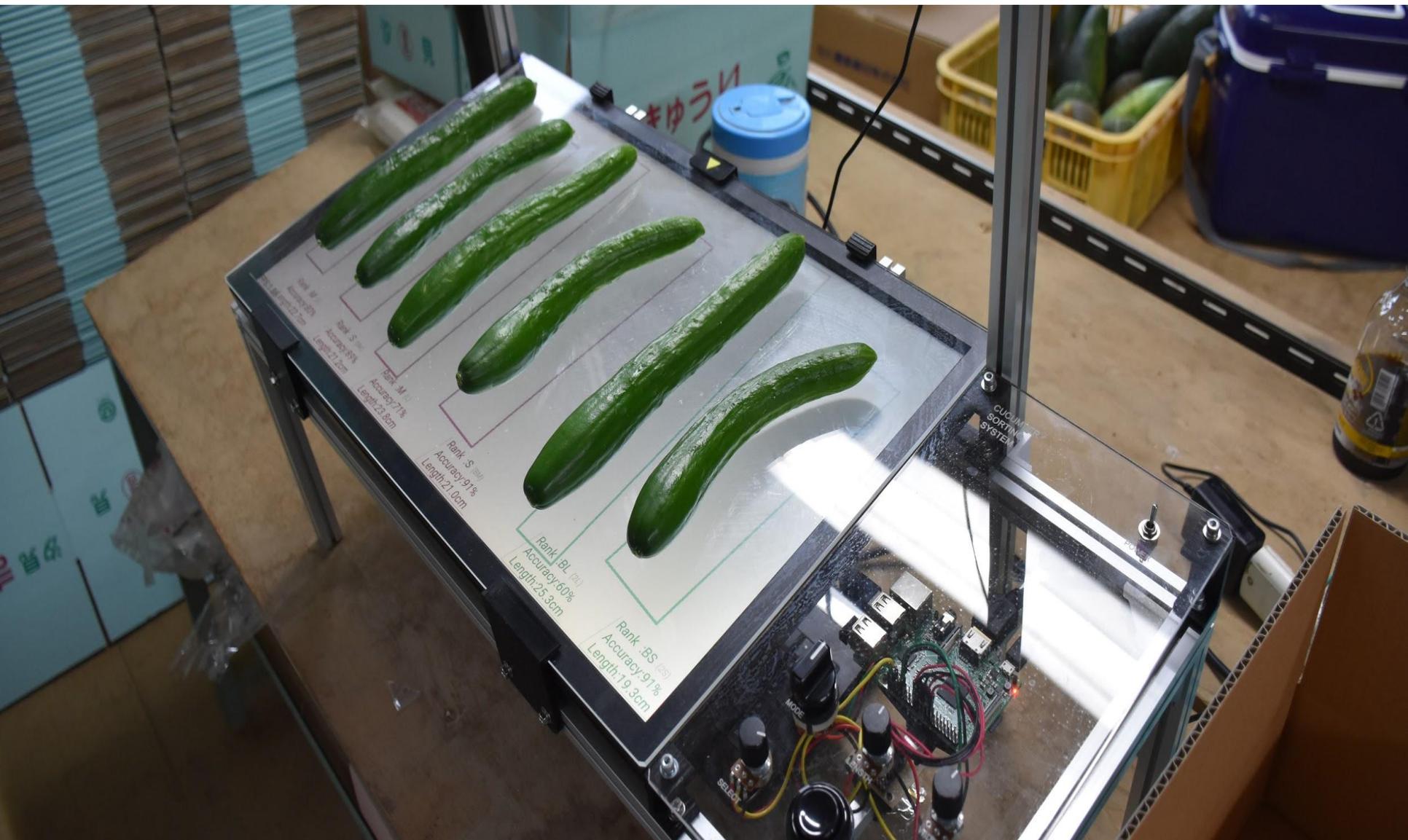
- 画像認識系はエッジ側で動かしたい
 - 成長時期や季節による果実形状の偏りに対応が必要
- ということが分かった

