

獣害対策のための アライグマ自動認識の試行

立命館大学 理工学部 電子情報工学科
大野真史 初田慎弥 ○泉知論 孟林

アライグマとその被害

過去、ペットとして人気で北米から輸入

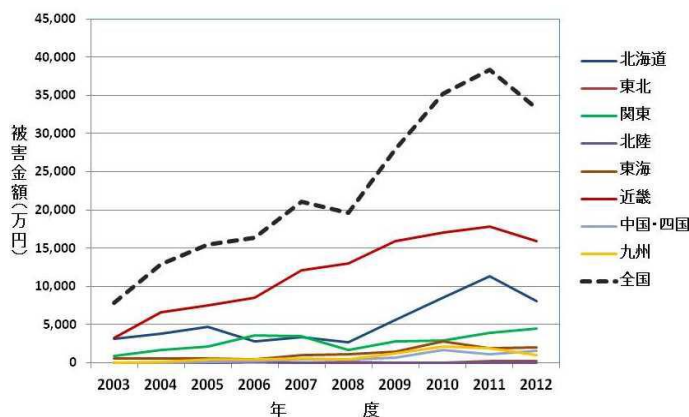
現在、野生化し特定外来生物に指定

様々な問題を引き起こす

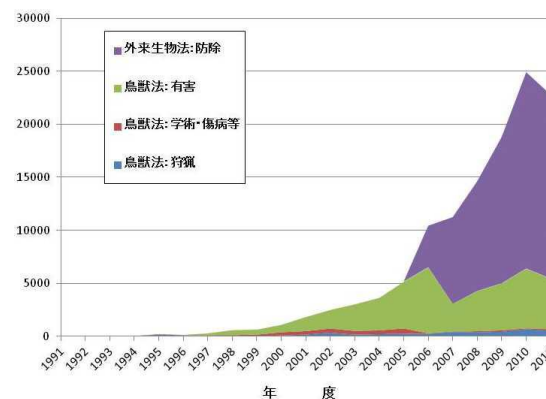
- 2012年度の農業被害額は約3億3千万円
- 建築物の汚損・破損など



「アライグマ防除の手引き（計画的な防除の進め方）」より



アライグマによる
地方別農作物被害金額の推移



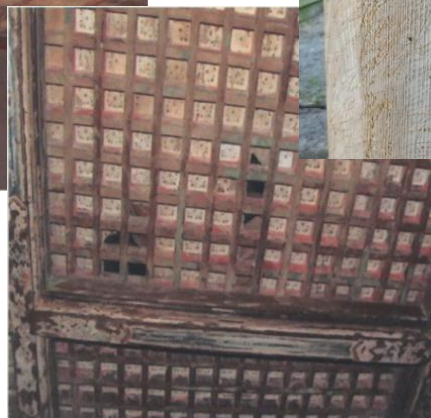
アライグマ捕獲数の推移

歴史的建造物の被害

歴史的建造物（文化財）など寺社に被害
棲処として主に屋根裏に侵入、繁殖
足跡、爪痕、屋根板等を食い破る、糞尿
数か月棲みついては移動



協力：大石神社（京都市山科区）



神社の柱の爪痕



撮影 2017.5.27
大石神社（京都市山科区）

アライグマの被害対策

鳥獣保護法：狩猟獣
外来生物法：特別外来生物

これらの法律に基づき捕獲・駆除

2010年には約2万5千頭を捕獲

早期に発見し，確実に生息していることを確認する作業が重要

現在の確認方法

- ・ 餌トラップ
- ・ 足跡トラップ
- ・ カメラトラップ
- ・ 目撃・痕跡調査



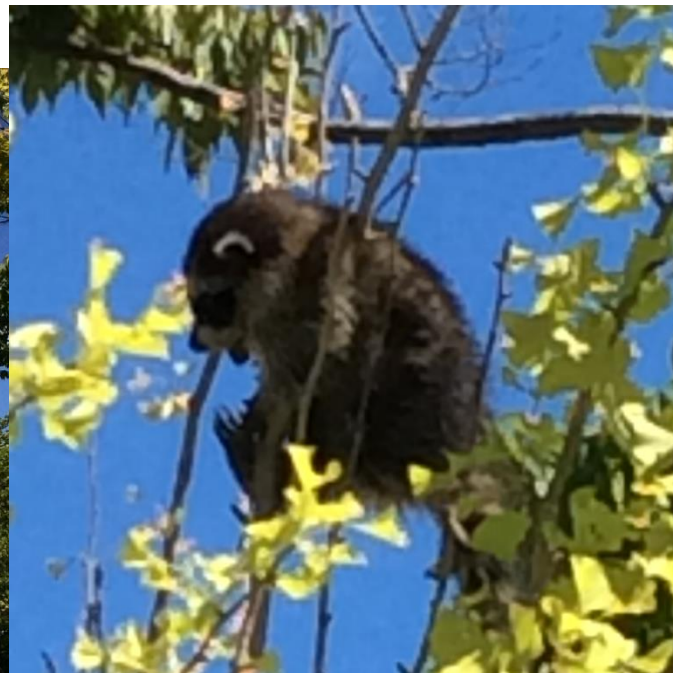
人手による確認

「アライグマ防除の手引き（計画的な防除の進め方）」より引用

広範囲・長期間・継続的に生息や侵入を監視するには時間と費用がかかる

B K Cのアライグマ

撮影 2015.9.11 立命館大学 びわこくさつキャンパス



フィールドカメラ@B K C緑地



動物を自動認識するカメラシステム

- フィールドカメラの高度化
 - 生息、侵入、移動の把握・分析
- 監視カメラの高度化
 - 害獣の確認、対策
 - 不審人物検出の高精度化（動物の除外）



- 動物（アライグマ）の認識
 - 画像処理
 - 機械学習
 - データベース作成
 - 組み込みシステム

アライグマ画像データベースの構築

アライグマ画像データベースの構築

- 機械学習には大量の学習用データが必要
- 特に寺社に侵入する様子
登る、ぶら下がる、もぐりこむ
- 自ら撮影しデータベースを構築する

動画撮影



画像化
ラベル付け



画像切り出し



対象動物の撮影

撮影場所：弥生いこいの広場

協力 ... 弥生いこいの広場、弘前市、関西野生生物研究所

日時：2016年11月14日

撮影対象：アライグマ、タヌキ、
アナグマ、ハクビシン、キツネ



ペット用玩具等を用いて柱を登る様子や穴に体の一部を入れている様子などを撮影した

ラベル付けと切り出し



- 既存の画像形式変換ツールを駆使
- ラベル付けを支援するGUIアノテーションツールを作成 (Visual Studio C#)
- 手作業で連番画像へラベル付け
- 画像切り出しプログラムを作成 (Python)