3号機の開発着手

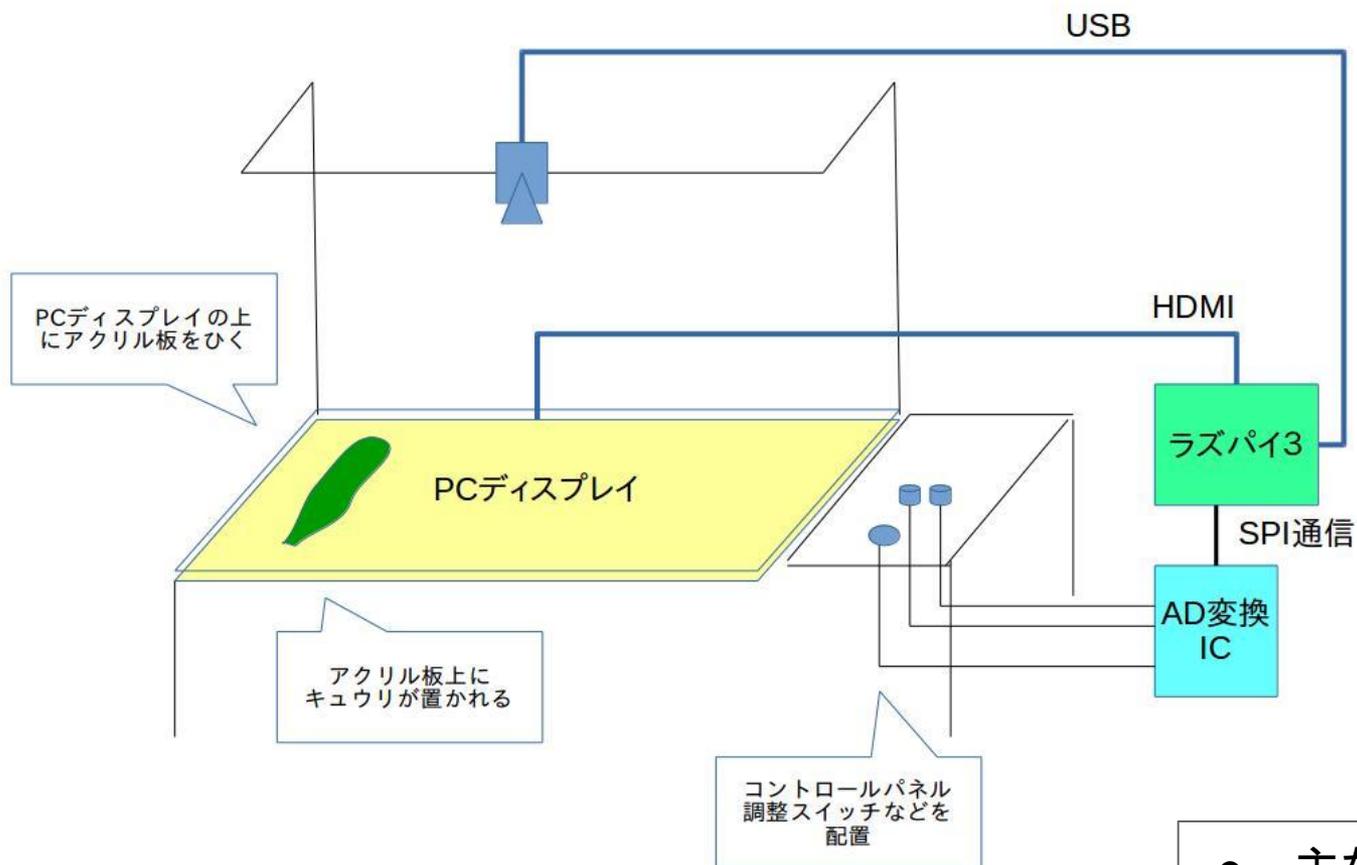
- 開発コンセプトの変更
 - 『AIによる自動化』→『AIのサポートによる効率化』



テーブル型キュウリ選別システム



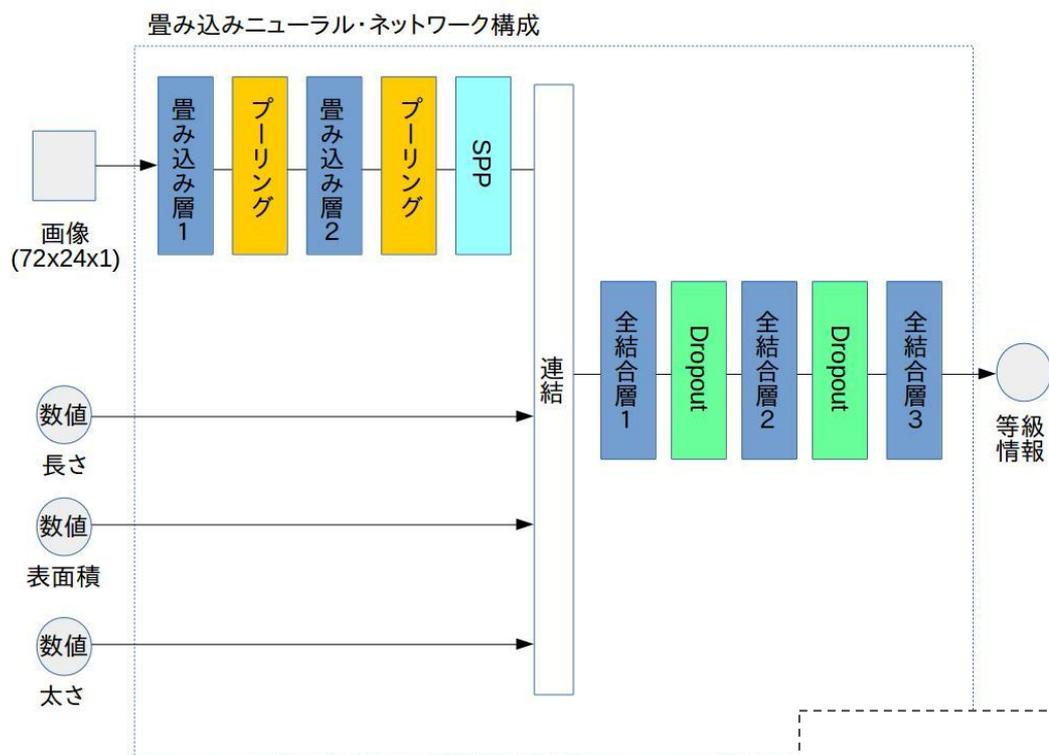
システム構成



- 主な使用ソフトウェア
 - TensorFlow v1.1.0
 - OpenCV 2.4
 - Kivy 1.8.0

ニューラルネットワーク構成

- 5層の畳み込みニューラルネットワーク



【前回から変えたところ】

画像に加え、キュウリの長さ、表面積、太さの数値を入力

→等級判定のキャリブレーションのため

SPP[1]

→認識精度向上 & 画像変形に対するロバスト性向上

活性化関数にかける前にBatch Normalization[2]で正規化

活性化関数: Relu
最適化関数: Adam

[1] Kaiming He, et al. "Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition"