学習データセットの作成

「アライグマ」「タヌキ」「背景」についてラベル付け

ラベル付けの合計は36578枚

ラベル付けの内訳

	アライグマ	タヌキ	背景	合計
枚数(枚)	11747	2776	22055	36578

それぞれ左右反転を行いデータ拡張

最終的な学習データセットの枚数は合計73156枚

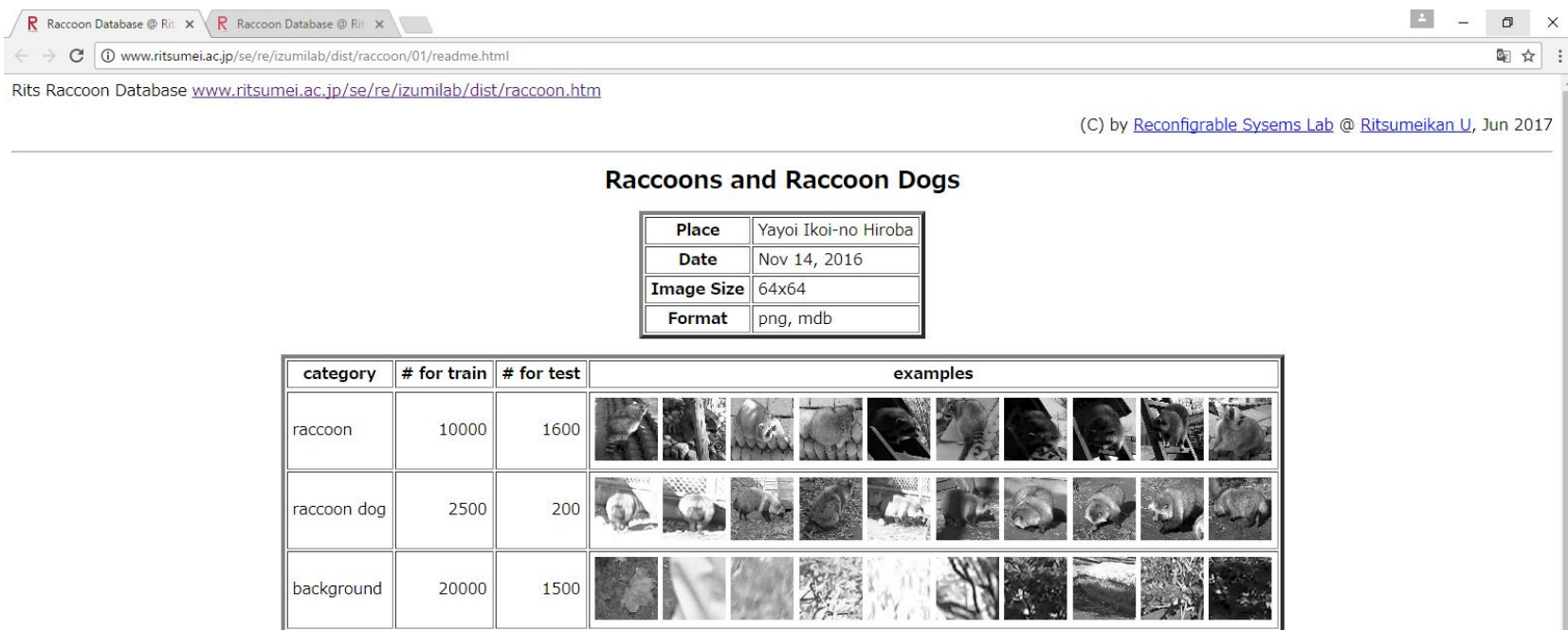


データセット画像の例

データベースの公開

- 以下に公開しています

<http://www.ritsumei.ac.jp/se/re/izumilab/dist/raccoon.htm>



The screenshot shows a web browser window displaying the Raccoon Database page. The page title is "Raccoons and Raccoon Dogs". It includes a table with metadata for the dataset and a table showing the distribution of images across categories.

Place	Yayoi Ikoi-no Hiroba
Date	Nov 14, 2016
Image Size	64x64
Format	png, mdb

category	# for train	# for test	examples
raccoon	10000	1600	
raccoon dog	2500	200	
background	20000	1500	

- Download from [here](#).
- Caffe LMDB data is generated by convert_imageset.bin without resize and grayscale options.
- The images are taken by Reconfigurable Systems Lab, Ritsumeikan University supported by
 - Kansai Wildlife Research Association (www.kansaiwildlife.com),
 - Hirosaki City (www.city.hirosaki.aomori.jp), and
 - Yayoi Ikoi-no Hiroba (www.hirosakipark.or.jp/yayoi).

深層学習の試行

研究背景

本研究では監視カメラの映像を元に画像認識でアライグマを識別するための分類器を構築した画像認識には以下のものがある

ハフ変換

パターンマッチ

- 文字や標識など幾何学図形に適している

畳み込みニューラルネットワーク (CNN)

- 一般物体認識に適している

本研究ではアライグマを認識するので畳み込みニューラルネットワークを使用する

本研究の流れ

畳み込みニューラルネットワークを分類器として使用するには大量の訓練データが必要



対象動物を撮影し、訓練用データベースを構築



その他獣の画像をCIFAR-10 datasetより追加



ディープラーニングのフレームワークであるCaffeを用いて構築したCNNに学習させる



学習してできた分類器を評価

アライグマ画像データベースより

データベースから 32×32 、グレースケールに正規化する



構築したデータベースの枚数は以下の通り
今回は十分に撮影できたアライグマとタヌキを使用する

	アライグマ	タヌキ	合計
学習用	10000枚	2500枚	12500枚
テスト用	1600枚	200枚	1800枚

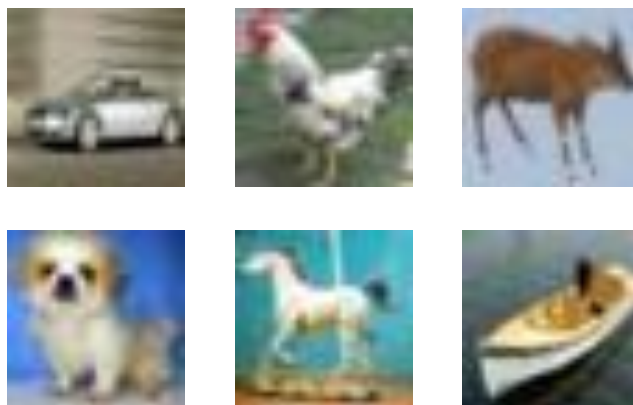
CIFAR-10 dataset

ラベル付けされた画像のデータセット

32×32のカラー画像が10クラス、それぞれ6,000枚

飛行機、自動車、鳥、ネコ、シカ、イヌ、カエル、ウマ、船、トラックの10種類

作者はAlex Krizhevsky, Vinod Nair, Geoffrey Hinton



データセットの作成

アライグマとタヌキの他に動物の種類を増加させるためにCIFAR-10 datasetからネコ、シカ、イヌ、ウマの画像を追加した

追加した後のデータセットの画像の枚数を以下に示す

	アライグマ	タヌキ	ネコ	シカ	イヌ	ウマ	合計
学習用	10,000枚	2,500枚	5,000枚	5,000枚	5,000枚	5,000枚	32,500枚
テスト用	1,600枚	200枚	1,000枚	1,000枚	1,000枚	1,000枚	5,800枚

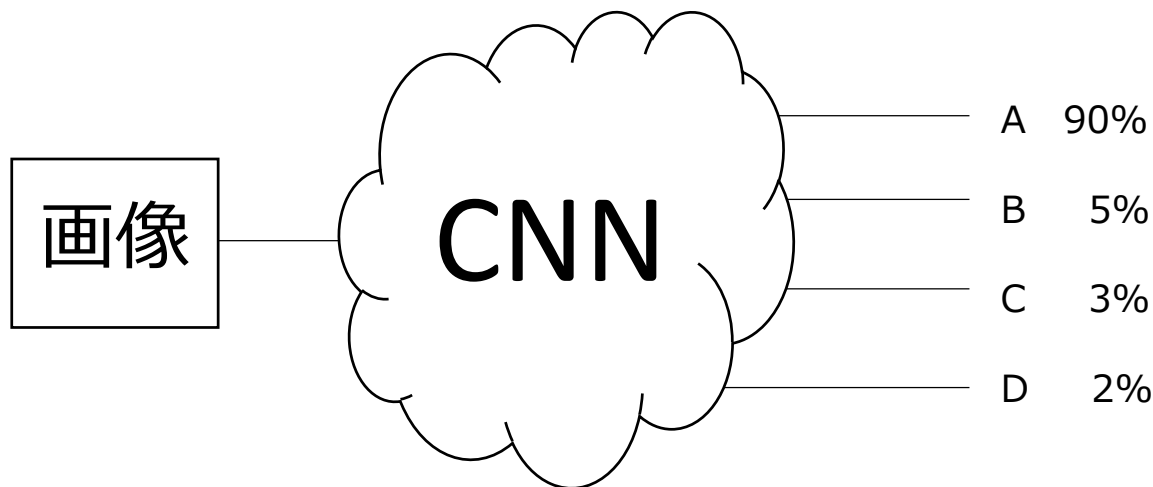
畳み込みニューラルネットワーク

機械学習とは

データからその規則性などをコンピュータやプログラムが自動で発見し、内部状態を更新すること

畳み込みニューラルネットワーク（CNN）とは

画像向けに最適化されたニューラルネットワークの一種
分類器としてのCNNは入力画像が用意したカテゴリに
属する確率を出力する

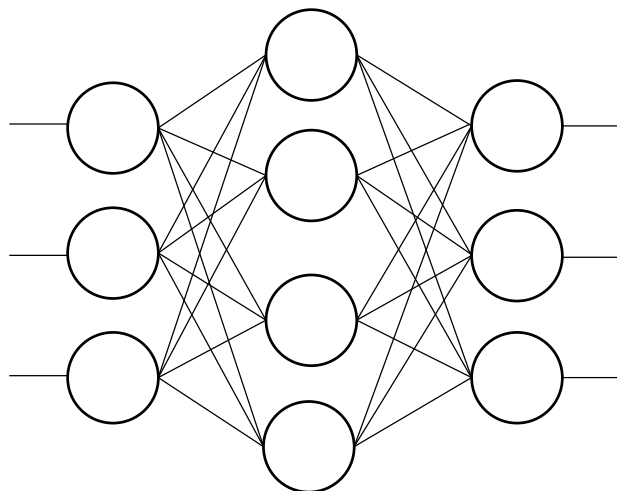


ニューラルネットワークと深層学習

ニューラルネットワーク

脳の神経回路網を模したモデル

複数のユニットによって構成されたネットワーク

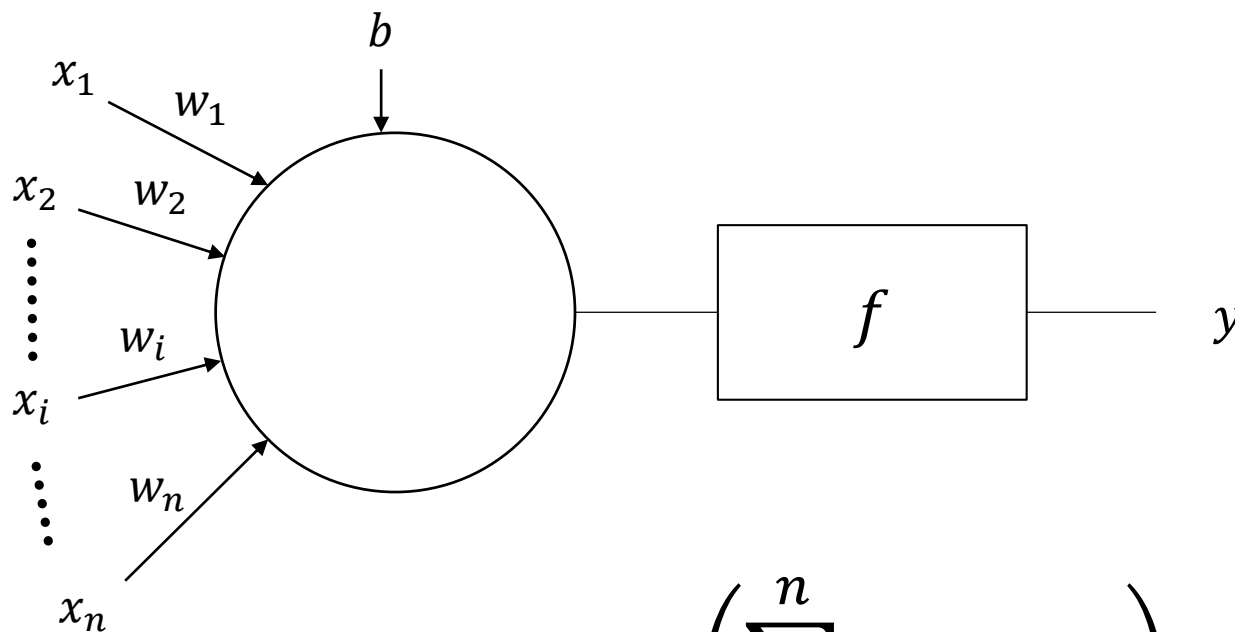


深層学習

階層型ニューラルネットワークにおいて多層構造であることを活用した機械学習

ニューラルネットワークのユニット

ニューラルネットワークを構成する最小単位
ニューロンをモデル化したもの

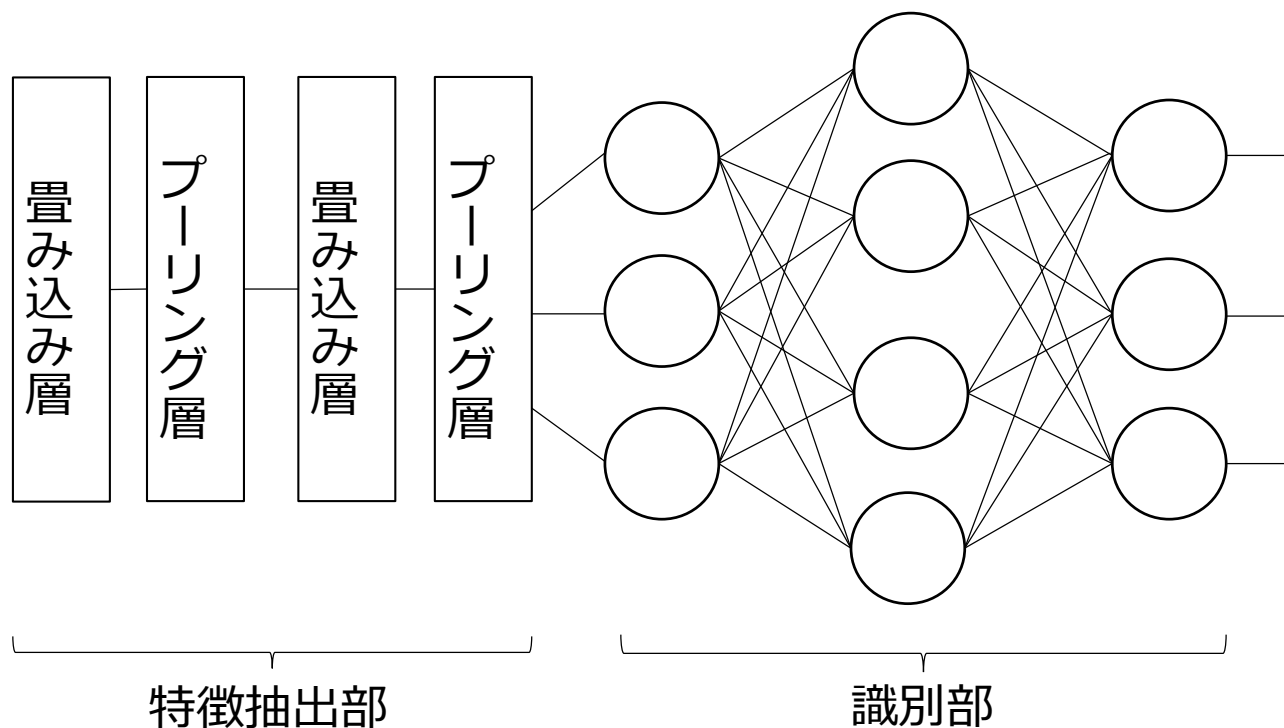


$$y = f \left(\sum_{i=1}^n x_i w_i + b \right)$$

畳み込みニューラルネットワーク

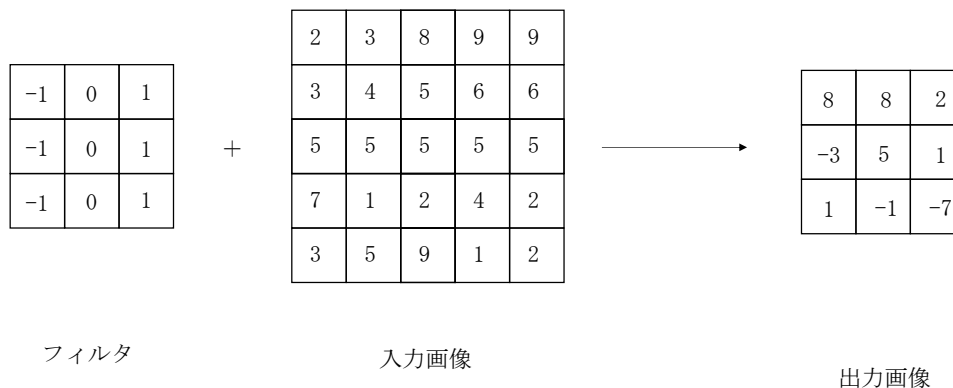
畳み込みニューラルネットワーク

ニューラルネットワークに畳み込み層とプーリング層の組を追加したもの

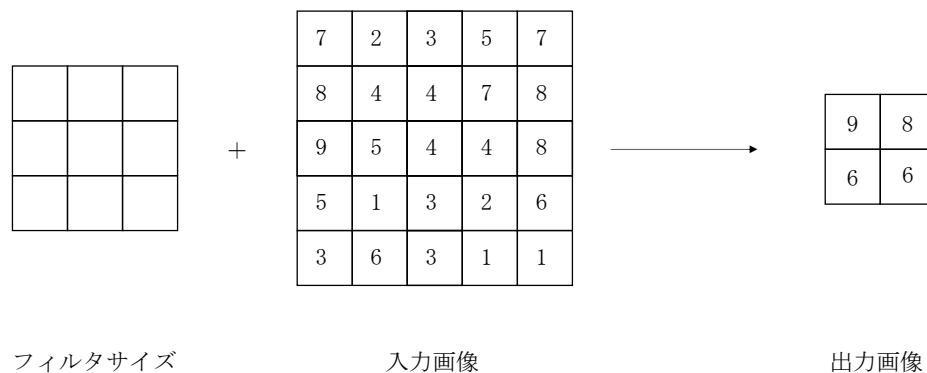


畳み込み層とプーリング層

畳み込み層



プーリング層

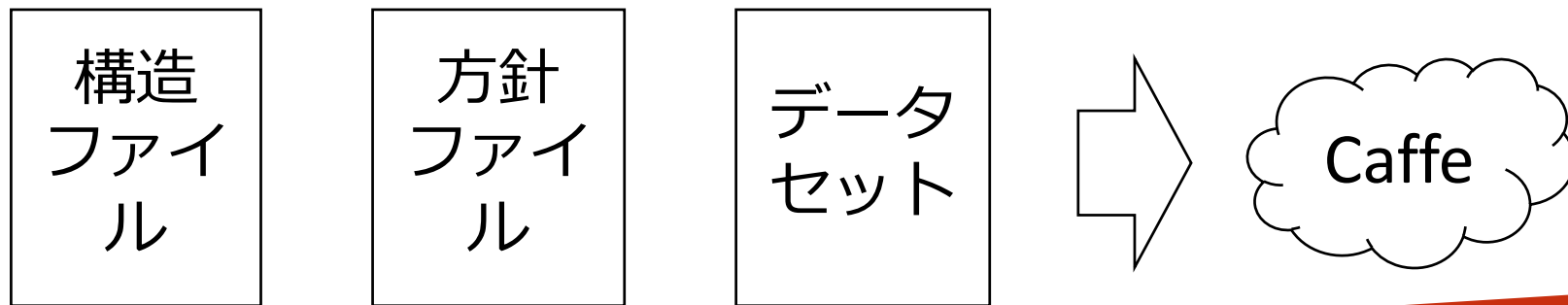


Caffe

Caffeとは、オープンソースのディープラーニングのフレームワーク（カリフォルニア大学バークレー校のYangqing Jia氏が開発）

- ・ CNNの構造を記したファイル
- ・ 学習の方針を記したファイル