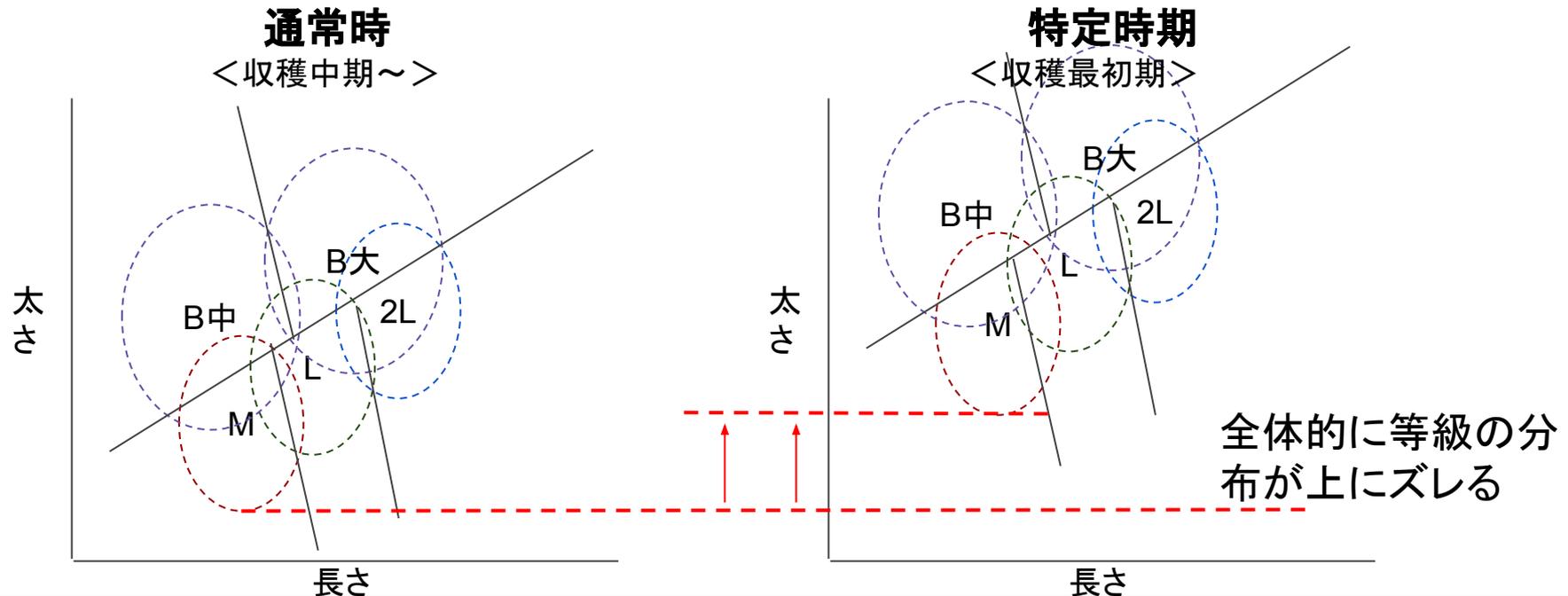
[2] Sergey Ioffe, et al. "Batch Normalization: Accelerating Deep Network Training by Reducing internal Covariate Shift"

前回開発を通してわかった課題

- 季節や成長段階によってキュウリ果実の形に偏りがある
 - 取れ始めのころは全体的に太め、徐々に細長くなる
 - 熟練者作業者は、全体の傾向を見極め等級判断基準を微調整している