- ・ データセットのファイル

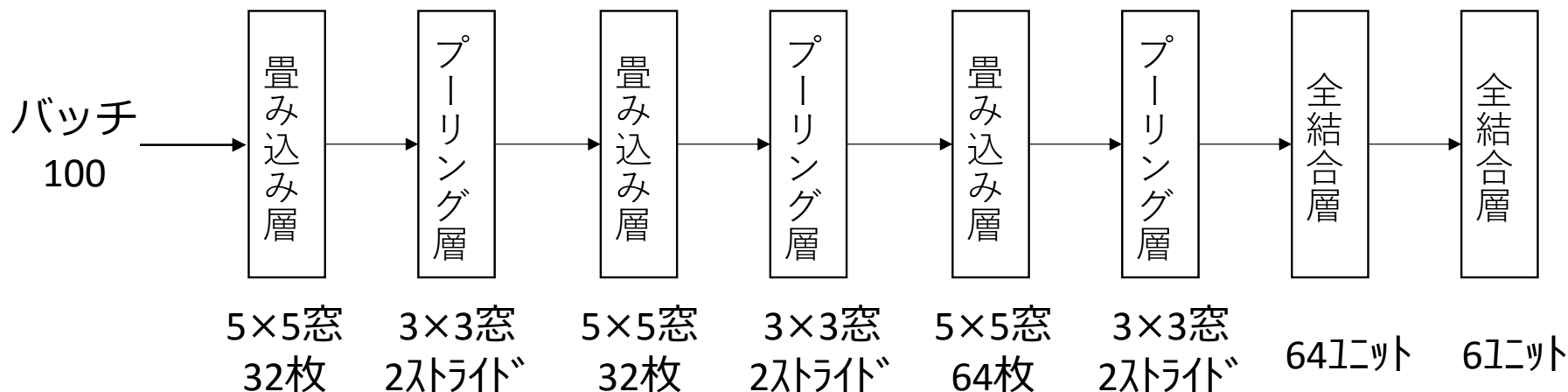
以上3つのファイルを用いて深層学習を行う



CNNの構成

CNNの構成を以下に示す

Caffeに用意されているCIFAR-10向けのサンプルを基にしている



下の数字

畳み込み層：フィルタサイズ
フィルタ枚数

プーリング層：フィルタサイズ
ストライド

全結合層：ユニット数

ソルバの設定

学習の方針を記したファイルはソルバと呼ばれる

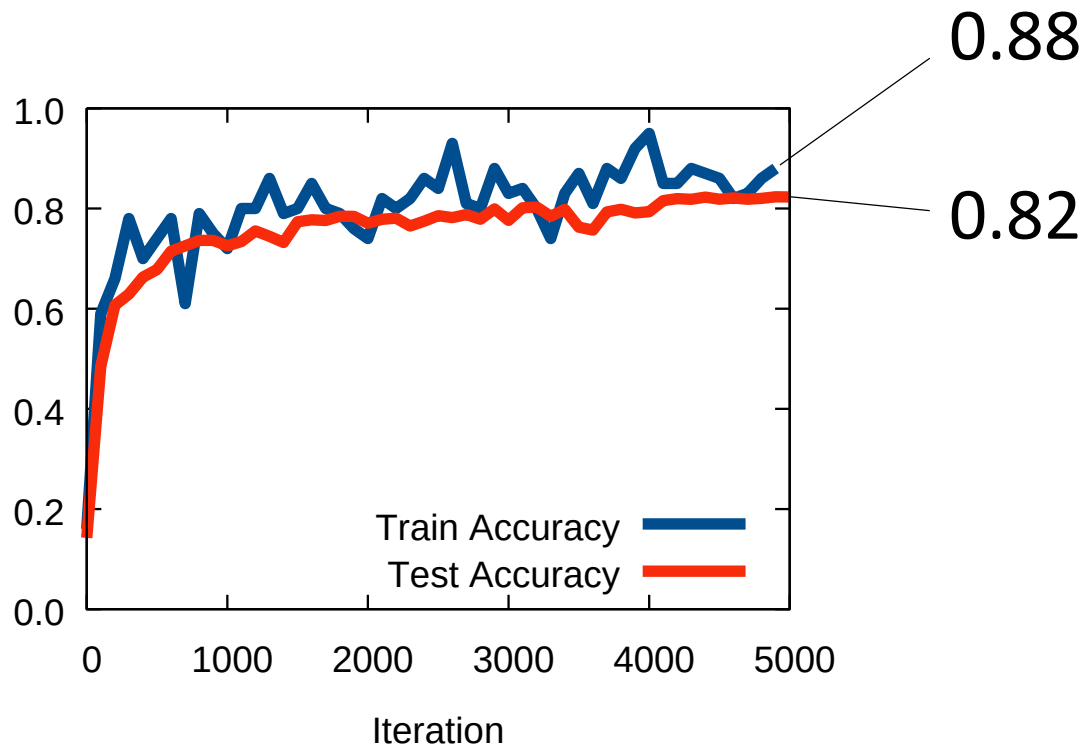
ソルバの設定を以下に記す

これもCIFAR-10向けのサンプルを基にする

- ・ イタレーション : 5000
- ・ テスト間隔 : 500
- ・ テストイタレーション : 58
- ・ 学習率 : 0.001 (イタレーション~4000) 、
0.0001 (イタレーション4000~5000)

以上の設定で学習を行う

学習と正答率の推移



- Train Accuracyから学習は十分
- Test Accuracyから汎化は十分
- × 異なる場所で撮影された対象動物への精度が未知

テスト用画像に対する分類器の回答の内訳

回答 出題	アライグマ	タヌキ	ネコ	シカ	イヌ	ウマ	FRR
アライグマ	1592枚	6枚	0枚	2枚	0枚	0枚	0.005
タヌキ	2枚	197枚	0枚	0枚	1枚	0枚	0.015
ネコ	2枚	0枚	670枚	91枚	204枚	33枚	0.230
シカ	3枚	1枚	85枚	789枚	65枚	57枚	0.211
イヌ	0枚	0枚	187枚	48枚	720枚	45枚	0.280
ウマ	1枚	0枚	50枚	66枚	77枚	806枚	0.194
FAR	0.005	0.034	0.325	0.208	0.325	0.143	

FRR : False Reject Rate (本人拒否率)

FAR : False Accept Rate (他人受容率)

Yes, we ♥ FPGAs!

まとめと今後の課題

- 本研究では害獣認識のためにデータセットと分類器を作成
- データセットは合計で37,300枚
- 分類器は約82%の認識率 ⇒有望
- 他の場所で撮影された対象動物をデータセットに追加、充実させる
- 学習の際のパラメータの適切な値の調査

組み込みシステム化の検討

組み込みシステム化

アライグマを自動認識するカメラシステムの開発

自動検知システムを多数設置 ⇒ 安価な組込システム！

様々な姿勢のアライグマを認識 ⇒ 機械学習を使用



限られた計算資源 & 機械学習は計算量が多い



いくつかの機械学習認識手法を試行し，比較検討

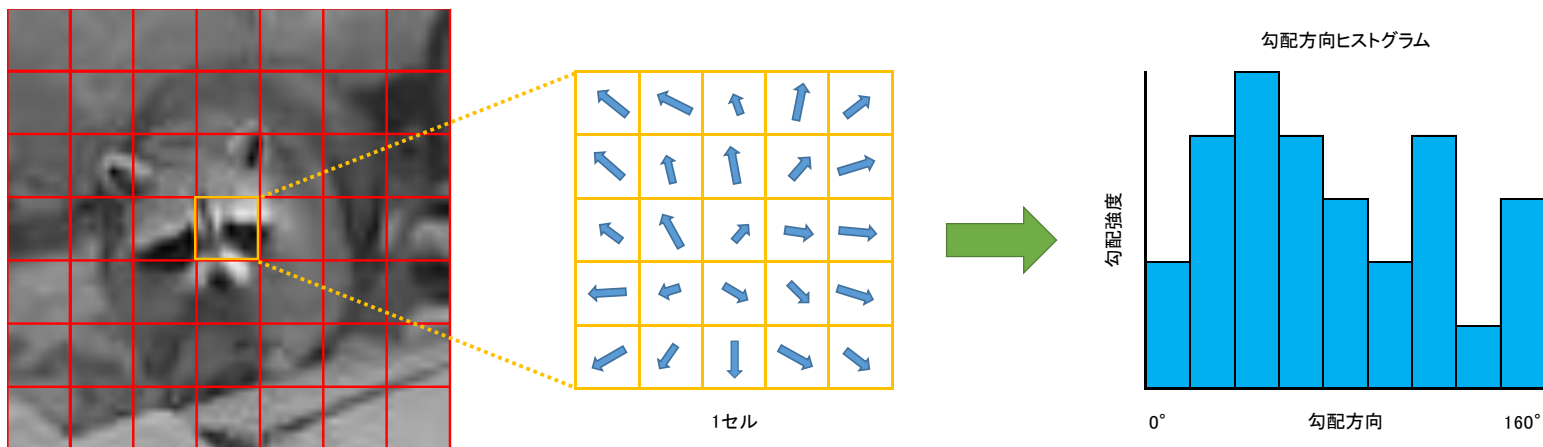
実際に組込システム上で動作させ，実行時間の計測

比較検討する認識手法

- **HOG特徴量 + サポートベクタマシン** (HOG+SVM)
 - 人検出によく使用され, 良い精度が得られている
- **HOG特徴量 + ニューラルネットワーク** (HOG+NN)
 - 3層ニューラルネットワークを用いてなるべくノード数の削減を行う
- **畳み込みニューラルネットワーク** (CNN)
 - 現在, 画像認識において最も注目されている手法であり, 良い精度が出ている
 - 3つの手法の中で最も認識精度が高いことが予想される
 - 計算負荷が高い

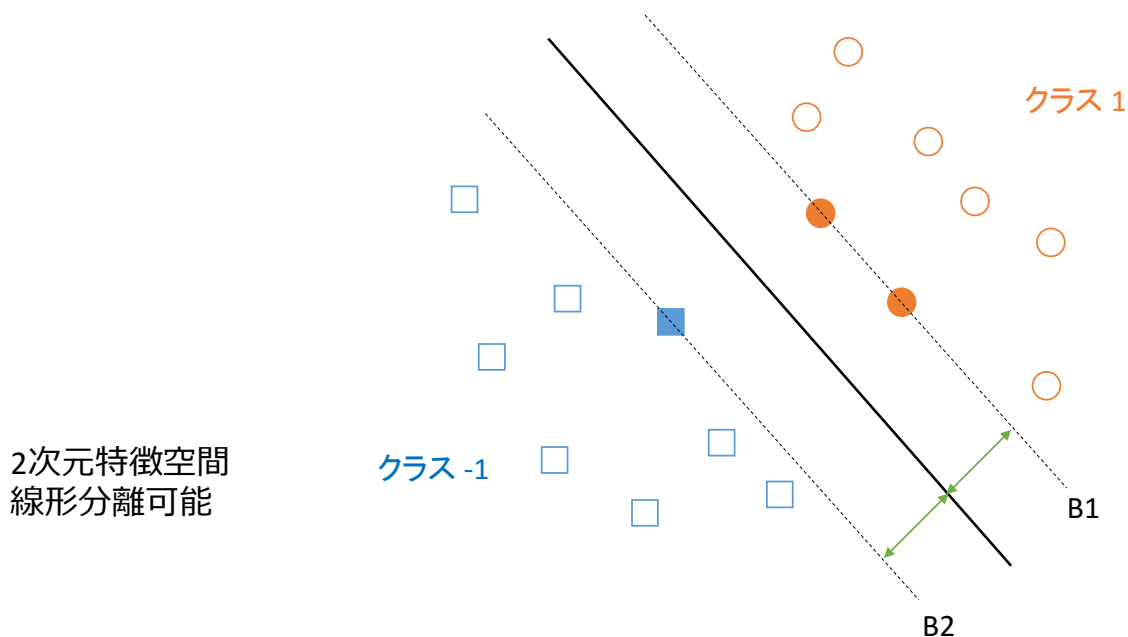
HOG特徴量 (Histograms of Oriented Gradients)

- 局所的な輝度勾配をヒストグラム化し，画像を表現する特徴量



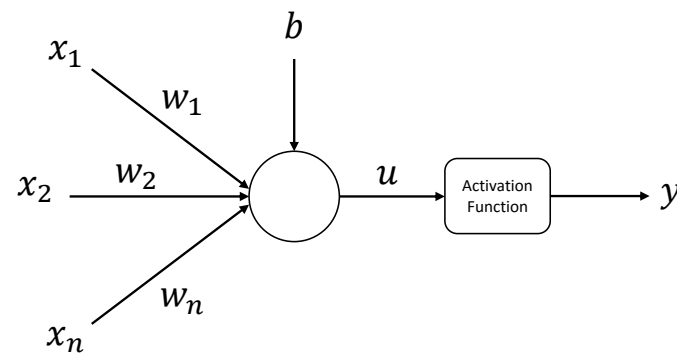
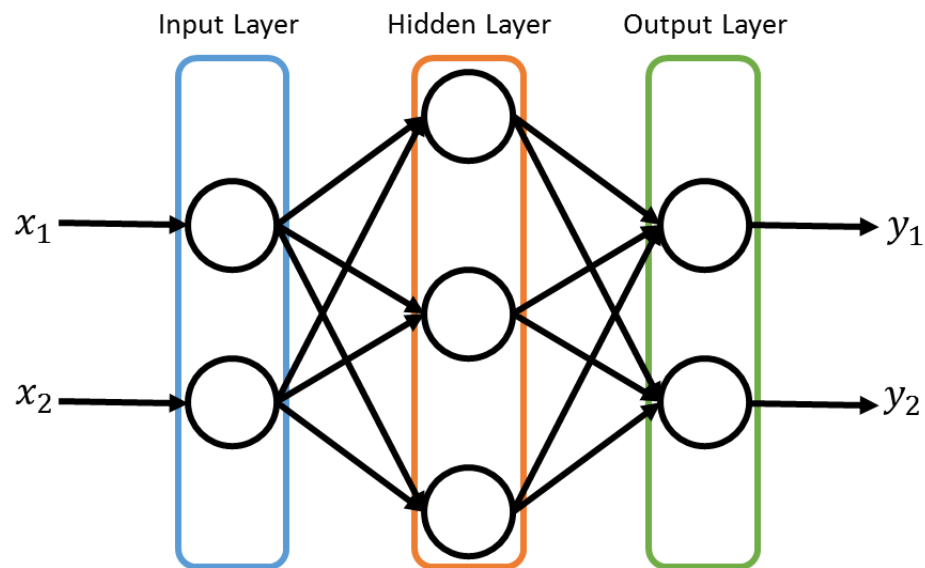
サポートベクタマシン (SVM)