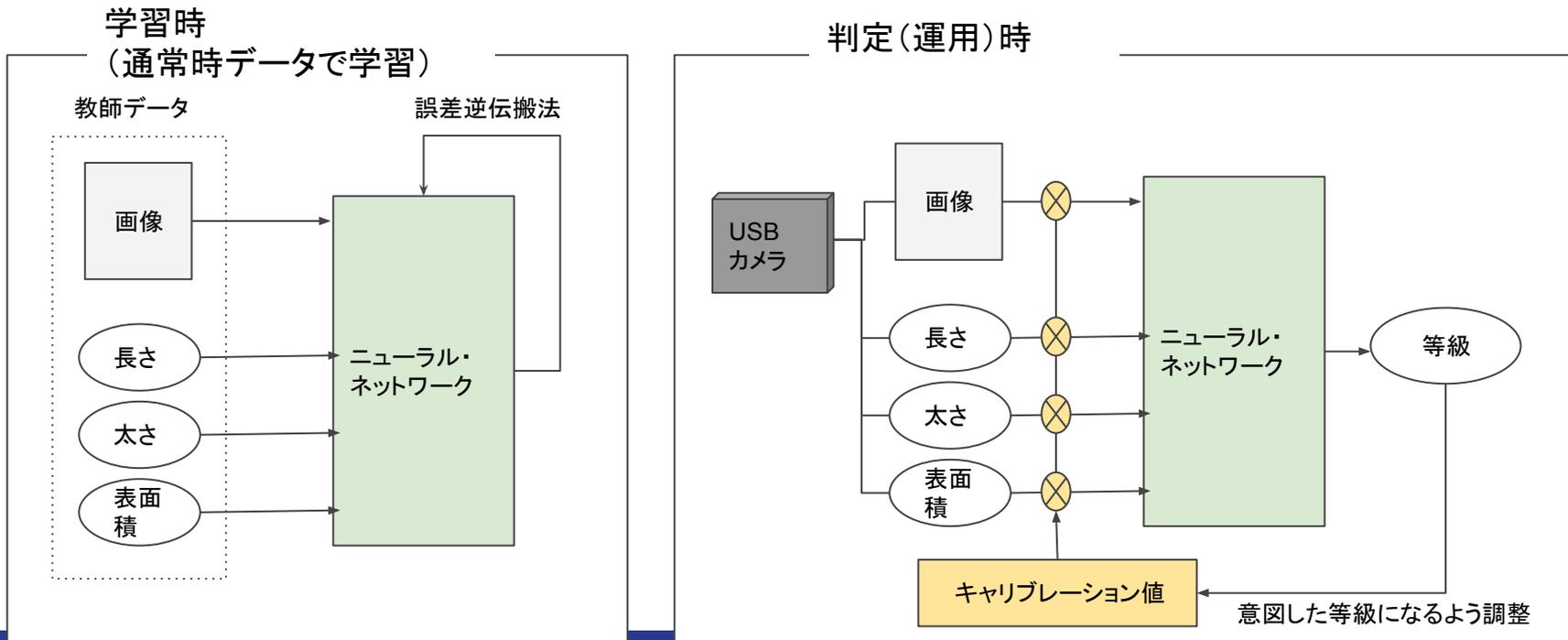
【長さ, 太さによるキュウリ等級の分布傾向】

※長さ, 太さによる等級の分布は, 経験から線形に変化することがわかっている

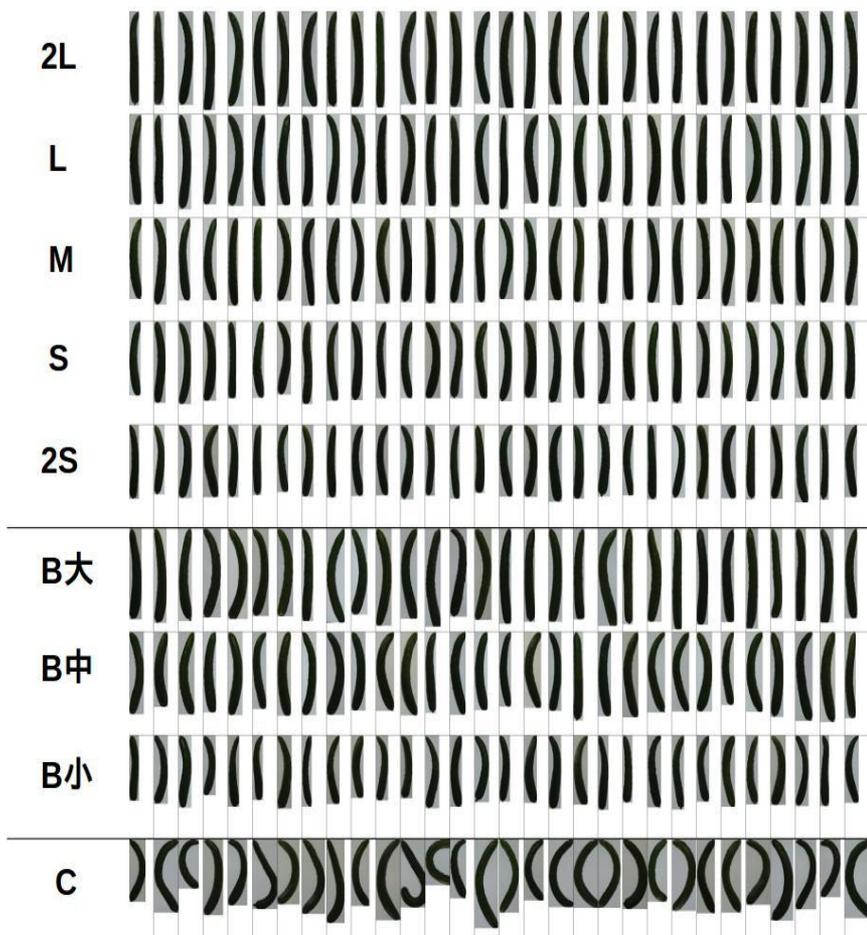


判定のキャリブレーションの仕組み

- 収穫時期に合わせニューラルネットワークへの入力を調整する
 - 学習時は通常時のデータで行う(標準基準を学習)
 - 運用時は目的の等級になるよう入力画像&数値を調整する
 - 例) 収穫初期で通常時なら”B品”だが”M”と判断してほしい→太さを細くする