- 二種分類によく用いられる機械学習手法
- 特徴空間の中のふたつの点集合を超平面で分離
- マージンを最大化するように識別境界面を決める



ニューラルネットワーク

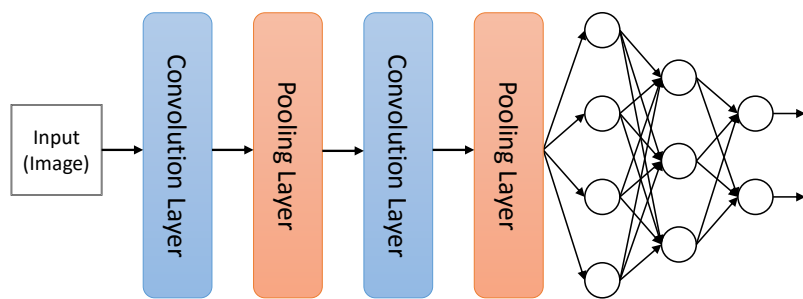
- 生物の神経回路網を模倣した人工の情報ネットワーク



$$u = \sum_{i=1}^n x_i w_i + b$$

畳み込みニューラルネットワーク (CNN)

- 主に画像認識に使用されるニューラルネットワーク
- 生物の脳の視覚野を参考に開発

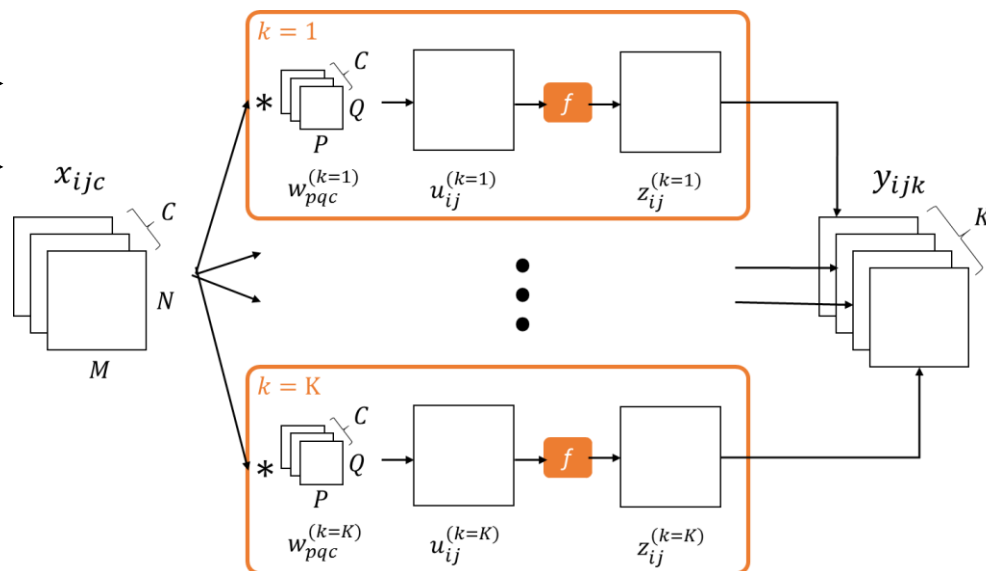


畳み込みニューラルネットワークのモデル

62	71	72	69	65	71	79	107
73	79	80	81	79	79	98	128
76	82	81	79	75	81	114	132
77	85	83	77	72	99	144	145
74	79	77	77	79	112	155	142
74	73	71	73	89	142	162	137
69	73	73	77	110	160	166	134
60	67	68	78	124	154	148	116

82	81	132
85	142	162
73	160	166

最大プーリングの例



畳み込み層の処理

使用する学習データセット

- 32×32のグレースケール画像
- 学習用画像枚数は65000枚
- テスト用画像枚数は2000枚
- 学習・テスト用画像は同じ場所で撮影
- 学習は「アライグマ」か「アライグマ以外」の二分類で行う

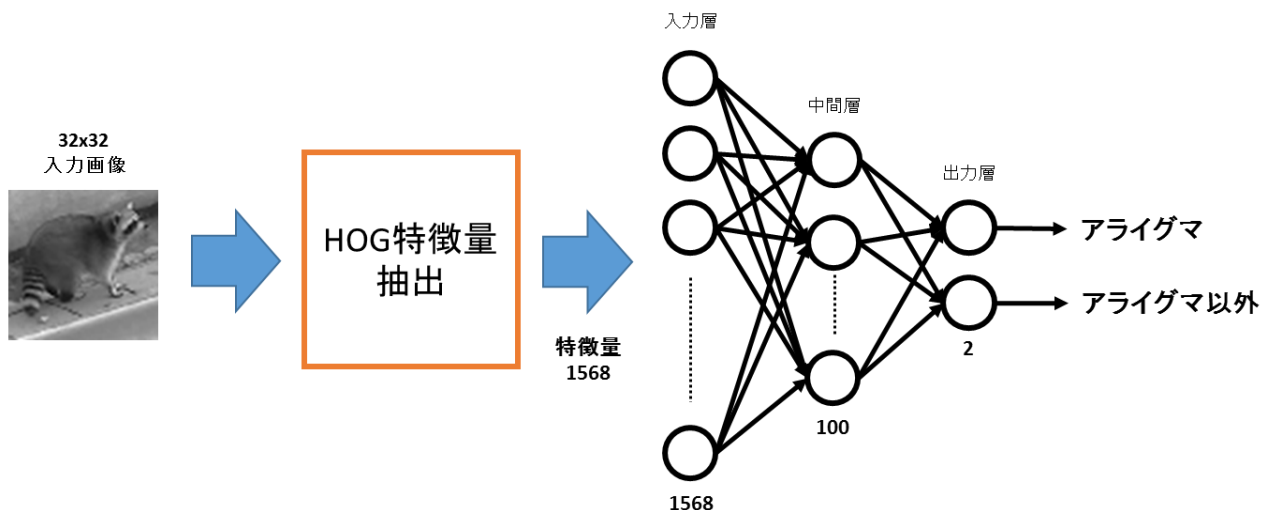
	アライグマ(0)	タヌキ(1)	背景(1)	合計
学習用	20000枚	5000枚	40000枚	65000枚
テスト用	1000枚	500枚	500枚	2000枚

HOG+SVMによる認識

- HOG特徴量抽出はC言語で記述し, SVMにはlibsvmを使用する
- libsvmは国立台湾大学で開発されたオープンソースのSVMライブラリ
- ターミナル上において"svm-train"コマンドで学習
- libsvmは"svm.h"をインクルードすることにより, C++から使用可能
- テストを行うプログラムをC++で記述
- 学習済みmodelファイルを読み込むことにより, このプログラムでテストを行う