教師データ集め



集めた画像 : 36,000枚

解像度 : 72x24x3

ラベル : 9種類

教師画像 : 28,000組

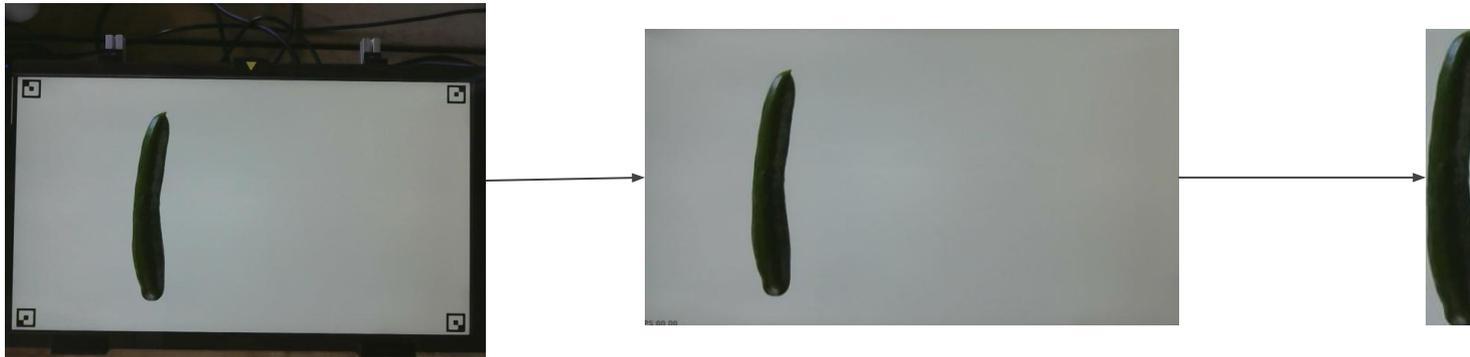
テスト用画像 : 8,000枚

1ヶ月間ほどかかった

機械学習を使った手法は、データの前
提条件(カメラの向きなど)が変わ
ると取り直しになる・・・コスト高すぎ
ね

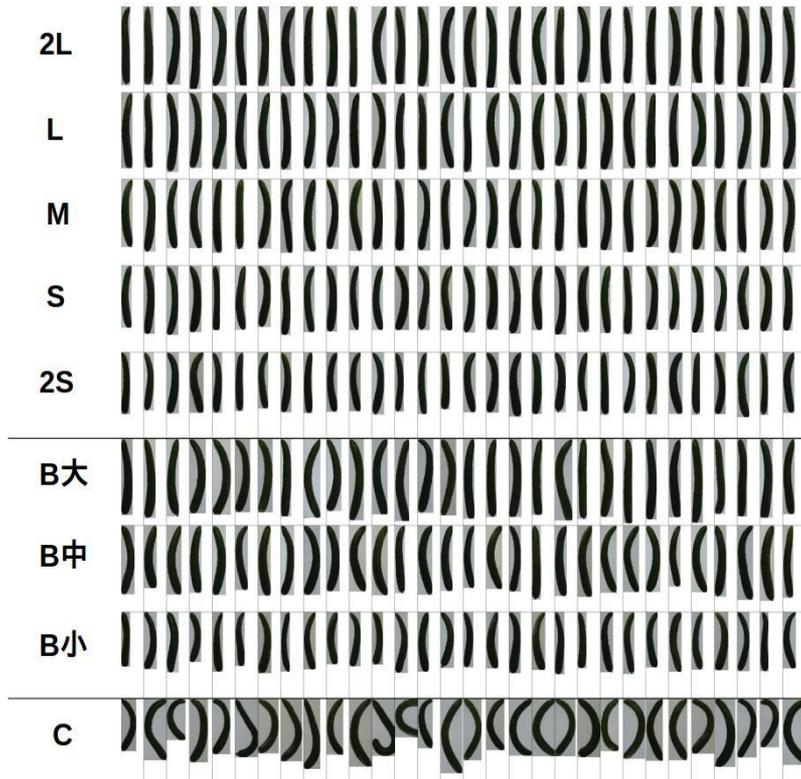
画像処理でテーブル上のキュウリを検出

- マーカー座標検出→射影変換→輪郭抽出→キュウリ画像切り出し



- 背景などキュウリ以外はなるべく減らす
- 背景を白く(バックライト)して形がくっきり見えるようにする

学習結果



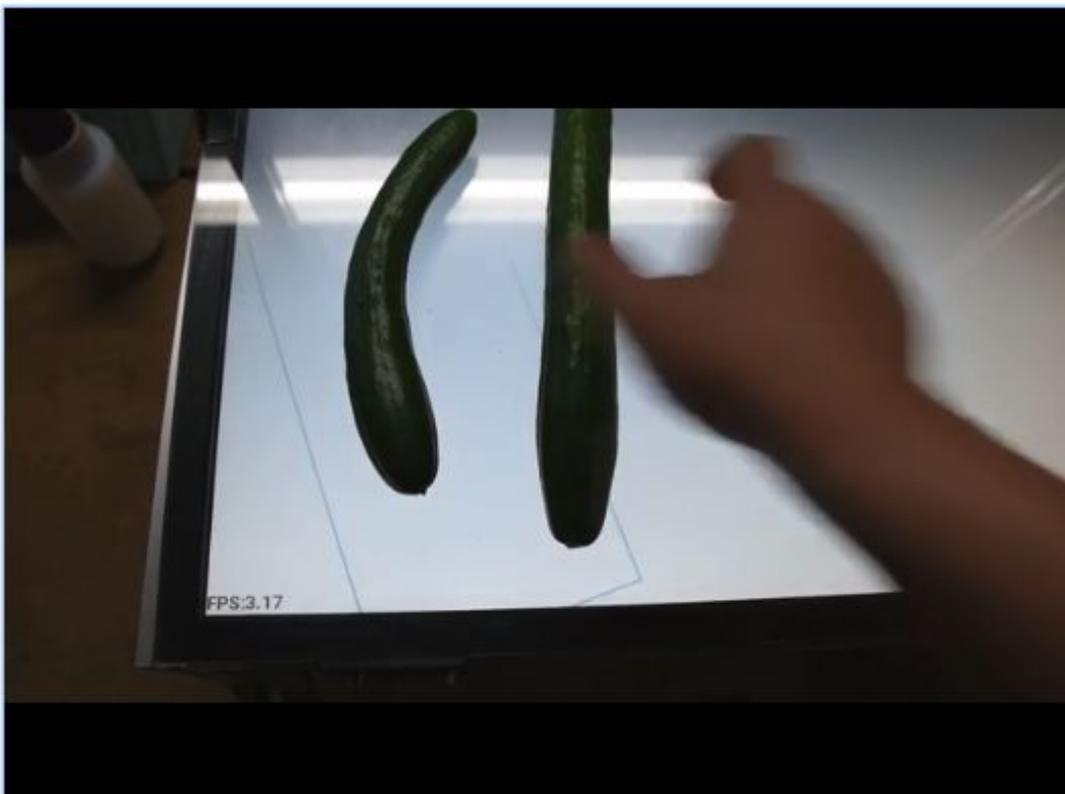
79.4%の認識率

おそらく、カメラの数を減らした影響か

91.6%→79.4%に下がった

ラズパイでの処理速度を考慮して、
画像解像度を落とした
(高解像度であれば約90%を確認)

実際の動作



実環境での評価

- 収穫初期の一番形状が安定しない時期の結果ですが・・・
 - M,S,2S,B中,B小,Cのキュウリ851本の判定結果

		熟練者の判断						適合率
		M	S	2S	B中	B小	C	
システムの判断	M	128	2	0	2	0	0	97.0%
	S	27	137	7	12	1	0	74.5%
	2S	0	33	120	6	32	0	62.8%
	B大	3	0	0	3	0	0	—
	B中	41	13	3	93	1	0	61.6%
	B小	0	4	32	5	71	0	63.4%
	C	0	0	0	0	0	75	100%
再現率		64.3%	72.5%	74.1%	76.9%	67.6%	100%	