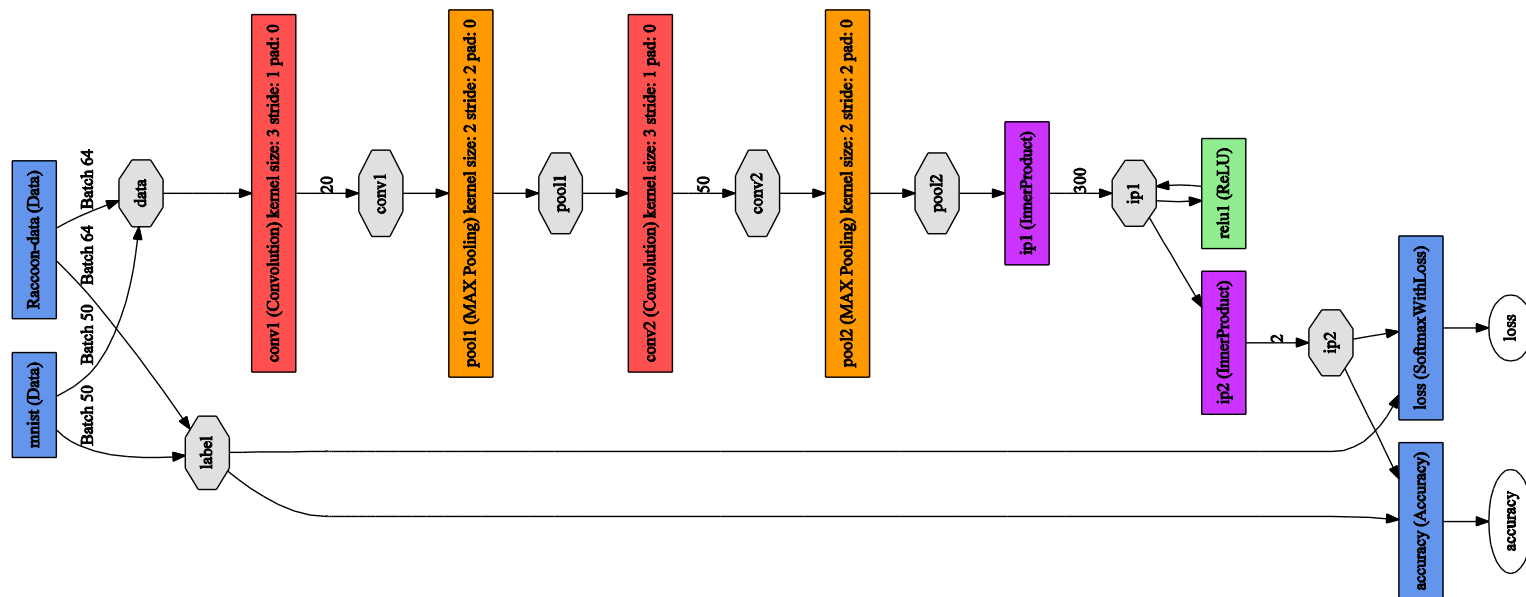
HOG+NNによる認識

- HOG特徴量抽出・ニューラルネットワークをすべてC言語で記述
- このプログラムにより学習からテストまで行う
- ニューラルネットワークの中間層のノード数は100
- 学習は150回



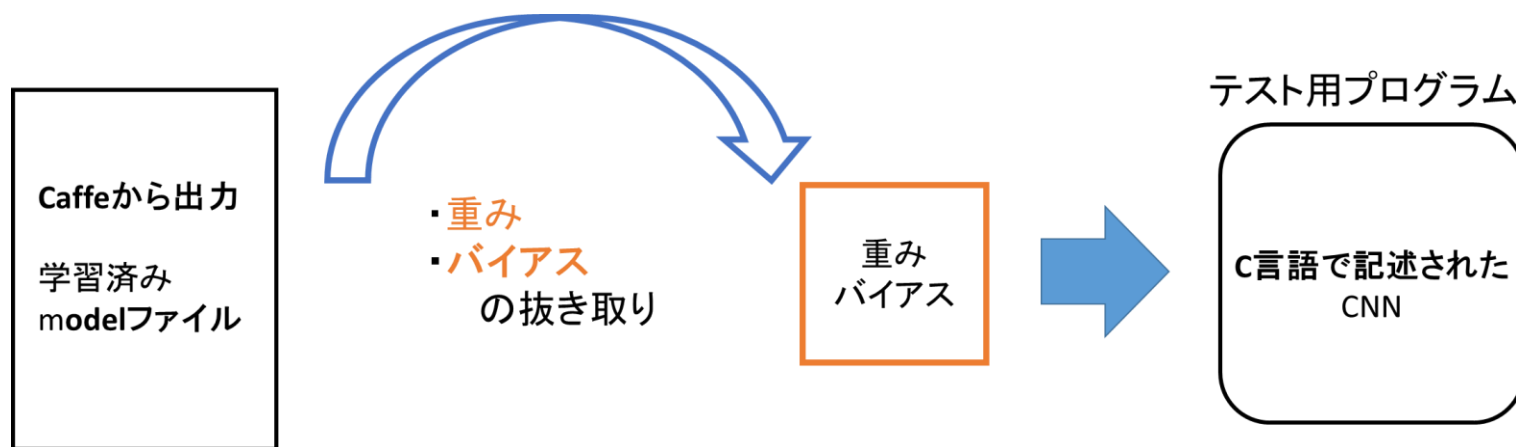
CNNによる認識 (学習)

- 学習にはCaffeを使用し, GPUを用いて行う
- Caffeは深層学習のライブラリであり, カリフォルニア大学バークレー校が中心となって開発している



CNNによる認識

- Caffeで使用したネットワーク構造と同様の処理をC言語で記述
- 学習済みmodelファイルから重みとバイアスのパラメータを抜き取るプログラムをPythonで記述
- 抜き取ったパラメータを使用し, C言語で記述したプログラム上でテストを行う



実験結果

各手法における正当率と計算時間

	正当率(%)	計算時間(ms)
HOG+SVM	89.75	111.355
HOG+ニューラルネットワーク	88.5	0.681
畳み込みニューラルネットワーク	94.45	11.898

実行環境

OS	Ubuntu 16.04 64bit
CPU	Intel core i7 6700K 4.0GHz 8MB

各手法における各画像の正当率

	アライグマ(%)	タヌキ(%)	背景(%)
HOG+SVM	86	89	98
HOG+ニューラルネットワーク	84.5	88.6	96.4
畳み込みニューラルネットワーク	91.9	95.6	98.4

インターネット上の画像によるテスト

- インターネット上の画像からテストデータを作成しテスト
- 合計100枚（32×32グレースケール画像）

	アライグマ	タヌキ	背景	ネコ	人間	合計
枚数(枚)	50	10	20	10	10	100

HOG+SVM

	アライグマ	アライグマ以外	正答率(%)
アライグマ	30	20	60
タヌキ	4	6	60
背景	0	20	100
ネコ	3	7	70
人間	1	9	90

HOG+ニューラルネットワーク

	アライグマ	アライグマ以外	正答率(%)
アライグマ	33	17	66
タヌキ	5	5	50
背景	0	20	100
ネコ	5	5	50
人間	0	10	100

畳み込みニューラルネットワーク

	アライグマ	アライグマ以外	正答率(%)
アライグマ	32	18	64
タヌキ	2	8	80
背景	0	20	100
ネコ	4	6	60
人間	0	10	100

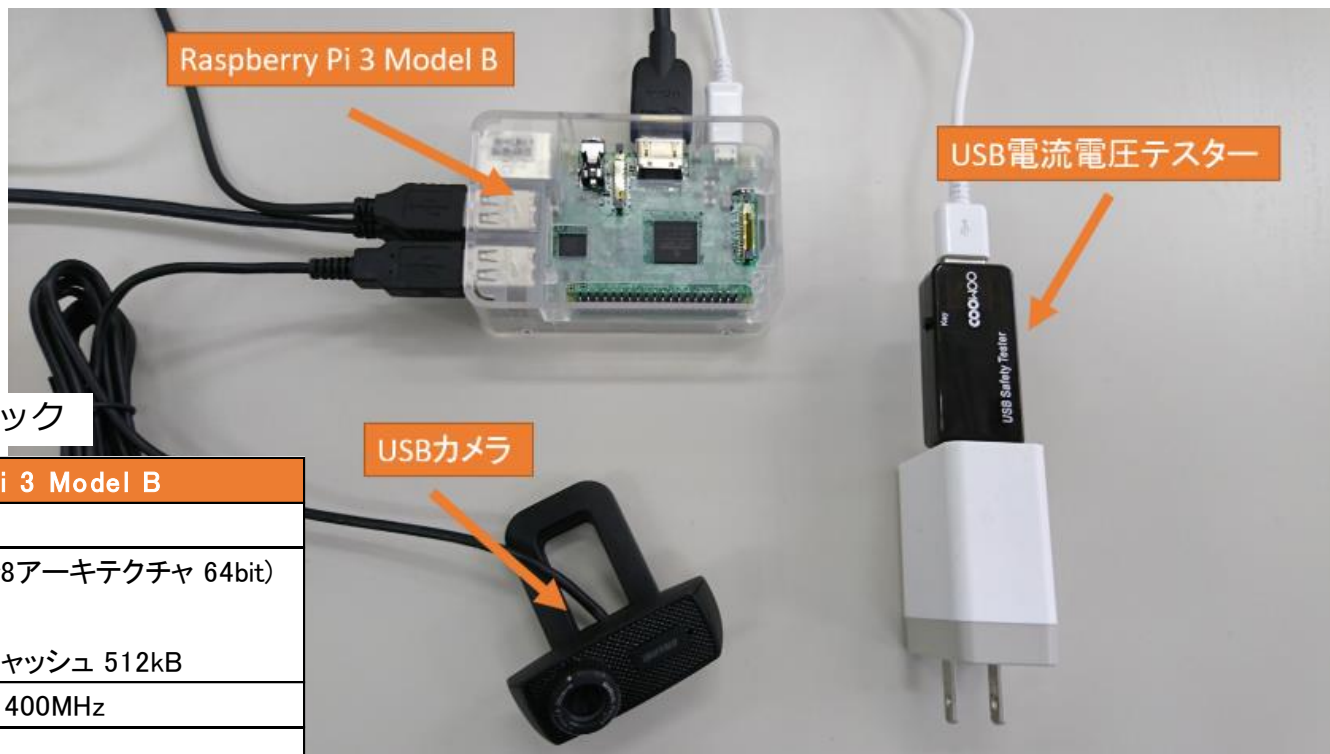
	正解	不正解	正答率(%)
合計	72	28	72

	正解	不正解	正答率(%)
合計	73	27	73

	正解	不正解	正答率(%)
合計	76	24	76

組み込みシステムへの実装 (使用機材)

- CPUボード…Raspberry Pi 3 Model B
- カメラ…バッファロー社製BSWHD06M (CMOSセンサ120万画素)
- OpenCVを使用し, USBカメラから画像を取り込む

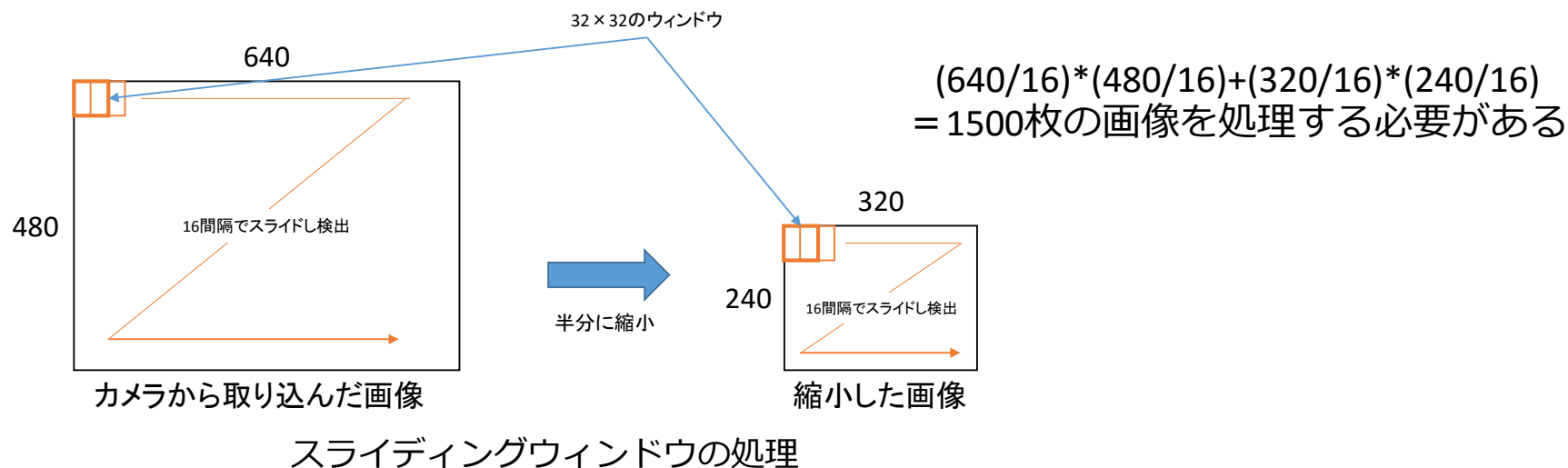


Raspberry Pi 3 Model Bのスペック

	Raspberry Pi 3 Model B
SoC	Broadcom BCM2837
CPU	ARM Cortex-A53 (ARMv8アーキテクチャ 64bit) 1.2GHz クアッドコア L1キャッシュ 32kB L2キャッシュ 512kB
GPU	Broadcom VideoCore IV 400MHz
メモリ	DDR2 450MHz 1GB

組込システムへの実装 (害獣検出方法)

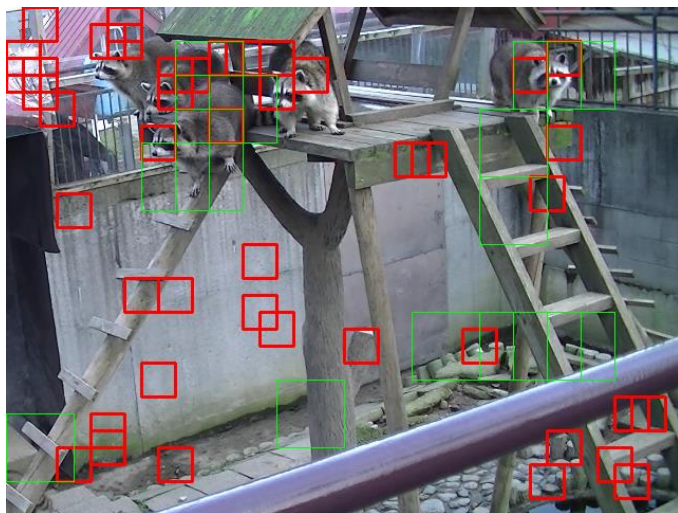
- VGAサイズ (640×480) 画像を処理
- スライディングウィンドウ方式によって検出
- アライグマが検出された場合, 画像上に矩形を描画



組込システムへの実装 (テスト)

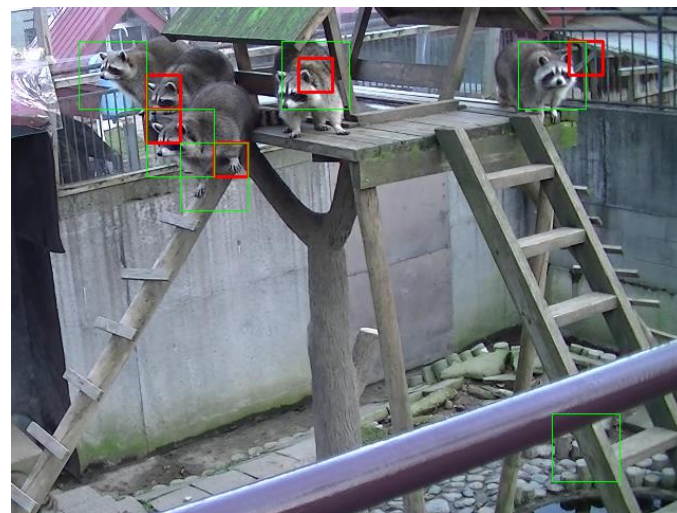
- カメラから取り込んだ画像を入力する前に，直接画像を入力してテスト
- 画像1枚あたりの処理時間を計測

HOG+NN



処理時間 : 約8.7s

CNN



処理時間 : 約45.8s

組込システムへの実装 (カメラを用いた検出)

- ディスプレイに表示させた画像をカメラで取り込む
- 畳み込みニューラルネットワークによる認識



処理時間 : 約41s

使用した物理メモリの最大 : 約37.5MB

検出処理中の平均消費電力 : 約2.5W

まとめ

- いくつかの機械学習認識手法を試行し，認識率や計算時間の観点から比較検討
- 実際に組込システム上で動作させ，実行時間の計測
- HOG+ニューラルネットワークが最も計算時間が速い
- 畳み込みニューラルネットワークが最も認識精度が良い
- 組込システム上にてVGAサイズの画像から検出を行った
- CNNを使用した場合，ある程度の精度が確認できた
- 処理時間削減のための工夫が必要