**正答率：
73.3%**

設置環境による外乱の影響で正答率は7割ほどに低下

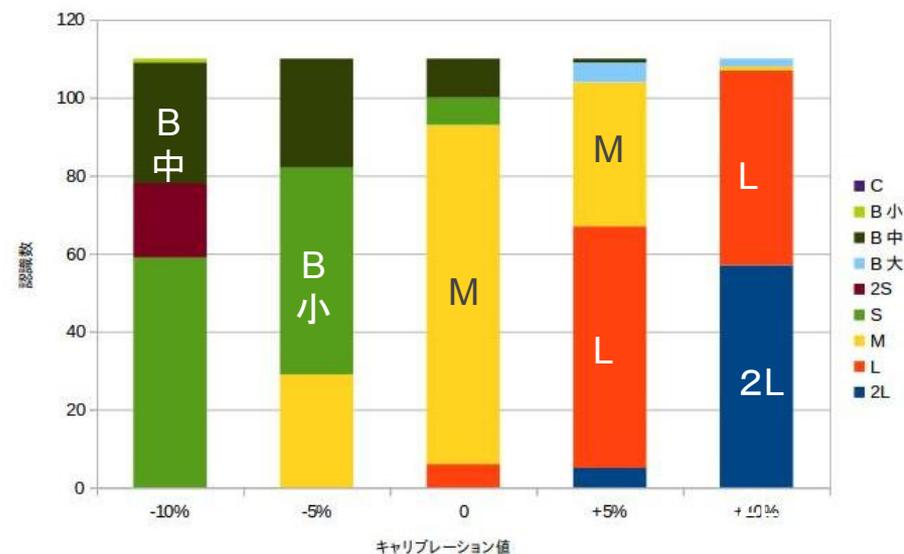
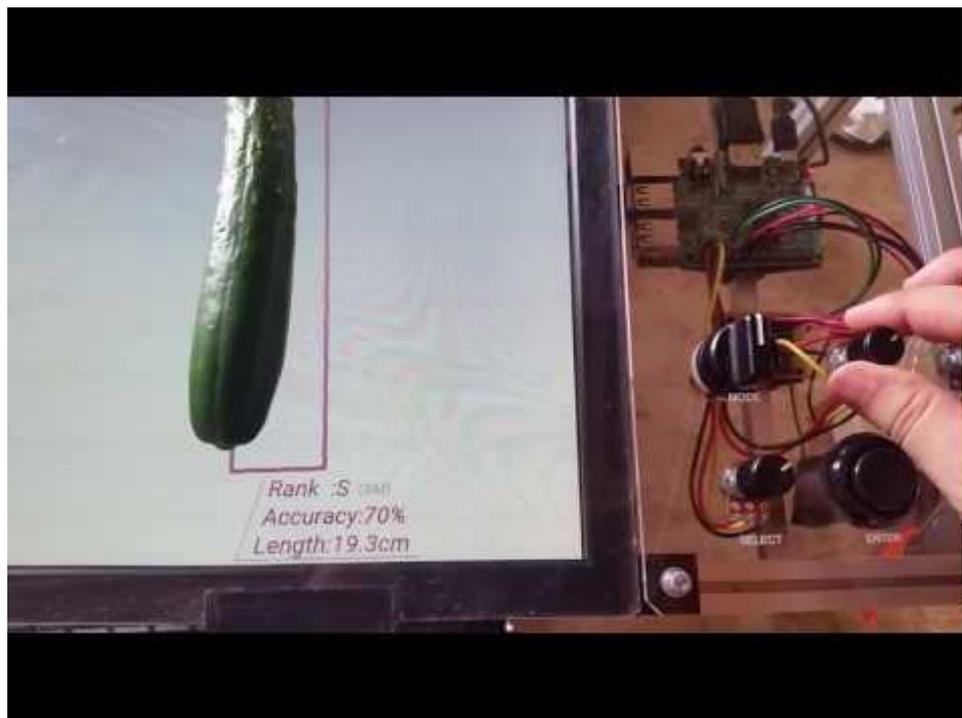
判定のキャリブレーション

- 選別作業を開始するまえに、コントロールパネル上のつまみで調整する
 - 長さ: $-20\% \sim +20\%$
 - 太さ: $-20\% \sim +20\%$
- つまみは直感的でわかりやすい



キャリブレーションの様子

テスト画像を使って、M等級110本に対し、長さのキャリブレーション-10~+10%を行った際のNN判定等級数の変化



期待通りの動きになっていることが確認できた

実務で使うまでになった



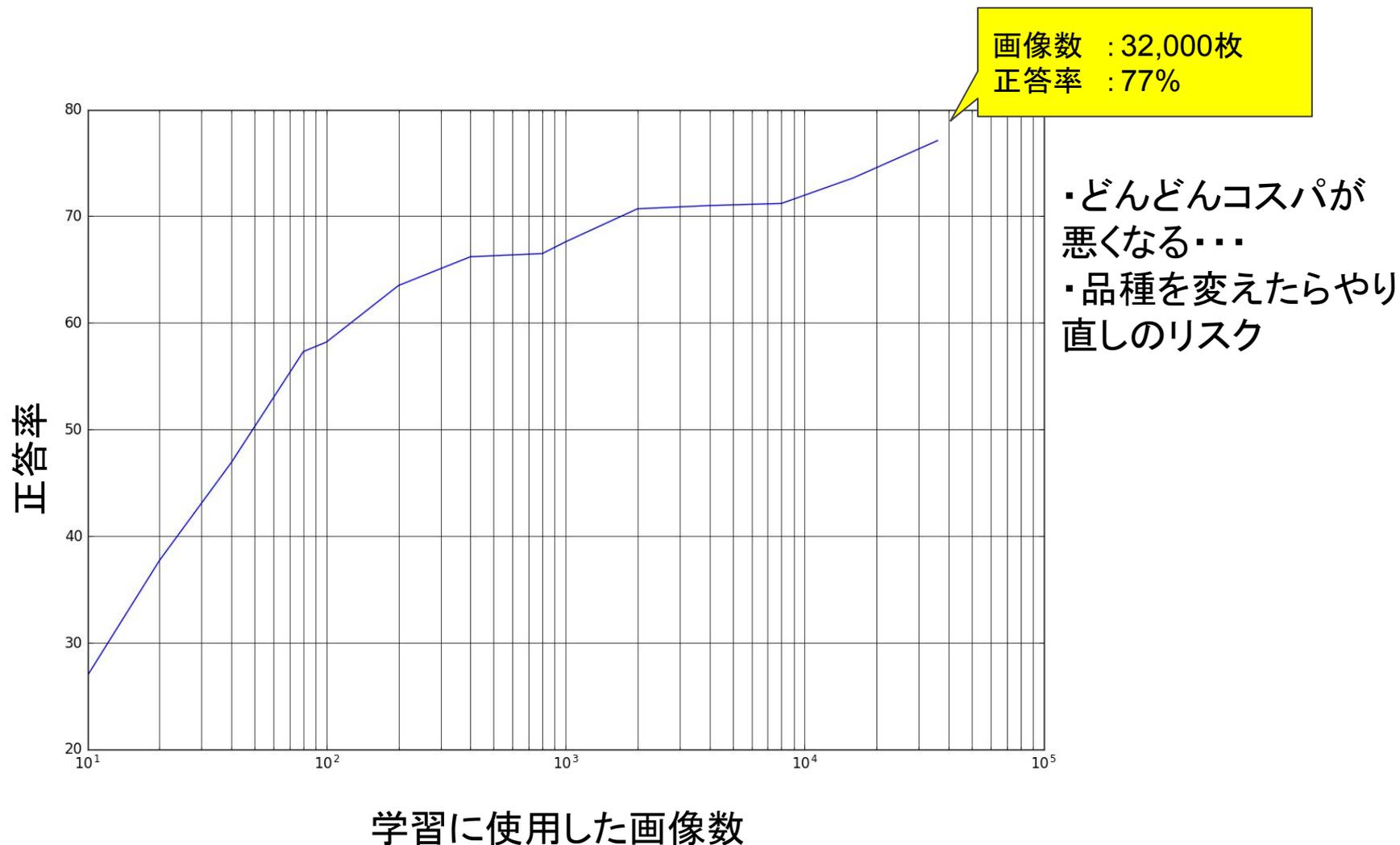
- ラズパイだけで、4本を約1秒ほどで判定
- 仕分けスピードが1.4倍になった
- 熟練者のスピードにはまだ敵わない

まあ、これなら使えるか？！



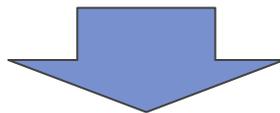
- 一応、ボスのOKをもらい実業務で使ってる
- しばらく使ってみるとまた色々問題が・・・絶賛改良中

データ数と正答率



役に立つ場面は多いと思う

- 完璧(正答率100%)な認識精度を目指すのはそもそも無理
 - 品質保証できないし・・・
 - 認識間違いがあることを前提に考える
- 画像処理のプロフェッショナルじゃなくても、ある程度の精度を出せる可能性大
 - 非専門家こそ活用すべき技術
 - ただし、範囲を絞って小さく初めることが重要(本来は高コストになりがち)



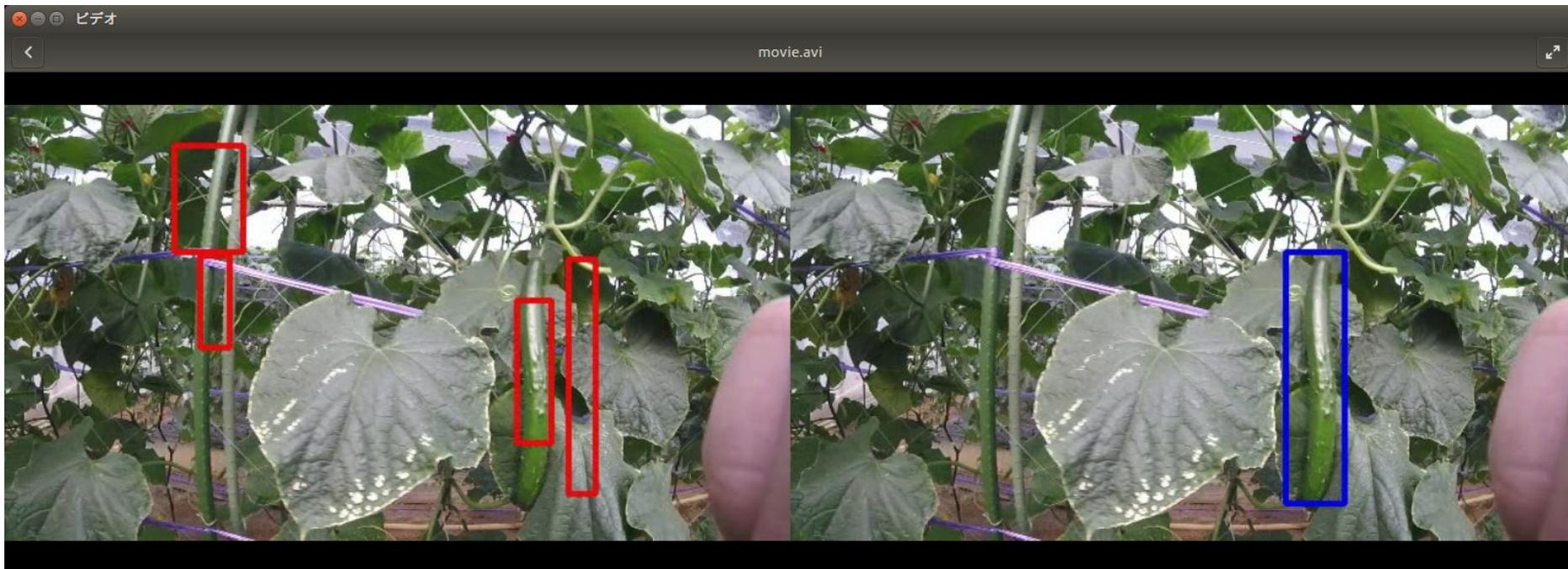
現状は、多少認識間違いがあったとしても、あったほうが助かるという場所を見つけると、
ディープ・ラーニングをスムーズに導入できそう

開発を通して分かったこと

- 認識精度を上げる & 精度を保証することが難しい
 - 運用しながら改善していくしかない
 - 大量のテストデータが必要
 - 機械学習+従来手法, 機械学習+外部システムなど
- 終わりのなきイテレーション？
 - 使えば使うほど賢くなる・・・こともない？
 - いつになったら開発が終わるのか？
- 深層学習は強力なのでどんどん活用したい
 - 人間の認知能力をかなり再現できる
 - ソフトに任せる部分を増やした設計

農業の活用も期待される

- 物体検知アルゴリズムYOLOを基に下記変更を加えたもので実験中
 - YOLOアルゴリズムからクラス認識部分を削除(不要なため)
 - YOLOアルゴリズムにおいて1セルで検出できる物体数を1に固定(まとめて検出しても問題ないため)



予測結果

教師データ

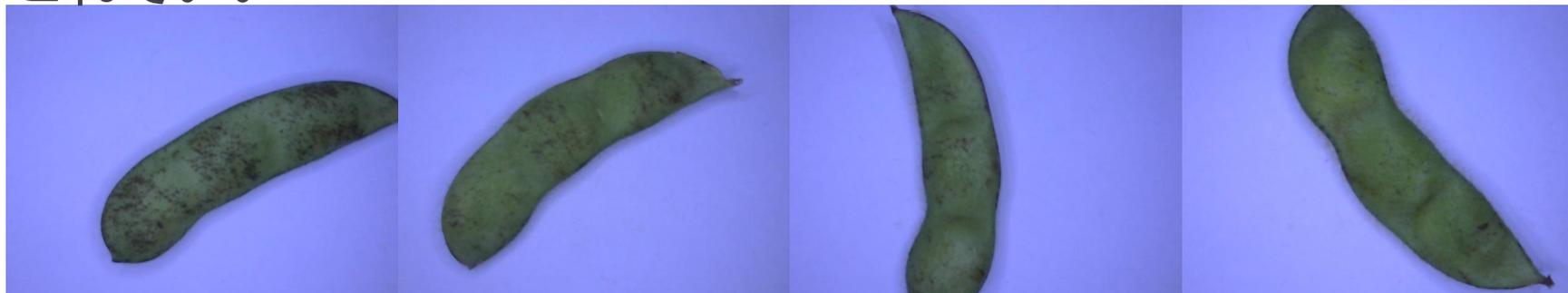
機械学習を前提とした設計

こうしがち



考え方を考える

これでいい

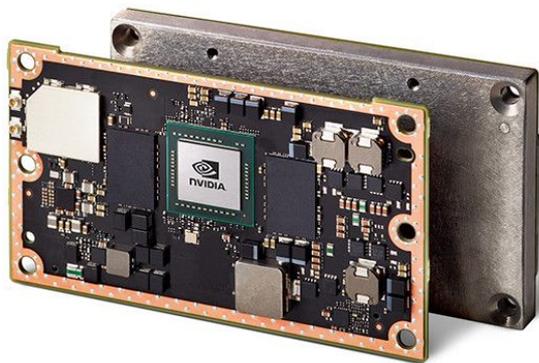


“ソフトウェアなんてハードを動かすだけのもの”ではない
積極的にソフトウェアに任せて素早く安く作る

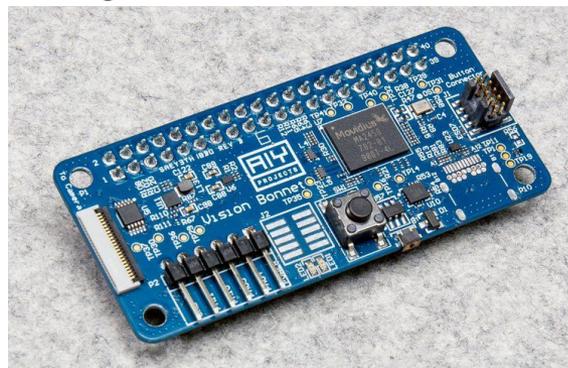
推論処理速度の向上

- 個人でも買える推論を高速に行うためのデバイスが増えてきた

NVIDIA Jetsonモジュール



Google AIY Vision Kit (Vision bonnet)



Intel Movidius NCS



Google Edge TPU



まとめ

- 仕分けノウハウの永続化・再利用
 - 人間の判断をそのままコンピュータに置き換えることができる
 - コンピュータを介してノウハウを保存・伝えることができる
 - 今までの伝承「人→人」、これから「人→AI, AI→人」
- 画像処理の専門知識がなくてもできる
 - データを集めるだけで、問題を解決できるAIを作ることができるかもしれない
 - プログラム不要で使えるツールが増えた
- エッジ推論デバイスを使えば速度も実用レベルに
 - 学習はクラウド, 実行はエッジというパターンが増えそう
 - 再学習など運用をどうするかが課題



キュウリの品質UP！収量UP！



技術の蓄積